

Reducing Reworkable Discrepancies through Standard Work for Writing Inspection Instructions and Data Distribution

by

Jessica Bashkoff

B.S. Mechanical Engineering, University of Virginia, 2009

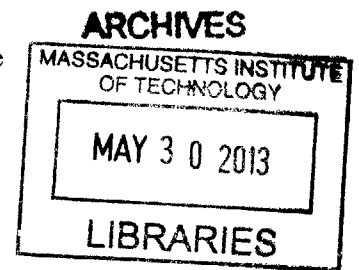
Submitted to the MIT Sloan School of Management and the Mechanical Engineering Department in Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration
and
Master of Science in Mechanical Engineering

In conjunction with the Leaders for Global Operations Program at the Massachusetts Institute of Technology

June 2013

© 2013 Jessica Bashkoff. All rights reserved.



The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author _____
Mechanical Engineering, MIT Sloan School of Management
May 10, 2013

Certified by _____
Daniel Whitney, Thesis Supervisor
Senior Lecturer, Mechanical Engineering and Engineering Systems Division

Certified by _____
Roy Welsch, Thesis Supervisor
Professor, MIT Sloan School of Management and Engineering Systems Division

Accepted by _____
David E. Hardt, Chairman, Committee on Graduate Students
Professor, Mechanical Engineering

Accepted by _____
Maura Herson, Director of MIT Sloan MBA Program
MIT Sloan School of Management

This page intentionally left blank.

Reducing Minor Rework through Data Distribution and Standard Work for Writing Inspection Instructions

by

Jessica Bashkoff

Submitted to the MIT Sloan School of Management and the Mechanical Engineering Department on May 10, 2013 in Partial Fulfillment of the Requirements for the Degrees of Master of Business Administration and Master of Science in Mechanical Engineering

Abstract

Each aircraft on the final assembly line inevitably has minor build errors and deviations from the intended design. Errors, like these, that can be corrected on the line are called reworkable discrepancies. Because hundreds of reworkable discrepancies can occur on each aircraft it is a huge opportunity for improvement and cost reduction. It would be impossible to analyze each one that occurs on the line. However, last year at X Aerospace another researcher created a tool to categorize discrepancies by key words based on free form text describing them. Root cause analysis and corrective action teams were put in place for the highest driving discrepancies but a more exhaustive analysis was not conducted.

In this research, to continue the root cause analysis, and gain a more detailed understanding of why discrepancies occur, a single aircraft was followed through final assembly and each discrepancy reviewed. Some key findings of this review were confirmation that the tool created previously was accurate, that there is a time lag between creation of a discrepancy and its discovery upon inspection, and that there is lack of access to specifications and data on the floor.

To address the issues with late discovery of discrepancies, more specific and guided inspection instructions are needed. Standard work has been developed for writing inspection instructions that are clearer. They include features such as guided and general tasks, references to specs for all directive instructions, specific locations on the assembly and drawings, and symbols and wording to indicate instruction versus reference information and which contain only instructions relevant for the particular inspection. Changes in management have halted the implementation of the standard work for writing inspection instruction. However, this standard work became the basis for changes being made to improve work instructions.

To address the lack of access to data, a tool which collects data and displays a cross Pareto heat map of the highest driving issues on each line, position, and work order has been developed. The line managers have been trained on how to use the tool to determine high driving issues in almost real time and implement corrective actions more quickly. In addition the tool is in use by quality engineers to help guide reduction efforts.

Thesis Supervisor: Daniel Whitney

Title: Senior Lecturer, Mechanical Engineering and Engineering Systems Division

Thesis Supervisor: Roy Welsch

Title: Professor, MIT Sloan School of Management and Engineering Systems Division

This page intentionally left blank.

Acknowledgments

First, I would like to thank everyone at X Aerospace for their help and support on this project. I was very impressed by the willingness of all the employees to candidly share their opinions and experiences. In particular I would like to thank my project supervisor who helped guide my project and ensured that I had the resources I needed to be effective.

I would also like to thank my thesis advisors Roy Welsch and Daniel Whitney for their constructive feedback on the progress and direction of my project along the way. Their guidance to resources also helped me significantly to find good ideas and best practices for handling rework and written instruction.

In addition, I would like to acknowledge the Leaders for Global Operations (LGO) program for the support of this work. I cannot imagine having an experience like I have had here on my internship with the ability to try to implement new ideas without it. I must also thank all my peers in the program who were always there to offer advice and support on anything from root cause analysis techniques to their experiences in dealing with difficult colleagues. I have learned so much from each and every one of my LGO classmates and cannot express how much I have appreciated this opportunity.

Last, but certainly not least, I would like to thank my family for always encouraging me to be my best and reach high for my dreams ever since I was a child.

This page intentionally left blank.

Author's Note

This thesis will refer to a large number of defects in aircraft final assembly. Because of this, readers not familiar with the aerospace industry or the production of complex systems may become concerned about the quality of the product. Most of the defects referred to are extremely minor and have no effect on product safety. X Aerospace and their employees are highly committed to producing the highest quality, safest product possible. In my work with X Aerospace I have observed this commitment to “quality first” in the work done at all levels. Though there are elements of the culture and the build process discussed in this report that can make it difficult to live up to the goal of “quality first” I have been impressed by the efforts of the company to reach this goal every day. This internship project serves only as a reinforcement of X Aerospace’s dedication to reaching their ambitious quality goals.

This page intentionally left blank.

Disguised Information

This thesis is based on the author's internship experience at a leading aircraft manufacturer located in the United States, identified in this thesis as X Aerospace. To protect sensitive information the name of the company and certain data will be disguised throughout this thesis. Identifiable sources will be masked, scale removed on certain graphs, data normalized, and certain analyses carried out with generic figures.

This page intentionally left blank.

Table of Contents

Abstract	3
Acknowledgments	5
Author's Note	7
Disguised Information.....	9
Table of Contents	11
List of Figures	15
1 Overview and Background	16
1.1 Company Background.....	16
1.2 Project Background and History.....	16
1.3 Cost of Poor Quality.....	17
1.4 Pareto Charts	19
1.5 Rework	20
1.6 Problem Description.....	22
1.7 Current State.....	24
1.8 Final Assembly and Inspection.....	25
1.9 Hypothesis	27
2 Organizational and Cultural Assessment.....	28
2.1 Interactions among Groups.....	28
2.2 Cultural Elements that Encourage Change.....	30
2.3 Cultural Elements that Inhibit Change	31
3 Detailed Analysis of Reworkable Discrepancies.....	34
3.1 Process of Analysis.....	34

3.1.1	Collecting discrepancy data.....	34
3.1.2	Tracking discrepancies and the defect analysis worksheet.....	35
3.1.3	Discussions with installers and inspectors.....	38
3.1.4	Gathering information from work orders and inspection instructions.....	40
3.2	Systemic Root Causes	40
3.2.1	Vague and inconsistent work and inspection instructions	40
3.2.2	Buried specifications and requirements.....	41
3.2.3	Lack of feedback and communication.....	42
3.2.4	Late discovery of discrepancies.....	43
3.2.5	Lack of training and awareness	44
3.3	Corrective Actions.....	44
3.3.1	Vague and inconsistent work and inspection instructions	44
3.3.2	Buried specifications and requirements.....	45
3.3.3	Lack of feedback and communication.....	45
3.3.4	Late discovery of discrepancies.....	46
3.3.5	Lack of training and awareness	47
4	Standard Work for Writing Inspection Instructions	49
4.1	Current Process.....	49
4.2	Literature Review of Inspection Improvement Techniques	49
4.2.1	Inspection allocation.....	50
4.2.2	Inspection training	50
4.2.3	Wearable virtual task guidance systems	51

4.2.4	Improvement of inspection instruction.....	52
4.3	New Standard Work for Writing Instructions	53
4.4	The Case for New Inspection Instructions	56
4.5	Current Progress of the Inspection Instruction Project.....	57
4.7	Future Use of Improved Instructions.....	58
5	Data Distribution Tool.....	60
5.1	The Shift in How X Aerospace Employees Think About Quality Improvement.....	60
5.2	The Need for a Data Distribution Tool.....	61
5.3	Stages of Development of the Tool.....	63
5.3.1	The first iteration: a manually made data display tool	64
5.3.2	The second iteration: a semi-automated display tool	65
5.3.3	The third iteration: a fully automated tool.....	66
5.3.4	The fourth iteration: added features.....	67
5.3.5	Technical Development and Coding of the Tool.....	68
5.3.6	The “Create_Tool” module	69
5.3.7	The “Get_Hyperion_Data” module.....	71
5.3.8	The “Tool_Use” module.....	71
5.3.9	Other forms and functions	72
5.3.10	Challenges in development.....	73
5.4	Roll Out of the Tool	74
5.5	Current Use of the Tool.....	75
6	Assessment of Results and Conclusion	77

6.1	Assessment of Corrective Actions.....	77
6.1.1	Standard work for writing inspection instructions.....	77
6.1.2	Data distribution tool.....	77
6.2	Assessment of Hypothesis.....	78
6.3	Areas for Continuing Improvement.....	79
6.4	Summary and Conclusion.....	80
7	References	82
8	Appendix	84
8.1	Defect Analysis Worksheet	84
8.2	Installer/Inspector Interview Questions.....	87
8.3	Design of work cards for aircraft inspection guidelines.....	88
8.4	Inspection Instructions Template/Standard Work.....	90
8.5	Heat Map Creation Program and Heat Map Tool Code.....	109
8.5.1	ThisWorkbook Module.....	109
8.5.2	Create_Tool Module.....	111
8.5.3	Get_Hyperion_Data Module	127
8.5.4	Script Sample.....	130
8.5.5	Tool_Use Module.....	131
8.5.6	TimeScrolling Form Module.....	132

List of Figures

Figure 1: Cost of Quality Iceberg (Krishnan, 2006).....	18
Figure 2: Effect of Varying Controllable PQC (Harrington, 1999).....	19
Figure 3: Pareto chart example.....	19
Figure 4: Recommended Scrap and Rework Reduction Approach (Berk, 2010).....	22
Figure 5: Vicious cycle of decreased long term solutions.....	32
Figure 6: Defect analysis worksheet (For a larger view see the appendix).....	35
Figure 7: Discrepancies written outside their build work order on Effectivity 17.....	43
Figure 8: Cross Pareto Heat Map Example.....	61
Figure 9: The user interface to create the heat map tool.....	66
Figure 10: Layout of the heat map tool.....	68
Figure 11: Time scrolling form.....	68
Figure 12: Simplified word parse table for creating heat maps.....	70

1 Overview and Background

1.1 Company Background

X Aerospace, established in the early 1900s, is a leading producer of commercial and military aircraft including helicopters and fixed-wing aircraft. They also provide spare parts and maintenance, and repair and overhaul services, with over 20 assembly, manufacturing, and service facilities around the globe employing almost 20,000 personnel worldwide. X Aerospace is arranged into divisions focused on the manufacture and assembly of various parts of the aircraft. The company is highly vertically integrated and historically produced most parts, exclusive of the engines of the aircraft, in its central factory. However, it has begun to utilize other internal and outside manufacturers for some of its parts. With the difficult economic climate X Aerospace has been striving to reduce cost of each aircraft it builds. One significant cost is rework on the aircraft. Reduction of the amount of rework will help bring X Aerospace's costs down and allow them to continue to be a leader in the industry.

1.2 Project Background and History

Rework has become a large focus of X Aerospace's cost reduction because rework adds up to tens of millions of dollars per year of direct costs to the aircraft. In 2011 the Vice President of Operations began an initiative to tackle the problem of rework. The goal of the initiative was a double digit reduction in rework counts. Additionally, reduction of rework was expected to help improve quality of the aircraft overall. In 2011, a quality engineer and a Leaders for Global Operations student were assigned responsibility for launching and sustaining initiatives to support the project. During this period the team determined a method to begin to manage rework through a number of methods. First the team had to determine the most common types of rework issues. Though there was a system in place to categorize the reported rework by numerical indicators, inspectors did not use the categories accurately or effectively. Because of this these categorizations did not give a clear picture of the highest drivers. In order to determine the most common problems a tool was developed to count the words in the free form text

written by inspectors describing a defect and create a cross Pareto of common words to determine most frequent parts and modes of failure. Once the most common rework was determined, teams were assembled to outline root causes and implement corrective actions. While most of the teams were successful implementing corrective actions, rework is an ongoing problem which changes constantly. Additionally, while the Pareto tool allows for the determination of specific parts and modes of failure, it does not address more systemic issues. This thesis aims to continue the reduction of rework begun in the year prior by confirming accuracy of the Pareto tool, expanding its use, and finding root causes and corrective actions for more systemic issues causing rework.

1.3 Cost of Poor Quality

For many years it was commonly believed in industry that providing higher quality products was more costly. However competition from international markets forced managers to begin to produce higher quality products to keep up with competition. This new focus on quality proved that high quality does not equate to higher costs and solving quality problems often resolves scheduling and cost issues as well. To make it easier to understand this, Armand Feigenbaum developed a dollar based system called “cost of quality,” now termed “poor quality cost (PQC)” which considered all costs that ensure jobs are done correctly every time, all costs to determine if output is acceptable, and all costs of failure to meet customer requirements or specifications. Optimum operating cost must be determined accounting for the cost of errors balanced with cost of quality (preventing errors). (Harrington, 1999)

There are two types of quality costs, direct and indirect. Direct costs include controllable costs such as those for prevention, appraisal, and other non-value added activities, resultant costs arising from internal and external errors such as the rework of these errors, and poor equipment quality cost. Indirect costs are more difficult to measure but cover things such as loss of reputation, customer dissatisfaction, and other lost opportunity costs. Most focus is generally on direct costs because they are more easily addressed even though indirect costs can be much higher. (Harrington, 1999) This is often depicted as an iceberg with the largest parts hidden beneath the water. See Figure 1. (Krishnan, 2006)

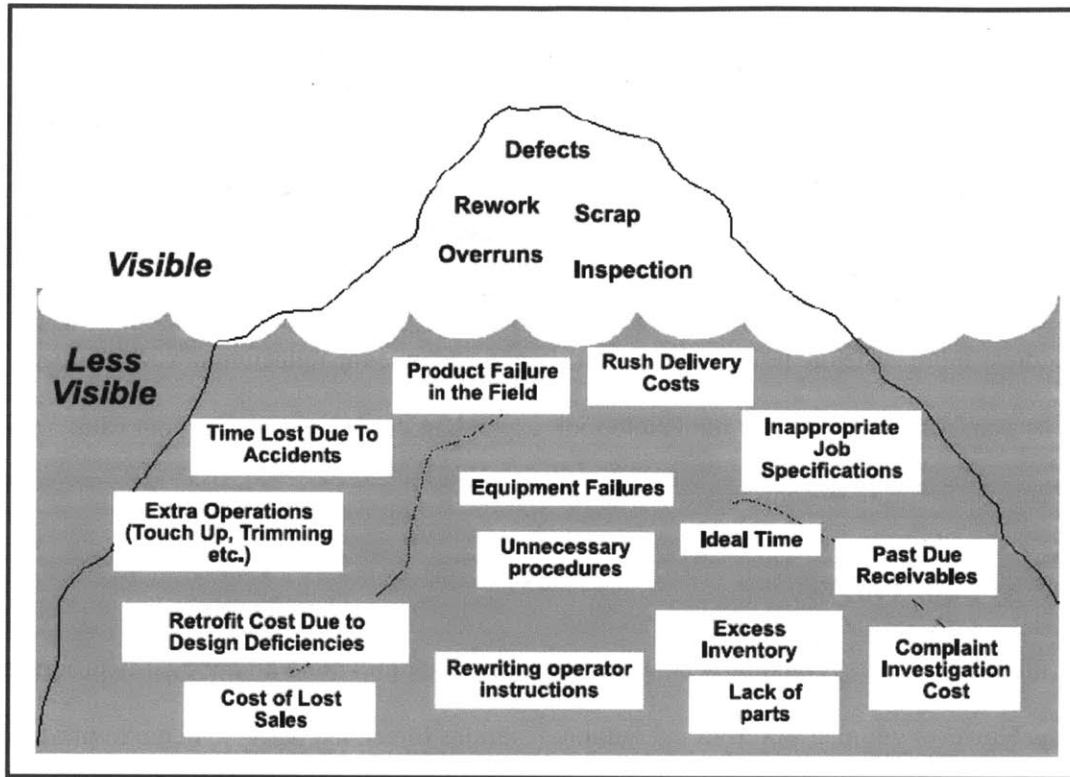


Figure 1: Cost of Quality Iceberg (Krishnan, 2006)

The effort of reducing rework is a direct, controllable cost of quality for prevention. Though prevention is usually touted as the best way to spend quality dollars it is often neglected because it is difficult to see the return on the investment. Actual inspection of work orders to find defects is a direct appraisal cost and the defects and the time to rework them are categorized as direct, resultant costs. Though determining how to reduce rework may increase prevention and appraisal cost, the resultant cost it saves should more than make up for it. This balance between controllable and resultant costs is shown below (Figure 2) and can help determine the best interim operating point. Increased appraisal reduces the risk of not finding a defect before the product reaches the customer or there is a product failure. Though not easily quantified this can also help reduce the indirect cost of not finding a defect such as customer dissatisfaction. (Harrington, 1999)

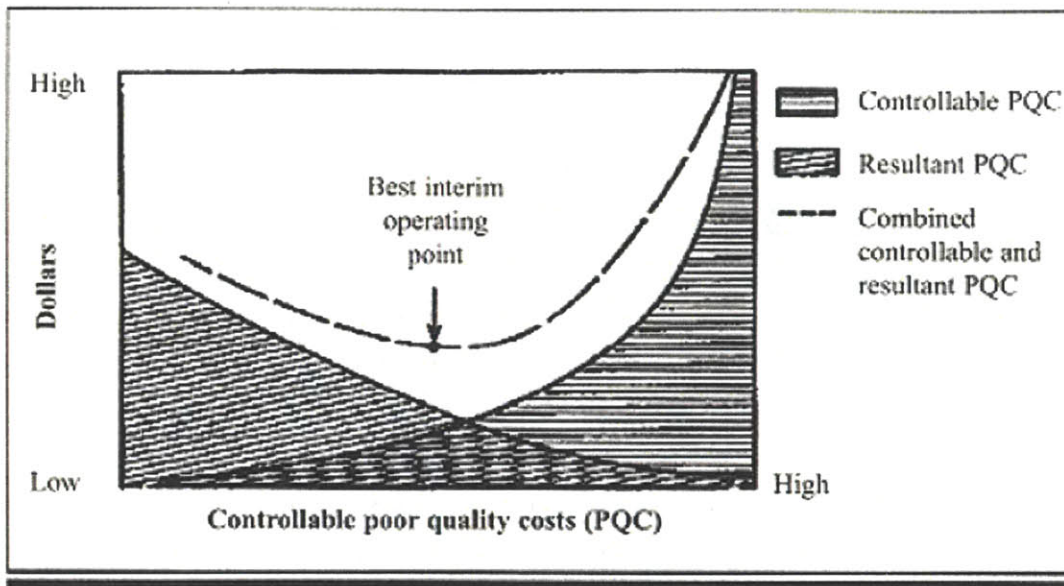


Figure 2: Effect of Varying Controllable PQC (Harrington, 1999)

1.4 Pareto Charts

The “Pareto Principle” states that a few specific contributors to the cost of quality (or anything else) are responsible for most of the cost. These are called the “vital few.” In this thesis the focus of most corrections will be on these “vital few” which are also termed “high drivers.” By working to correct these few issues you can significantly improve cost. A Pareto diagram is a chart that uses a bar chart of the counts of issues sorted by most common to least common on one vertical axis and a line graph to display

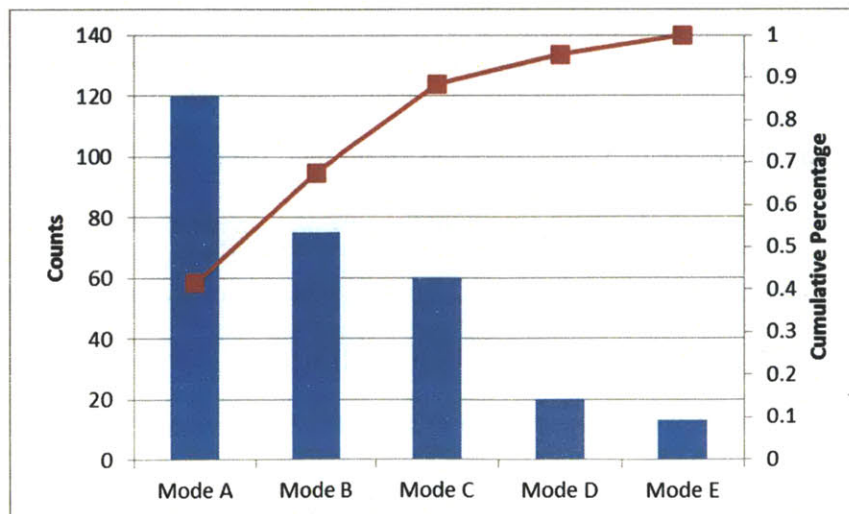


Figure 3: Pareto chart example

the cumulative percentage each issue contributes to the total on the other vertical axis. See figure 3. This clearly shows which issues are the vital few or high drivers. The Pareto chart can be used on multiple layers of quality issues including determining the highest driving defects, the defects' highest driving symptoms, and the symptoms' highest driving root causes. The analysis could also be used for cost rather than counts of defects because some defects may cost more to repair than others. A Pareto analysis helps control the scope of corrective actions. It does not make sense to spend the same amount of effort to address issues that only affect a small percentage.

Using the Pareto charts alone will help determine the best areas to focus corrective action. Using a Pareto priority index (PPI) can help further evaluate the value of a corrective action. PPI is defined as $\frac{\text{Savings} \times \text{Probability of Success}}{\text{Cost} \times \text{Time to completion (years)}}$. A project having the lowest cost or highest savings alone may not be the best to take on. Instead the project having a high PPI value should have the highest priority. (Gyrna, 2001) In this project PPI was not used because of the difficulty of estimating the probability of success, cost and time to completion.

1.5 Rework

Rework is a large component of cost of poor quality and is a common problem in many industries including construction (Love, 2000), automotive (Inman, 2003), aerospace (Dostaler, 2010), electronics (Scheffler, 1998), and even software (Boehm, 2001). Hegazy et al., defines rework as “effort of re-doing a process or activity that was incorrectly implemented the first time” (2011). Rework must be performed as a result of non-conforming material. Non-conforming means that “the product does not meet its drawing or specification requirements” (Berk, 2010). Non-conforming parts may also be termed “defects,” “errors” (Harrington, 1999), or “deviations” (Burati, 1992). When there is a non-conformance the part can (1) be used as is if it is deemed to meet form, fit, or function, (2) be repaired or replaced to meet the specifications, (3) be reworked to meet form, fit, or function but not necessarily specifications. The choice of which is acceptable is based on the needs of the customer. (Berk, 2010) Because customer

specifications are very stringent, non-conformances at X Aerospace are only dealt with in the second and occasionally the third manner. Rework at X Aerospace is broken down into three major categories, reworkable discrepancies, standardized repair for known recurring issues, and individual non-conformances which may require engineering intervention for mitigation. Reworkable discrepancies are variations and errors from the intended design specifications that can be corrected by installers on the floor using standard skills and processes. They usually consist of small mistakes such as misplaced or missing identification decals or areas requiring paint touch ups. This type of rework is the least expensive to carry out per error but is the most common. Reworkable discrepancies will be the focus of this thesis. The standardized repairs are slightly more expensive and require specific processes to correct defects. The final type of repair is the most expensive because it is on a part by part basis with unique solutions for each problem. Engineering must ensure that these types of repairs still conform to the original design intent in form, fit, function, and often specifications. Specifications can differ from required form, fit and function. For example, a specification may state a certain shade of paint must be used in touch ups to protect bare metal. Another shade of paint would give the same protection but not be as aesthetically pleasing. If for some reason the correct shade of paint may be more difficult to apply engineers may choose to use a different shade of paint even though it does not meet the specification but does afford the required protection.

Reworkable discrepancies, though lowest cost to repair on a discrepancy by discrepancy basis, cost X Aerospace over \$50 million dollars a year. This is calculated based on the cost of labor and the time to repair each defect including the inspection time necessary to validate the rework is acceptable. This cost excludes the additional cost of scrapped unrestricted material which is not sufficiently tracked but does add to the overhead cost of the aircraft. Because of the high yearly cost of rework, in 2011 management identified reworkable discrepancies a huge opportunity for improvement.

The cost of rework is often termed the “hidden factory.” Because rework cost is usually rolled up in normal manufacturing cost it becomes difficult to track and correct. Workers and managers become

accustomed to inspection and rework and do not realize the potential savings. In one example a motor home manufacturer hired 20% of its workforce just to perform inspections and touchups. When it is broken down so clearly into a touchup team it is easy to see the potential for significant reductions but it is usually not so simple to separate rework and repair costs. To move from detection to prevention, companies generally must have a shift in thinking. In order to begin to preventing defects rather than just repairing them Berk recommends the following approach (Figure 4). The major difference in the approach followed by X Aerospace and that recommended by Berk is that X Aerospace used many root cause analysis tools rather than just the tool of Ishikawa diagrams (also called fishbone or cause-and-effect diagrams), which categorize causes of an issue based on different categories, often including machine (technology), method (process), man (labor), measurement (inspection), and milieu (environment).

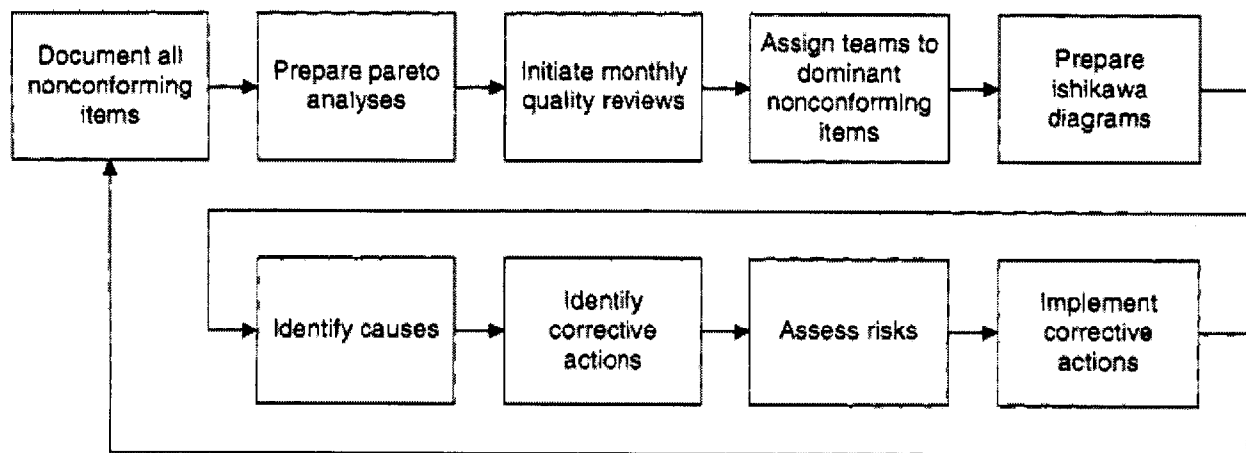


Figure 4: Recommended Scrap and Rework Reduction Approach (Berk, 2010)

1.6 Problem Description

At X Aerospace there is a large amount of inspection data available but it is difficult to access and utilize. In 2011 X Aerospace followed a process similar to that described by Berk above to focus on defect modes and parts and did reduce defect counts. However, reduction of rework is an issue requiring continuous monitoring and improvement. While some high driving modes and parts of failure were addressed last year there was not a focus on more systemic, overarching issues in regard to reworkable

discrepancies. Determining the root cause of these more systemic problems and implementing corrective actions is a key step in the reduction of rework. By taking corrective actions against systemic issues reductions can be made across many defect modes which will supplement the corrective actions taken to reduce specific defect modes. Rework reduction is important because it reduces costs, improves the speed of the aircraft build and improves quality of the aircraft overall.

To determine these systemic issues a root cause analysis of multiple discrepancies was performed. The process of the analysis is described in chapter 3. From the root cause analysis several systemic issues were discovered. The first major systemic root cause was with the clarity of work and inspection instructions. Many discrepancies arose due to the misunderstanding of work or inspection instructions. Additionally, because work and inspection instructions, as well as specifications, have some grey areas there can be disagreement among installers and inspectors on what constitutes a defect. With this confusion some discrepancies that are not considered defects by one inspector are discovered at a later inspection by another inspector. This creates more late stage rework. The second major systemic root cause was the lack of easy access to specifications and data. Specifications, which contain requirements for certain build qualities, were often difficult to find. They were often not listed in work instructions and required build specifications could be listed in multiple locations. Additionally, well organized, clear data on defects was not available to managers and supervisors on the manufacturing floor. Without data on what the highest driving types of defects (the vital few as described in section 1.4) are it is difficult to take corrective actions to fix these problems. Quality handles most of the discrepancy data and is therefore given the responsibility for correcting the root causes of defects. However, in order to effectively implement these changes leadership and support of the managers and supervisors is required. Without access to discrepancy data it is very difficult for those closest to the issues on the floor, managers and supervisors, to be involved in creating, applying, and monitoring the success of solutions.

In general, reworkable discrepancies are trivial errors when looked at individually, but these errors compound and take a large amount of time, labor, and therefore money to repair. The resources used to

repair discrepancies could be utilized, instead, in correcting larger, more critical problems if reworkable discrepancies could be reduced.

1.7 Current State

The most common form of rework at X Aerospace is termed a reworkable discrepancy. Reworkable discrepancies are small errors that are discovered through inspection after an installation has been completed. They are then corrected very soon after discovery and reinspected to ensure compliance with drawing requirements and specifications. Reworkable discrepancies cost X Aerospace \$50 million per year. Though X Aerospace has found a way to quantify the cost of rework, that in itself will not reduce the costs but can be used as a good baseline for measuring improvements.

Last year there was high enthusiasm for the project of the reduction of reworkable discrepancies. However, this enthusiasm has waned. At the start of the project there had been weekly meetings to talk about the progress and results of the project but as the year progressed and manufacturing schedule pressure became greater these weekly meetings were canceled and became per month occurrences. Additionally, as the corrective action teams completed many of the steps required to implement their changes their improvement projects became less pressing. While all the steps needed to implement changes were in place, the most important step, actual implementation, was not carried out. The teams consisted of employees from the Engineering, Manufacturing, and Quality functions. However, it seemed as if the teams felt that their responsibility was to complete their action items but the responsibility of actual implementation was to fall on the Quality function. The reduction project as it had been initially conceptualized was losing momentum and failing because all functions were no longer vested in the project. Management wanted to move away from the approach that reduction efforts were only Quality's responsibility. To make lasting changes all functions needed to be involved. There was a push to make all functions aware of how big the problem of rework was and to get each one involved in making changes in order to reduce defects and rework.

1.8 Final Assembly and Inspection

At X Aerospace the final assembly of the aircraft is conducted on an assembly line. The plant has multiple final assembly lines each for different aircraft model, though occasionally, other models are built on each line. Each assembly line has a number of positions where work on the aircraft is conducted. Each position has specific work orders which must be completed before the aircraft can roll to the next position. Work orders are tracked by a computer system. Every work order in the computer system contains instructions for the installation divided into specific operations. Some work orders contain additional instructions such as step by step picture guided installations, drawings, or specifications. Others have very limited instruction. Drawings and specifications, if required, may also be found on the X Aerospace intranet. In general, many work orders are purposefully developed to be able to be completed by an installer in an eight hour shift. Employees must clock in at the beginning of a shift on the work order they will be working on for that day and sign off on each operation as it is completed. The system tracks the installer and the time it takes to complete operations. However, because the installer will often work several operations before logging back into the system, the recorded buyoff time for several operations may be seconds apart and the times for each operation are therefore not particularly accurate. Work orders are also divided into two skill sets, mechanical and electrical. Electricians work on electrical work orders such as wire routing and clamping. Mechanics work on mechanical installs such as plumbing or door installations. Because installation functions are split in this manner there may be both a mechanical and an electrical work order to perform installations in the same area.

After each work order is completed an inspector checks the work to ensure that it meets drawing requirements and specifications. Because there are fewer inspectors than installers inspections may not occur immediately after the install is complete. Inspectors are also divided into mechanical inspectors and electrical inspectors. Both mechanical and electrical inspections have a guide document to help the inspectors to know what to look for during an inspection. However, the guide is incredibly long, detailed, and is not broken up by work order. The inspector checks the install and records the discrepancies into the

computer system using free form text to describe the defect and a three dimensional location (station line, water line, and butt line) to describe the location of the defect. Each reworkable discrepancy should be reported separately, though similar errors in the same area may be reported together. The inspectors do not inspect operation by operation and though discrepancies are supposed to be recorded for each operation it is inspectors do not look at the work order to try to determine which operation a discrepancy belongs to. Additionally, it is often difficult for inspectors to know in which operation a defect was created. Most defects are recorded simply under the inspection operation.

Each day, after reworkable discrepancies are reported the installer who seemed to have bought off the install is given a list of the reported defects by their supervisor. Because there is not always clarity in who created the defect, the installer who created the defect is not guaranteed to perform the repairs. They must log into the computer system, open the work order on which there are defects, repair the defects, and buy off the work again. The corrected defects are then reinspected and may be signed off by the inspector or resubmitted for additional rework. If they are submitted for additional rework the installer must make the correction and the inspector must then reinspect the work.

In addition to the inspections after each work order there are also some within the work order and work orders that are solely for inspection. Work orders that are solely for inspections are called intermediate shakes. Intermediate shakes usually occur before an installation that makes inspection of prior installs difficult, such a putting down the floor panels which cover much of the electrical wiring. During intermediate shake inspections discrepancies are recorded on the intermediate shake inspection work order though they were most likely created during an install. Finally, at the end of final assembly the aircraft goes through a total shake inspection where all parts of the aircraft are inspected. With intermediate shake inspections any installer working in the aircraft position who is familiar with the area may be asked to repair the discrepancy, and for final shake inspections any familiar installer may be asked to perform the repair. Then, similar to work order inspections, the repair is reinspected.

1.9 Hypothesis

The author's hypothesis consists of two assertions. First, that improved inspection instructions will not only help improve categorization and enhance early discovery of rework but will help reduce rework in itself. Second, that using better distribution of data to the line managers and supervisors will increase their involvement in rework reduction and allow for sustained reduction efforts. Much rework could be prevented with better work and inspection instructions and reporting of rework to managers, which encourages their involvement in creating and applying corrective actions.

As a manufacturer of complex machines, X Aerospace faces the problem of reworkable discrepancies, defects that occur and are repaired during the aircraft build. Though each defect may be small, the repair costs are significant and represent a huge opportunity for cost reduction and quality improvement. Though initially there is a cost to implement changes that reduce defects this cost must be balanced against the recurring cost of rework to obtain the optimal operating cost in regards to cost of quality. Though by using a tool to categorize defects into types and determine the highest drivers or "vital few" is an excellent strategy to reduce specific defect types X Aerospace must also determine and repair systemic issues that contribute to the root causes of all defects. By performing a root cause analysis the systemic issues of poor instruction clarity and lack of access to specifications and data were discovered. By improving instructions to remove obscurities and clarify directions and by distributing data to floor managers and supervisors rework can be reduced and the success rates of reduction efforts can be monitored and addressed.

2 Organizational and Cultural Assessment

X Aerospace is a well-established organization. As with many aerospace companies, their size and slow speed of implementation can make change difficult. In order to implement change the organization must be well understood. This chapter will discuss first the interactions between groups and individuals and how those interaction impact work at X Aerospace. Next, it will focus on elements of the organization and culture that drive change and how these can be used in implementing corrective actions to reduce rework. Finally, the chapter will address the cultural elements that inhibit change and how these issues can be overcome.

2.1 Interactions among Groups

As in all large companies X Aerospace requires input from many functions to design, build and improve their products and processes. The functions that have an influence on the reduction of internal reworkable discrepancies are Engineering, Manufacturing, and Quality. Supplier Quality may also become involved if there are defects coming in from suppliers. Engineering designs the parts of the aircraft and the manufacturing and assembly processes. Manufacturing focuses on the actual build of the aircraft. Quality focuses on improving the quality of the product and efficiency and execution of processes. Though the overall goal of all three groups is to output the best quality product at the lowest cost many times the short term objectives do not align. In addition to the misalignment of functional objectives there is not enough communication between functions and because of ambiguity in defined roles it is difficult to determine which functions are responsible for different problems with the build as they arise. Some examples of a few of these challenges are described here.

- Quality sees the opportunity for improvement of a process but because Manufacturing is focused on completing product build they may not willing to take the time to implement changes suggested by Quality.

- Manufacturing finds a difficulty with an installation. They report it to Engineering who takes the steps to implement a process change but these changes take months to be put into practice. While the old build method meets product drawing specifications it is still more difficult. When the changes are rolled out Manufacturing is not notified and continues to build to the old challenging process, believing that Engineering did not make the change.
- Quality would like to improve work instructions but work instructions are developed by Engineering. Engineering believes the work instructions are acceptable and does not want to make the changes.

Each department aims to meet its departmental goals because they believe that meeting these departmental goals is the best way to support the entire organization. For example, Engineering does not design for manufacturing or inspection because that increases the time for design. This leaves grey areas which can be left to interpretation by installers or inspectors. Inspectors may find discrepancies that are not justified because the discrepancy does not fall into their interpretation. Since finding defects is the definition of the inspector's job they believe they must report discrepancies to show a complete inspection. Manufacturing focuses on building the aircraft quickly to meet production schedules.

Each of the groups seems to work rather independently unless interaction is required. Additionally, each group does not seem to have a full understanding of what the other functions do. There is a lack of communication between functions. When issues occur they are not discussed between functions, so if a problem occurs in Manufacturing, Engineering may not ever know the issue is occurring. Instead blame is placed on other functions. Each function sees the others as taking credit for work they did not complete or blaming them for problems they did not cause. This incongruence causes problems to be passed from department to department until the issue is no longer viewed as pressing.

2.2 Cultural Elements that Encourage Change

X Aerospace is highly committed to the quality and safety of its product. All employees do seem to share this strong commitment to quality and safety. In general there is a common understanding that there must be continuous improvement to keep the quality of the product high. Because of this understanding there is a general openness to trying new ideas. Employees are also highly motivated to learn. They are interested in why things change and how the changes will make positive effects on the work and the product. In addition X Aerospace encourages continuous learning by supporting employees who wish to continue their education, providing the opportunity to continue technical training, and by holding seminars to teach employees about different topics. This curiosity also causes employees to recognize and question inefficiencies in their work and the system. There is a frustration from all employees when they do see inefficiencies in both the build process and the process improvement system which leads to the desire to improve both.

One element at X Aerospace that strongly encourages continuous improvement is the lean operating system. All employees are trained in the lean operating system and are required to adhere to the system which aims to reduce waste, improve the quality of products, and make processes more efficient. The overarching goal of the operating system is to continuously reduce cost while improving customer satisfaction. All employees from installers to engineers to are taught quality methodologies and are required to participate in continuous improvement events. Because employees are very familiar with continuous improvement they are more comfortable taking on changes to processes and systems so long as they understand the purpose of the change. Additionally, they receive feedback when improvements made through the lean operating system are complete, so they can see how their efforts impacted the organization.

Another part of the X Aerospace culture that is of note is that they are very data driven. They have many methods to collect and utilize data and base many of their decisions on data. This is positive in that it allows them to use metrics to measure the success of improvement efforts and monitor work for

degradations. Once an employee has data that shows there is an issue, they immediately want to enact change to improve the numbers. While the data driven culture is generally a benefit it can sometimes make implementing changes more difficult. This will be further discussed below in section 2.3.

2.3 Cultural Elements that Inhibit Change

Though there are many elements that drive change they are in competition with factors that make change much more difficult. In order to enact new, more efficient practices, including those to reduce rework X Aerospace must be open to adopting new and different processes. Because X Aerospace is a well-established company it has operating standards which have been followed for many years. Many people are very comfortable with these standards and processes. They believe “If it isn’t broken, don’t fix it.” So, even if changing a long standing procedure may have a great impact on cost of quality or efficiency, people are not inclined to make a change unless they see a marked decline in quality or efficiency, costs rise, or a specific issue arises. This goes along with the data driven culture. Though above the benefits of being data driven are discussed, note also that the reliance on data can inhibit change. Often times an employee sees repetitive issues, but because they do not have data showing the impact of the issue they will not take corrective actions. Their hunches are usually correct and had they taken action to rectify the issues early they could have been fixed more easily. However, without data to give a starting metric on the issue they will not take action. Data may be slower to show a trend than the intuition of a manager who is on the floor working everyday with issues that come up. Additionally, though X Aerospace does have an abundance of data available it often is not used appropriately.

One of the key deterrents to change is the lack of accountability. Because of the union environment there is a general lack of reprimand for poor work quality so often floor workers do not feel strong accountability for their work. Additionally, there is poor role definition for each function. When something goes wrong no functional group wants to take responsibility for the issue. This results in the finger-pointing and blame described in section 2.1 above. More time is spent trying to determine what function caused the problem than is spent actually correcting issues. Additionally, this blame makes

getting buy-in on improvement projects difficult because each function will not agree to take time to work on a project unless the other functions have committed themselves as well.

Another issue that inhibits change is that while there is a culture of continuous improvement it does not necessarily focus on the most pressing problems. This often leads to firefighting problems as they occur. Because firefighting focuses on immediate issues that need immediate solutions these solutions become merely band-aids for more serious problems and do not solve the root cause. The employees who solve the immediate issues are commended.

Often times when firefighting ideas for long term solutions are developed but they are never implemented. Long term solutions take longer to see results. Because their positive impact is more difficult to see, the motivation for creating

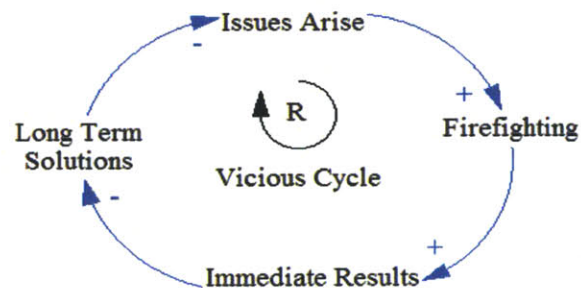


Figure 5: Vicious cycle of decreased long term solutions

long term solutions waned quickly. Additionally, when an employee is called to firefight they must drop any work on long term solutions to focus on solving the immediate problem. This forms a vicious cycle of decreased long term solutions (Figure 5).

In order to successfully implement initiatives to reduce rework X Aerospace must embrace change. Their culture has both elements that enable and inhibit change. Regardless of the quality of rework reduction solutions they may not be adopted unless the enabling cultural elements are utilized and the inhibiting elements addressed. One area where X Aerospace has the opportunity for improvement is in communication among functions. To help address communication issues clarity of instructions can play a key role. For example, if manufacturing instructions match inspection instructions there is less that can be lost in translation between installers and inspectors and, therefore, there is less chance for miscommunication. X Aerospace's highly data driven culture can enable change when data is available

but can inhibit change when data is not accessible or clear. The ability to easily access discrepancy data is an important change driver. By using their culture of high commitment to quality and continuous improvement X Aerospace can overcome their difficulties with role definition and a focus on short term “band-aid” solutions to implement changes that effectively reduce reworkable discrepancies.

3 Detailed Analysis of Reworkable Discrepancies

In order to better understand the rework that is required on an aircraft the author conducted a detailed root cause analysis of all the reworkable discrepancies reported. The goals of this analysis were to verify the accuracy of the cross Pareto tool and to determine systemic root causes of reworkable discrepancies. The beginning of this chapter will cover in detail the process followed to collect data on discrepancies, perform root cause analysis and develop corrective actions. Next, it will describe the findings of the root cause analysis. Finally, it will discuss potential actions and the decision on which corrective actions were executed in the scope of the internship project. The two corrective actions take will be discussed in detail in chapter 4 and chapter 5.

3.1 Process of Analysis

Because of the large number of reworkable discrepancies that occur on each aircraft it would be impossible to determine the root cause of each one. Instead one aircraft (Model A, effectivity, also known as aircraft number, 17¹) was followed from the beginning of final assembly to the final shake inspections. As described earlier, when the aircraft goes through final assembly each work order is inspected and, further, shake inspections are conducted. When a defect is discovered the inspector records the defect into a computer system. All these defects are identified with a number and stored in a database containing information on the discrepancy. The exception is during the final shake inspections, when the defects are documented on paper record sheets.

3.1.1 Collecting discrepancy data

A query tool such as Hyperion/Brio or Microsoft Access is used to download data on each discrepancy. To track all the discrepancies related to effectivity 17 a query was run a few times each day to download data on all discrepancies written against effectivity 17 since the previous query. This query displayed the following data for each discrepancy:

¹ Model and effectivity have been disguised. Effectivity is the term used for the aircraft number.

- Discrepancy identification number
- Model
- Effectivity (Aircraft number)
- Line position of the work order the discrepancy is recorded on
- Work order number that the discrepancy is recorded on
- Work order title that the discrepancy is recorded on
- Component part number
- General location on the aircraft
- Station line (X dimension location of the discrepancy)
- Water line (Y dimension location of the discrepancy)
- Butt Line (Z dimension location of the discrepancy)
- Installation type (mechanical or electrical)
- Discrepancy description (free form text)
- Time the discrepancy is recorded in the computer system
- Time the work order is bought off by the installer on the computer system
- Installer identification number
- Operation the discrepancy is recorded on (usually the inspection operation)
- Inspector

3.1.2 Tracking discrepancies and the defect analysis worksheet

An excel workbook was created to keep track of each discrepancy. For each discrepancy a new worksheet was created (Figure 6).

Section 1			Section 2		Section 3	
Model #	Effectivity	Position	Is this a repeat discrepancy?	Defect Type	Why is discrepancy occurring?	
Defect #		W/O #	How did the discrepancy occur?			
Defect Time Written		Oper #				
Location:			Who was the installer?		Why?	
Station Ln	Water Ln	Butt Ln			Why?	
Type	Component Part #				Why?	
Inspector					Why?	
Remarks			When did the discrepancy get created?	What Position?	Why?	
			What is the responsible unit?		Why?	
			What is the responsible function?		Why?	
			What is the build criteria?	(Attach documents if necessary)	Why?	
			What is the inspection criteria? (Attach documents if necessary)		Why? If more whys are needed continue on back of sheet	
					Root Cause:	
MFG Stamp	MFG UserID				How can this be avoided in the future?	

Figure 6: Defect analysis worksheet (For a larger view see the appendix)

The first section of the defect analysis worksheet contains data on the defect pulled directly from the database. The second and third sections required looking up and reviewing work instructions, specifications, and drawings, and speaking with installers and inspectors on the floor.

3.1.2.1 Defect Analysis Worksheet: Section 2

The second section of the defect analysis worksheet was used to help categorize the discrepancy, collect information on the accuracy of the discrepancy data and ensure that the discrepancy was valid.

The question “Is this a repeat discrepancy?” was asked to check if it was (1) a common defect type, (2) a repetitive defect that was known and had a standard repair already in place, or (3) a reopened defect, one which upon inspection after repair was still found to not meet specifications. The difference between a common defect type and a repetitive defect is that a common defect type occurs in many places, on many aircraft while a repetitive defect is one that always occurs and must be repaired and is often waiting for a permanent design change. For example, a common defect may be loose wire ties that occur in different places on each aircraft, and a repetitive defect may be a specific bracket that comes in from suppliers slightly too large and must be cut to size to fit. The bracket may be in the process of redesign but the build must continue with the larger bracket that must be modified for every aircraft until the design change is complete.

As defects were written and defect data was downloaded, patterns emerged in regard to common defect types. “Defect type:” helps categorize the defects into groups similar to the way in which the Pareto tool categorized them.

Though the installer number, position, and time of installer buyoff are pulled directly from the installation database, these refer to the work order which the discrepancy is written against. This may not be the same work order in which the discrepancy is created. For example, a loose wire connection discrepancy may be written on a floor panel install work order even though this work order calls only for placing and attaching the floor panels. Before the floor panels are installed all work that would be difficult

to inspect or repair after the install is double checked as part of the floor panel install work order. The issue of discrepancies written on a work order in which they were not created is further addressed in section 3.2.4 below. The questions “Who was the installer?”, “What Shift?”, “What Position?”, and “When did the discrepancy get created?” are in regard to the actual work order in which the discrepancy is created as opposed to the one in which it was written. This was often difficult to determine.

Similarly, some defects written on a work order are actually created upstream by a supplier. The question “Who is the responsible unit?” refers to whether the discrepancy was created by an installer or if a part was delivered from an upstream supplier with the defect already in place. The question “Who is the responsible function?” is generally in conjunction with “Who is the responsible unit?”. If the discrepancy is created upstream, Engineering usually tries to determine why the part is coming in defective. If the discrepancy is created by an installer it is Manufacturing’s responsibility to correct the discrepancy and hopefully work to prevent it in the future. However, preventative measures are not often completed as discrepancies are discovered. If an installer created defect is due to a difficulty with product or process design, Engineering may also be involved.

The final two questions in Section 2 of the “What is the build criteria?” and “What is the inspection criteria?” aim to first, ensure that the discrepancy is actually violating a design specification and second, make sure that the installer is building to the same specifications as the inspector is inspecting to. The build criteria are outlined in the work order in which the install occurs. This can be based on instructions in the work order, specifications, or drawings. The inspection criteria are outlined in two documents, one for mechanical install inspections and one for electrical install inspections. These documents are very general and are discussed further in Section 3.1.4 below.

3.1.2.2 Defect Analysis Worksheet: Section 3

The third section of the defect analysis worksheet was used to perform a root cause analysis and determine corrective actions. The tool used to perform the root cause analysis was the “Five Whys.” The

“Five Whys” tool is simply asking why a defect has occurred five times (or more) building on the previous answer until the nature of the problem becomes clear. This leads us to the “One How” or how to solve the problem. The simple example given by Taiichi Ohno, the father of the Toyota Production System is as follows:

1. *Why* did the machine stop?
There was an overload and the fuse blew.
2. *Why* was there an overload?
The bearing was not sufficiently lubricated.
3. *Why* was it not lubricated sufficiently?
The lubrication pump was not pumping sufficiently.
4. *Why* was it not pumping sufficient?
The shaft of the pump was worn and rattling.
5. *Why* was the shaft worn out?
There was no strainer attached and metal scrap got in. (1988)

In this example there could be continued questions asking “Why was there no strainer?” to get to an even deeper root cause, but for this level of causation the corrective action to take would be to replace the worn shaft and attach a strainer to prevent metal scrap from getting into the shaft. The “Five Whys” tool was chosen because it was important to delve very deep into the systemic issues rather than focusing on why a specific defect was occurring. This tool of analysis meets that criterion rather than focusing on the surface issue which can lead to firefighting rather than correcting the root cause.

3.1.3 Discussions with installers and inspectors

To gain familiarity with the installs, discussions and interviews with installers and inspectors were conducted. Every attempt was made to conduct these discussions as soon after a discrepancy was created as possible so that the inspectors and installers would be as familiar as possible with the defect or install.

First the author spoke with inspectors. If the free form text describing the discrepancy was difficult to understand or was an unfamiliar issue they were asked to further explain the issue and if possible show the issue to the author. Inspectors were also asked to describe the criteria they used to define defects. Because many types of discrepancies were common and could be grouped together and because all inspectors had the same inspection instructions, asking these types of questions only once or twice

throughout the analysis process for a certain type of discrepancy provided sufficient information on the inspection process.

Speaking with installers was more complex because each work order is different and may have different specifications attached or different indications of how work should be completed. As not to disturb the installers from their work too often, interviews were conducted with an installer after a number of discrepancies were written against a work order that installer had worked on. Additionally, as common issues arose a standard set of questions was developed. These questions can be found in the Appendix. If there was an unusual issue further questions were asked. Each discussion always began with a very broad and general question of “What is difficult about his install?” Often times this question alone prompted a discussion about many of the issues each installer faced on a daily basis and ended up answering questions that were planned in the interview already. If the installer had a specific type of discrepancy repeatedly come up on their install that type of defect was focused on in the interview.

There was one major challenge when speaking with installers. Because a discrepancy is written specifically against work that an installer has performed they sometimes would become defensive and simply place blame on the inspectors for a faulty defect report. When speaking with installers it was important to approach them with the intent of (1) wanting to learn and (2) wanting to make their work easier in the long run. Specific defects written against an installer’s work were only mentioned if absolutely necessary to understand the defect and determine the root causes. These methods could also be utilized to encourage installers and inspectors take part in improvement activities. By approaching them with the attitude of, “How do you think we can make this install or inspection easier for you and others?” helps gather ideas from the front lines, make employees feel more empowered to make a difference, and be more likely to buy into a change.

3.1.4 Gathering information from work orders and inspection instructions

Because instructions and specifications were not standardized it was important to understand the instructions, drawings, and specifications given in each work order. It was particularly important to determine whether the work order and inspection instructions contained the specification criteria that the defect did not meet. It was also important to check that all work and inspection instructions were supported by either specification or drawing requirements because those govern the design. Sometimes defects were so long-standing that even the inspector could not point out the original specification that described the requirement pertaining to a defect. Instead inspectors would base these defect reports on their prior experience including informal training and exposure to similar defects.

3.2 Systemic Root Causes

After analyzing each of the discrepancies that arose on the single aircraft followed through the final assembly line, some common root causes were found. These issues were repeatedly discussed by installers and inspectors or discovered in analysis of work orders and specifications regardless of the specific type of defect analyzed.

3.2.1 Vague and inconsistent work and inspection instructions

In viewing each work order that had discrepancies written against it, it was found that none were consistently written. There were even work orders describing the exact same install on the left and the right side of the aircraft that were not the consistent. Instructions in the actual work order were vague and unclear and required use of additional documents. For example, a work order installing grommets may have the instruction “Apply a uniform coat of adhesive to the grommet in an area .25 long along the periphery of the grommet,” however, a uniform coat does not describe the volume of adhesive required. A common defect written in regard to grommets is excess dripping adhesive. If the volume of adhesive required was controlled it is likely that there would be less dripping adhesive defects. Though in this case there was no more additional guidance on the adhesive in the drawing or specifications, in general, installers were expected to use drawings and specifications as their main installation guides.

Specifications provide additional requirements for the product that are common to many areas and installs. Drawings are very detailed and contain information on multiple variations of a design and are used over multiple work orders. Because they contain a large amount of information it is often difficult to determine which pertains to a particular work order. They also do not give specific instructions on the installation process but focus on how the installation should look when complete. In addition, drawings are physically large and difficult to bring onto the aircraft while performing the installs. Similarly, a single specification may give requirements for many different types of installations and often they were not linked to the work orders. This is discussed further in the following section 3.2.2.

Inspection instructions are also difficult to understand. There is one document for all mechanical inspections and one document for all electrical inspections. These documents are over 100 pages each and aim to address every discrepancy that could come up on every possible install. It is difficult to determine which parts of the instructions are required for which work order inspections. The instructions are so long and detailed that inspectors rarely read the instructions unless they have specific concerns.

3.2.2 Buried specifications and requirements

As mentioned above, because instructions in work orders were vague, installers were expected to use drawings and specifications to guide their work. However, while speaking with installers the author learned that many did not know how to find specifications and rarely, if ever, looked at them. Commonly, the only time an installer would try to find a specification would be to contest an inspector who had written a discrepancy against their work which the installer believed was unwarranted.

This was understandable because in looking for the build and inspection criteria it was found that often a specification relating to the work order was only referenced in another specification that may have been linked to the work order or the drawing. Installers would perform installs that meeting the drawing requirements but were unaware of the additional requirements that were listed in a specification and would not know that the install was incorrect until a discrepancy was written against their work.

Not only does this issue affect quality, it affects the speed of the build. If an installer or inspector does need to find a specification it can possibly take hours, which is wasted cost of labor. Additionally, because specifications are difficult to find, multiple specifications may contain guidance for the same type of install but the different specifications may be contradictory.

3.2.3 Lack of feedback and communication

There was multiple communication issues revealed through the root cause analysis. The first is between installers and inspectors. The free form text the inspector writes to describe the discrepancies is not standardized and sometimes installers do not understand the defects the inspector writes against their installs. Also, though locations should be called out in the defect description often times it is still difficult for the installer to find the defect. This mostly leads to reopened discrepancies. Additionally, installers and inspectors do not really discuss defects unless the installer is contesting a discrepancy written against their work. In this case there is not much knowledge transfer between parties because they are solely focused on determining who is correct.

The second communication issue is that if an installer or inspector makes a suggestion for change to improve a process it is not followed up upon. The suggestion seems to be lost even though it is being worked on by Engineering. The timeframe to make changes is very long and there is not enough feedback as work is being done. Because of this installers do not make suggestions for change but instead keep doing things the same way even if they cause difficulties with the install or increase the number of discrepancies.

The third communication issue is that managers and supervisors do not know what the highest driving discrepancies are. This is because they do not have access to clear and concise data. Without data it is difficult to see patterns in what discrepancies are written and to take corrective actions on specific types of discrepancies. They may get a general sense of which are discrepancies are occurring but because X Aerospace is a highly data driven company they will not take action unless data supports that

there is a problem. Though there is a tool that can display this data, the word Pareto chart created last year, only Quality has access to the tool. Managers and supervisors are extremely busy and need to stay on or near the floor as much as possible so going to ask a quality engineer to gather this data and waiting for them to organize it into a usable format is not a plausible expectation.

3.2.4 Late discovery of discrepancies

An interesting discovery during the analysis of reworkable discrepancies was that about 50% of discrepancies were written against a work order other than one in which the discrepancy was created.

If a discrepancy is not found on initial inspection of the work order in which it

was created it is found, instead, during a later work order inspection or in a shake inspection. Ideally defects pertaining to a work order should all be found and reported on inspection of that work order.

Defects found outside the work order where they are created is a symptom of a number of problems including the following:

- Inspectors do not find the defects on their first inspection of an area.
- Different inspectors have different criteria for discrepancies.
- Discrepancies are created after the initial work is inspected due to others working in the same area.

When discrepancies are found late it makes it more difficult to determine the actual high driving work orders. Additionally, it is more difficult to correct the discrepancy because it may be harder to access the install area and an installer who is not familiar with the original install may be asked to do the repair.

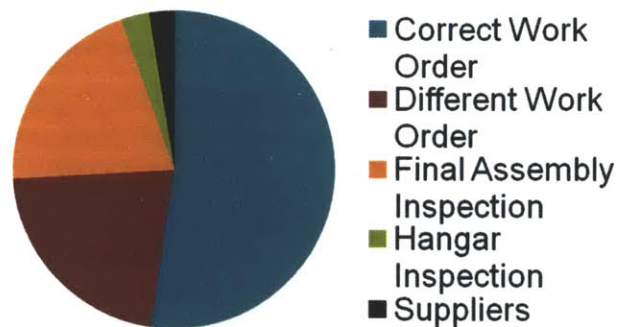


Figure 7: Discrepancies written outside their build work order on Effectivity 17

3.2.5 Lack of training and awareness

Almost every installer interviewed felt that they did not have adequate training before beginning their work. They also indicated that they did not feel like they had mentors or people they could ask if they had questions about an install. Many said it was easier to try the install and have a discrepancy written than to find someone to give them guidance on the install. There is a lot of on the job learning and with on the job learning there are on the job mistakes. Though, training is available if an employee requests it, it seemed as if employees would have benefited from more pre-job training and mentorship.

3.3 Corrective Actions

To address these root causes the following corrective actions were developed. Because of the time frame available only two were addressed during the internship project. The corrective actions chosen were chosen because their scope could be addressed during the internship time frame and because they could at least partially address multiple root causes. The key for all the corrective actions given is to focus on the highest driving or most commonly occurring issues.

3.3.1 Vague and inconsistent work and inspection instructions

In order to address the vague and inconsistent work and inspection instructions the suggested corrective action is to standardize work and inspection instructions. Work instructions should have a standard layout and vernacular and more process instruction in the actual work order. Rather than requiring installers to look at specifications and drawings to fully understand the install, the specifications and drawings should supplement the work instructions and be easy to find and access. All instructions should be supplemented by simple images and photos rather than having only a complex drawing to refer to. While developing the work instructions engineers should work with installers and inspectors from the beginning of development to determine what is unclear about the current instructions and to help ensure new verbiage is clear.

Inspection instructions should be handled similarly to work orders with specific instructions and guidance relating only to the work order that is being inspected. Work order instructions and inspection instructions should draw clear parallels. Standard inspection instructions were one of the corrective actions addressed during the internship project and are further discussed in chapter 4.

3.3.2 Buried specifications and requirements

As part of the work order and inspection order updates the specifications required to complete an install or inspection should be referenced and linked immediately after the instruction based on the specification. All instructions should be worded to match the specification (Direct excerpts from the specifications would be best.) and have a drawing or specification requirement attached to them. If a requirement is referenced in multiple specifications both should be included but the superseding specification should be indicated.

3.3.3 Lack of feedback and communication

In order to address the lack of communication between installers and inspectors it would be useful to include both in any discussions on installations. Though, this may be challenging at first due to some of the cultural elements discussed in chapter 2, surprisingly, in one awareness training where both installers and inspectors were present there was constructive discussion between the two groups to aid each in understanding of the specifications being discussed. Discussion and questions on the floor should be encouraged and installers should be paired with an inspector so they feel there is someone they can speak with candidly on inspection questions. Additionally, to help installers understand discrepancies inspectors should use a standard format for reporting. This standard format would be described in the inspection instructions.

To address the lack of feedback for continuous improvement suggestions a tracking tool for reported suggestions is recommended. X Aerospace has already begun the IT development of such an online tool. The challenge, however, with such a tool will be to build employees' trust of the tool.

Because of the past failures it will be difficult to get employees to report ideas. Additionally, because of the long turnaround time for implementation, it will need to be ensured that even between large milestones for an engineering change, progress is reported in the system. Setting step goals to achieve small wins can show progress more easily. These types of small goals could be shown in the tool which would help gain employee trust of the tool and build morale with the small wins. Communication between engineers and installers through the system or offline will need to be encouraged and rewarded.

Finally, if there are improvements or degradation in the number of discrepancies managers and supervisors have no way of knowing that they have occurred because they do not know where to go to get this data. The solution to this is to make the data easily accessible to managers and supervisors through use of very simple tool which allows them to see not only their highest drivers but how they are changing over time. This corrective action was followed through with during the internship project and its development is further discussed in Section 5.

3.3.4 Late discovery of discrepancies

The late discovery of discrepancies is a problem because it becomes not only more difficult to repair them in the short term but more difficult to take proper corrective actions to reduce these defects in the future.

Defects that occur due to damage after the original install is completed occur when other work orders are completed in the same area. Often, mechanical installs that occur after electrical installs require that the electrical components be moved to perform the mechanical install. These can inadvertently create damage to the already complete and inspected electrical install. To address this, the sequence of installs should be carefully considered or mechanical and electrical installs should happen in parallel so repairs to the electrical install can be made as steps in the mechanical install are completed then, both work orders inspected simultaneously. These defects can also be created as accidents that occur as part of normal working conditions. For example, someone stepping on a part and breaking it.

Encouragement of self-reporting of damage would help mitigate this issue. Regular use of protective covers on areas of the aircraft susceptible to damage helps prevent it. X Aerospace has already taken on an initiative to audit, measure the regularity of use, and give metrics for such protective covers. This, however, was outside of the scope of the internship project.

The other issues that lead to late discovery of defects have to do with the inspectors not having clear instructions for inspections. Because inspections are not fully standardized some inspectors may miss defects others will find later or may have different definitions of what constitutes a defect. As described briefly above in Section 3.3.1 and in detail in chapter 4, standardized, concise, but complete inspection instructions will ensure that all inspectors have the same criteria for inspection and aid them in finding all defects in their first inspection.

3.3.5 Lack of training and awareness

Installers use discrepancy reports as a learning tool rather than having the proper training in the beginning which increases the number of discrepancies. As discussed above, when installers do not know about a specification the only way they learn the specification exists is when a discrepancy is written against an install that does not meet the criteria laid out in the specification. To combat this issue when an installer or inspector is hired or switches to a new install they should be trained on the install and taught how to properly use work orders, drawings, and specifications. Additionally, each installer and inspector should have a mentor who has experience with their install in case they have questions on the install or need someone to confirm the quality of their work before it is submitted for inspection. While the installers do have supervisors the supervisors play more of an administrative role, making sure every task is complete, that the aircraft moves positions on time, and reporting to upper management. The supervisors do not serve as mentors or provide training to installers.

Not only should training be conducted when an installer or inspector begins a new job, there should be continued awareness of high driving issues. These awareness sessions can be used in conjunction with

the tool displaying high drivers. Last year, as part of the project to reduce specific defect types, awareness training packages were created for some of the most common discrepancies. Packages should be created for all the highest drivers and as supervisors and managers see these types of defects rise they can have an awareness session to remind installers and inspectors of the proper installation techniques.

Building on the corrective action for improving communication all training should be conducted for both installers and inspectors at the same time. In this way they will both be getting the exact same criteria for install and inspection. It also allows for discussion between the groups if there are differences in understanding of a specification. Finally, in these awareness sessions installers and inspectors should be able to voice their concerns to managers about the installs or the instructions and suggest changes. A neutral or well respected moderator could help make sure that the suggestions are constructive and the session does not become simply one of unhelpful complaints.

In order to determine the systemic root causes of reworkable discrepancies all the discrepancies that occurred during one aircraft build were analyzed and the common root causes determined. Data on each defect was collected from the computer work order system using a database query. Information from work order instructions, drawings, specifications, and installer and inspector interviews were used to perform a root cause analysis with the “Five Whys” method. A number of systemic root causes were discovered and ideas for corrective actions were developed. Due to time constraints only two corrective actions were taken. The first corrective action was to improve instructions. This addressed two of the root causes found during the analysis, vague and inconsistent work instructions and the difficulty of finding the specifications related to a work instruction. The second corrective action taken was to address the lack of communication of discrepancy data to managers and supervisors through the use of a simple discrepancy display tool. These corrective actions will be discussed in detail in the chapters that follow.

4 Standard Work for Writing Inspection Instructions

One of the corrective actions that was addressed as part of the internship project was to create better more standardized inspection instructions. Inspection instructions were chosen as the focus because while work orders were currently controlled and written by Engineering there were no specific or standard inspection instructions. Because in the scope and timeline of the internship project it would have been impossible to write instructions for inspection for every work order, instead a standard process and template for writing inspection instructions was developed. This chapter will first examine the current process for performing inspections. Second, it will report best practices and techniques for improving inspection. Next, the development of improved inspection instructions and why improved instructions were chosen for this project will be discussed. Finally, the chapter will close with the current state of the implementation of the instructions and future opportunities for improvement.

4.1 Current Process

Currently, after a work order is completed and bought off by an installer, soon thereafter an inspector checks the installer's work. The only instructions the inspector has are that the work must meet work order instructions, drawing requirements and specifications. There is a document for each mechanical inspections and electrical inspections to help guide the inspector to common discrepancies, but they are vague, unclear, and unfocused. The same guides are used for all work order inspections so the guide has a large amount of extraneous information. They also do not contain guidance on where to find the original specification from which the requirements defined in the inspection instructions came.

Because the current inspection instructions are so long, inspectors do not use them regularly in their everyday inspections but only as a reference if a difficulty arises. This is a problem because instructions may change if a specification changes. However, sometimes if a specification changes the instructions are not even accurately updated to match the specification change.

4.2 Literature Review of Inspection Improvement Techniques

Research studies have been conducted on different methods to improve inspection and on how to improve instructions. Some of these methods include allocation of inspections to the best times during the manufacturing process (Viswanadham, 1996, Lee, 1998), virtual and on-aircraft inspection training (Gramopadhye, 1998; Nickles, 2003), wearable virtual inspection task guidance systems (Ockerman, 1998; Ockerman, 1999), and improvement of inspection instructions (Patel, 1994).

4.2.1 Inspection allocation

Lee and Unnikrisnan present a method of allocating inspections only to the times in production where it is most important to find defects. They argue that limiting inspection to only these times will maximize chance for finding defects while minimizing the inspection costs. To determine the appropriate locations they use a non-linear mixed integer optimization program. Their program trades off cost of additional inspections at an earlier stage with cost of finding defects at a later stage in the manufacturing process. (1998)

Currently, at X Aerospace inspections occur after each work order. Reducing the number of inspections could significantly reduce inspection cost and help prevent new defects from being found on areas that have already been inspected as described in Section 3.2.4 above. However, this method of inspection improvement was not chosen because of the large number of defects found at each inspection and X Aerospace's commitment to highest quality. If in the future defects are reduced enough that there are clear times in the manufacturing process with higher defect rates this type of optimization could be successful.

4.2.2 Inspection training

Most inspection at X Aerospace on the final assembly line is visual. This means that inspectors are visually scanning an area until they see something that does not conform to their expectation of what meets a specification. When they notice something is awry they further investigate the issue, most of the time still visually, but occasionally using a measurement tool if necessary. Nickles et. al describes this as

“a series of eye fixations followed by saccades, or brief eye movements, from one fixation area to another.” (2003) Individual characteristics such as age, gender, visual acuity, personality and experience all have an impact on how adept an inspector is at discovering defects. If this search behavior was perfect each inspector would scan the area fixating on each spot only once and discovering every defect on first scan. However, this process is far from perfect, but techniques can be applied to help improve it. Visual inspection training has been shown to significantly improve parameters quantifying eye movement and, more importantly, improve the accuracy and speed of detection of defects. (Nickles, 2003)

Many training techniques are proposed but one such training technique is to use a computer program to display an “inspection area” some of which have “defects.” The inspector then reports and categorizes the defect. Their performance is measured on the number of defects correctly identified, the speed of identification, the time looking at each “inspection area,” and the number of false defects detected. By repeated use of this type of program the inspector becomes aware of defects more easily. (Gramopadhye, 1997)

This method was not chosen to be implemented at X Aerospace because of the large number of types of defects and the complexity of building such a training program.

4.2.3 Wearable virtual task guidance systems

One technology that has been in development is a wearable computer that can guide an inspector as they are performing inspections. A wearable computer is connected to a wireless network and places a visual display in front of the eyes to project over reality. These computers can usually be voice controlled. They make it possible to access large amounts of data in the form of text, graphics, videos, animations, and sounds without ever leaving the work area or the need to carry cumbersome computers or documents. In this way all steps of procedures and specification can always be available to the inspector. They can also help the inspector while scanning an area for defects by helping to divide it into a grid. (Ockerman, 1998)

The problem with wearable computers and the high level of task guidance they provide is that they create an overreliance on this guidance. Inspectors will only check for things they are specifically told to and may miss other defects, however blatant, because they are not specifically looking for them. (Ockerman, 1999) For this reason, as well as the lack of appropriate hardware and software currently available for this purpose, this method of inspection improvement was not chosen for X Aerospace.

4.2.4 Improvement of inspection instruction

Patel et.al have developed a methodology for the design of useful work cards (instructions) for inspection. They conducted a study using instructions developed following their methodology which asked inspectors to rate the new design as compared to the work instructions they were currently using on 14 dimensions. 13 of the 14 dimensions saw significant improvements.

There are four important issues in design of instruction documentation.

1. Information readability – This includes dimensions such as typographic layout, word and sentence structure.
2. Information content – This includes both textual and graphical information and the origin, appropriateness, accuracy, completeness, and understandability of each.
3. Information organization – This includes classification and layering of information into categories, as well as the form of the instructions which should include a command verb, an action qualifier, and an object.
4. Physical handling and environmental factors – This includes the form of the physical instruction document.

In their paper Patel et. al describe in detail how to meet these four requirements in writing instructions. These guidelines can be found for reference in the appendix.

Improving inspection instructions was chosen as the method by which X Aerospace would improve its inspection processes. Not only was this an area quality engineers could influence but there was not a

strong existing structure in place for inspection instructions. Additionally, in the future similar techniques for improving instructions could be applied to work instructions to improve the whole system and not just the inspection side. Many of the guidelines suggested by Patel et. al were considered in the design of the new instructions for X Aerospace.

4.3 New Standard Work for Writing Instructions

In order to develop standard work for writing inspection instructions, instructions were written for two of the highest driving work orders (that were not shake inspections) on each final assembly line. The work order, drawing prints, and specifications in relation to the install were carefully analyzed to determine what areas of inspection were required. The intent of writing this standard work is to allow X Aerospace to easily write inspection instructions for each work order in final assembly instead of forcing inspectors to use an extremely long, general inspection guidance document. The design of the inspection instructions was based on the guidelines laid out by Patel et. al.

The inspection instructions include the following features:

- Tasks are divided into guided and general inspections
- Each inspection instruction references a specification, drawing call out, or work order instruction
- Related drawing views, sheets, and aircraft location are listed
- Guided tasks have both aircraft and drawing location information
- Only related inspection information is included
- Inspections are written in directive language and identified with a → symbol
- Guided tasks have check boxes next to them to ensure completion
- A grid is laid out for breaking up general inspections
- Information is layered to make instructions easy for use by novice and experienced inspectors

These features and their purpose are described below and can be seen in more detail in the example inspection instruction template in the appendix.

Because task guidance can lead to inspectors missing defects other than those listed in the specific instructions but is important for guiding the inspector to critical inspections, the instructions were broken up into a guided task section where specific installs called out in the work order or drawing are given explicit inspection instructions and a general inspection section which contains instructions on inspecting for many other defects which may occur within the install. For an example of a guided task, if the work order says “Plug jack A into receptacle A” in the guided tasks section there would be an instruction “→ Verify jack A is secured to receptacle A.” Also included in this specific instruction would be guidance on what “secure” means based on specifications. All guided tasks must be checked off by the inspector to buy off the inspection. For an example of a general inspection, all installs require that the installer ensure that any area they work on is properly painted and touched up if necessary. The general inspection instruction would be “→ Verify that wherever paint has been removed the area has been touched up to match the original finish.” Since general inspections are more of a reminder to the inspector to check while they are scanning the area for defects and multiple defects of the general type can arise they do not have individual check boxes to ensure sign off. The general inspections section does, however, break the area of the aircraft covered by the work order down into a grid. This way the inspector can scan a small area and move to the next. They would then check off each area of the grid.

Currently if an inspector needs to find a specification to determine if an install has a defect they must look through the one-hundred-plus page inspection instructions to see if the general defect is called out there. If it is, they must determine which specification the instruction references, look up the specification, sometimes be referenced to another specification from the first one, and read and understand the specification before confirming a defect. Because this process is long and grueling most times inspectors just go with their experience which is often correct instead of actually looking up the specification. Experience can be based on informal training, exposure to similar defects, and following

the inspection guidance document which is meant to guide inspection but not be the design specification. By placing the reference to the specification directly in the inspection instructions and using the exact or very similar wording of the specification these steps are highly simplified. The inspector will now rarely need to look up the specification and if they do they know exactly which specification to check. There is also a list of all specifications that have to do with the work order being inspected in the beginning of the inspection instruction document.

Drawings used for installations are also large and cumbersome. Each work order may have multiple drawings and each drawing contain multiple views and multiple versions of an installation. It is very easy to accidentally look at the wrong part of the drawing. To mitigate this, the inspection instructions contain a chart indicating all drawing views utilized in the installation, the sheet of the drawing where they are located, the grid location on the sheet, other views that the given view references, and location on the aircraft the view encompasses. Additionally, on all guided tasks the drawing view and aircraft location should be referenced.

As, mentioned above, the current inspection instructions are not specific to each work order and therefore contain guidance on every type of defect that may occur. If an inspector is checking a work order and they need guidance they must look through a very long document. The new inspections orders contain more information and guidance but only in regard to the work order being inspected. There is not extraneous information to sift through. In addition, the information is more organized. There is a table of contents and a guide to using the document at the beginning of the instructions. All the guided tasks and general inspections are listed in the table of contents to make each particular inspection instruction easy to find. Another simple feature that improves usability of the document is that if possible each section or specific inspection instruction was kept together on one page rather than splitting it between pages.

One key goal that the new inspection instructions needed to meet was to be useful for both brand new inspectors and experienced inspectors. To meet this goal the instructions are layered. For

experienced inspectors the headings of each of the guided tasks and general inspections can serve as reminder of what inspections need to be performed. Other information is provided below if they require it. Additionally, many inspections instructions contain reference information if it is required by the inspector. Reference information is easily distinguished from inspection instructions because only inspection instructions have the → symbol in front of them and are written in directive language. (“Verify that X is Y”). A new inspector can use all the information provided step by step to instruct them specifically and in a high level of detail on how to perform the instruction. More experienced inspectors can use the high level information with the rest of it as reference.

4.4 The Case for New Inspection Instructions

The new inspection instructions have the potential to save X Aerospace a large sum of money each year. Based on the fact that inspectors currently spend a portion of their time looking for specifications and that they often miss defects on first inspection of an area an estimated cost reduction was calculated. By taking even a small percentage off the number of defects and the time spent on inspections a significant savings was found. Even with as little as a one percent reduction in defects per aircraft and a five percent reduction in inspection time there would be a savings of close to a million dollars per year.

In order to write the instructions for each work order contractors would be hired solely for this purpose. Contractors are a much less expensive option than using current manufacturing or quality engineers. Since standard work for writing these inspection instructions was developed contractors could easily follow the standard work to write the instructions. Additionally, because it would be difficult for current engineers to put their full time into the project the time to write the instructions would be too great. Based on writing the sample instructions it was estimated that twelve hours would be required to write each instruction. Based on this estimate and the cost of contract labor the investment cost and time to return on investment was also calculated. The project would have a relatively low fixed cost and maintenance cost each year compared to the savings it would bring and would have a return period of as little as seven months.

In addition to the obvious monetary benefits outlined above the inspection instructions could have secondary benefits of reducing the amount of labor hours required to repair a defect, improving communication between installers and inspectors (because of the instruction on how to write up defects), improving understanding of specifications, reducing defects found late, and improvement of work instructions, discussed below.

4.5 Current Progress of the Inspection Instruction Project

At the time of the completion of the internship the standard work for writing inspection instructions had been presented to management but had not been implemented. A budget for hiring contract employees to write the instructions for all work orders had been approved. However, a change in management has stalled the program. Instead, the development of better inspection instructions sparked the conversation on the need for better work instructions in general. Many of the techniques found in the literature review, used in the development of the inspection instructions, and presented to management are now being used in the development of improved work instructions. This is in conjunction with the roll out of an improved computer system for presenting work orders and installation and inspection buyoffs. Additionally, while inspection instructions are not being created for each work order as intended with the standard work, the need for including the references to specifications in the instructions became clear. Specifications are now being added to the current inspection guidance document. Both these improvements are being implemented by Engineering since they are responsible for writing work instructions and design specifications.

4.6 Future Use of Technology

The use of an improved computer system is the first step in using improved technology to improve inspection and work order instructions. With this implementation of the new system tablet computers could easily be used to guide installations and inspections. One idea is to use the camera on the tablet to have installers take a photograph of the operation that they have just completed and to use that as the buy

off. Then, when the inspector is checking the area if they find a defect they can simply write a description of the defect, then circle the area in the photo where the defect has occurred making it much easier for installers to go back and repair any defects. This would also help determine if the defect was created by the installer on the work order or if it was damaged after the install was complete, which can help determine the best order for installs to occur or whether a protective covering is needed in an area.

4.7 Future Use of Improved Instructions

While currently work instructions are being rewritten to meet some of the guidelines for good work instructions mentioned above, standard work for writing work instructions is still not being implemented. By using a standard work document X Aerospace would ensure that all work instructions contain the same information and can be easily followed by any employee because they will be familiar with the layout. Standard work for writing work instructions could be very similar to the standard work for writing inspection instructions. In fact, the standard work for writing inspection instructions could serve as a baseline for writing standard work for installation instructions. The main difference would be that in the guided tasks section instead of verifying tasks are complete to a certain drawing print, specification, or work instructions the installer would actually have to complete the work and sign off on each operation as complete.

A further step for use of standard written inspection instruction would be in the case of shake inspections. Shake inspections currently use the same long general document that work order inspections use for guidance. However, shake inspections are different from work order inspections in that they are more of a scan and inspectors cannot look at all areas because some are hidden by other inspections. Even though shakes are more general inspections, if there are areas that commonly have defects, these could be placed in a guided tasks like section while others are placed in a general inspections like section.

To address the poor clarity of instructions the corrective action of creating standard work for writing inspection instructions was taken. Though both work and inspection instructions require improvement

inspection instructions were chosen because there were no specific or standard inspection instructions other than a general inspection document. This document is long and hard to follow so it is often not even used by inspectors during their everyday inspections. Additionally, due to the time scope of the project, instead of writing inspection instructions for each specific work order, standard work that could be continuously used to instruct employees on how to write inspection instructions for each work order were created. To determine how best to improve inspection instructions a literature review was conducted to find best practices. Some of these features implemented in the new style of inspection instructions were the splitting of tasks into guided and general inspections, the inclusion of references to related specifications, drawing views and aircraft locations, the limiting of information to only that which is pertinent to the particular work order inspection, the use of symbols and directive language to clearly indicate inspections versus reference information, and the layering of information to make instructions useful for both experienced and novice inspectors. Due to changes in management this style of inspection instructions is not currently being implemented. However, many of the features and techniques for writing better instructions laid out in the standard work are being utilized by Engineering to improve work order instructions. Additionally, the technology for presenting and reporting installations is being improved. In the future, with the new system, installations and inspections could be improved further. For example, tablet computers could be used to keep all instructions immediately at hand and to photograph completed installations and defects.

5 Data Distribution Tool

The other corrective action taken to mitigate one of the root causes found in the defect analysis was to create a tool that would easily collect and display discrepancy data for supervisors and managers on the floor. This would help solve the communication gap between Quality and Manufacturing by allowing supervisors and managers to be part of the defect reduction effort.

This chapter will cover in detail, the development of the data distribution tool. First, the need for the tool will be discussed. Next, the chapter will speak to the steps for both interface and technical development. Finally, implementation and current status of the tool will be presented.

5.1 The Shift in How X Aerospace Employees Think About Quality Improvement

Last year when X Aerospace chose to focus on reduction of reworkable discrepancies as a key quality and cost improvement area the project was viewed solely as Quality's responsibility. Because of this Quality was collecting the data, determining how to organize and display the data in a useful manner, determining how best to use the data and to determine root causes of high driving types of discrepancies, and forming teams to perform corrective actions to mitigate the root causes. Though the teams brought in people from Engineering and Manufacturing, these functions were not fully vested in the project because the project was Quality's responsibility. Though all groups were aware of the cost of discrepancies and were presented with the data showing the highest driving discrepancies through the use of the cross Pareto heat map they did not believe it was their problem to solve because Quality had taken on the project. So, while initially appearing willing to help, enthusiasm of the other functions soon decreased.

As this reduction effort began to lose drive, management started to think about how to keep momentum on the defect reduction project going. They believed that instead of making Quality the sole function responsible for the reduction effort, it would be better to require all functions to have a part in the effort. While Quality has the data on reworkable discrepancies and are the best function to take on root cause analysis, they are not the best function to implement corrective actions. Manufacturing and

Engineering are much closer to the issues on the floor and have more resources to effect change and ensure that change is sustained. In addition, if Manufacturing or Engineering finds a root cause or implements a corrective action they will be more vested in making sure it succeeds.

5.2 The Need for a Data Distribution Tool

Though last year an Access Macro (Cross-Pareto tool) was created that had the ability to categorize and organize discrepancy data by types of defects, the piece of the puzzle that was missing for Engineering and Manufacturing to take on more responsibility in reduction of reworkable discrepancies was the access to the Cross-Pareto tool. Supervisors and managers who are closest to the issues on the floor need to be able to see data supporting what they see and hear on the floor in regards to rework. X Aerospace is a very data driven culture. Changes do not occur unless there is data supporting that a change is required. The need for tool that allowed supervisors, managers, and others see discrepancy data was clear but it was unclear as to the form this data should be presented.

One of the line managers had come to Quality and asked for data on their line's discrepancies so that they could see what and where their highest driving discrepancies were. Since Quality had already developed the cross Pareto heat map in the year prior and had had success with using it to determine high drivers this became the clear choice for the display of data. Figure 8 shows an example of this type of cross Pareto heat map

Part/Mode	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7	Mode 8	Mode 9	Mode 10	Mode 11	Mode 12	Mode 13	Mode 14	Mode 15	Total
Part 1	8.89%	0.56%	1.57%	0.11%	4.39%	2.02%	0.90%	0.11%	0.11%	0.45%	0.00%	0.00%	0.00%	0.11%	0.11%	20.81%
Part 2	0.22%	2.14%	0.90%	0.11%	1.01%	1.69%	1.35%	0.00%	1.80%	1.35%	0.56%	0.00%	0.67%	0.00%	0.00%	14.85%
Part 3	0.34%	5.62%	0.79%	0.79%	0.67%	0.90%	0.67%	0.00%	0.00%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	12.15%
Part 4	0.00%	1.46%	0.56%	1.57%	0.22%	0.34%	2.47%	0.00%	0.00%	0.00%	0.45%	0.22%	0.00%	0.00%	0.00%	8.89%
Part 5	2.81%	0.34%	0.34%	0.22%	2.14%	1.57%	0.45%	0.00%	0.34%	0.00%	0.00%	0.00%	0.67%	0.00%	0.00%	8.66%
Part 6	0.00%	0.00%	3.94%	1.01%	0.00%	0.00%	0.11%	1.24%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.11%	8.66%
Part 7	2.25%	0.34%	0.22%	0.11%	1.80%	1.35%	0.11%	0.00%	0.22%	0.00%	0.00%	0.00%	0.00%	0.00%	0.11%	6.64%
Part 8	0.11%	4.72%	0.22%	0.00%	0.00%	0.56%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	5.85%
Part 9	0.00%	0.00%	0.56%	0.00%	0.00%	0.00%	0.22%	0.11%	0.00%	0.00%	1.24%	0.00%	0.00%	0.00%	0.00%	4.50%
Part 10	0.00%	0.45%	0.45%	0.67%	0.11%	0.11%	0.56%	0.00%	0.11%	0.00%	0.11%	0.00%	0.00%	0.00%	0.00%	3.15%
Part 11	0.22%	0.11%	0.34%	0.34%	0.45%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.92%
Part 12	0.00%	0.34%	0.34%	0.00%	0.34%	0.11%	0.11%	0.79%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	2.70%
Part 13	0.22%	0.00%	0.22%	0.56%	0.11%	0.34%	0.11%	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.00%	2.36%
Part 14	0.00%	0.00%	0.67%	0.00%	0.00%	0.00%	0.22%	0.00%	0.00%	0.00%	0.22%	0.00%	0.00%	0.00%	0.00%	2.02%
Part 15	0.11%	0.11%	0.22%	0.00%	0.56%	0.11%	0.34%	0.00%	0.00%	0.00%	0.00%	0.22%	0.00%	0.00%	0.00%	1.69%
Part 16	0.00%	0.11%	0.11%	0.00%	0.11%	0.00%	0.11%	0.00%	0.00%	0.00%	0.00%	0.22%	0.00%	0.00%	0.00%	1.57%
Part 17	0.00%	0.11%	0.45%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.46%
Part 18	0.00%	0.11%	0.11%	0.79%	0.11%	0.11%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.35%
Part 19	0.00%	0.00%	0.34%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	1.12%
Part 20	0.22%	0.00%	0.22%	0.00%	0.34%	0.22%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.90%
Part 21	0.00%	0.00%	0.45%	0.00%	0.45%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.00%	0.90%
Part 22	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.90%
Part 23	0.00%	0.00%	0.11%	0.00%	0.11%	0.00%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.56%
Part 24	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.11%	0.00%	0.00%	0.00%	0.00%	0.56%
Part 25	0.00%	0.00%	0.11%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.45%
Total	14.62%	11.81%	11.70%	11.25%	9.67%	6.64%	5.29%	2.47%	1.80%	1.35%	1.35%	0.79%	0.67%	0.67%	0.11%	1.00%

Figure 8: Cross Pareto Heat Map Example

Collecting this data was a simple but time consuming task. Data was collected using Hyperion/Brio database query software, imported into an Excel file, and analyzed using a Visual Basic Macro in Microsoft Access. The Access Macro to analyze the discrepancy data, categorize it, and create the cross Pareto table was developed in the year prior. The results of the Access analysis were then copied back into Excel and formatted to show the coloring of the heat map and sorted from highest drivers in the upper left hand corner to lowest drivers in the lower right hand corner. To do this for one set of data, one work order, one aircraft, one position, one line or all lines was a quick job for a quality engineer. However, if data was to be split into sets such as by multiple work orders, multiple positions, or multiple aircraft the task became exponentially large.

The data the line manager had asked for was to be split by line position and top five work orders with the most discrepancies (highest driving) in each position for the last twenty completed aircraft. Depending on the line this could be over fifty cross Pareto heat maps. Showing the top five work orders allowed the highest driving problem installs to be the focus. Based on the Pareto principle, by focusing on high drivers where most of the defects resided, larger reductions could be made than by focusing on many lower driving work orders. To create this type of analysis would take up to or even over eight hours per line to follow the steps above for each position and top five work orders in each position.

Another issue with presenting the cross Pareto heat maps was that while the heat maps gave general information on the most common types of defects it was difficult to look at the data to get the details of those defects. There needed to be an easy way to display the data for each type of defect so that managers and supervisors could gain a better understanding of where the issues were occurring, what installers and inspectors were working on the defects, and what the full text of the discrepancies was.

Finally, there was no simple way to collect data on all defects and defects split by type as separated by time frame. Seeing how the data changed with time was a necessary feature. This would allow managers and supervisors to determine if issues were recurring or if there was a change in

occurrence rate and if there was improvement or degradation in the number of discrepancies. Since the heat map displayed totals and defects by type, being able to look at the heat map in a certain time frame alone and easily be able to switch between time frames would help show the changes in a quick, intuitive, visual representation. Scrolling through time period to time period it would be easy to see if different defects became “hot” in each period.

Another requirement to note is the use of a standard of having no more than five steps to use a tool of this sort. So for the tool the goal was to simply have the user (1) choose the line of interest, (2) enter the numbers of the aircraft of interest, (3) push the start button, (4) wait for the tool to run and (5) be able to use the tool to look at relevant data.

5.3 Stages of Development of the Tool

Based on the needs above the data display tool was developed with certain features. The first step was to determine how best to organize and display the data. The line manager’s needs and request drove much of the development. Because the raw data pulled by the query and the heat maps created by the Microsoft Access Macro were displayed in a table, Excel was the clear choice for displaying the required data. The manager had asked for the heat map for the five work orders with the most discrepancies written in each line position for the past twenty aircraft that had been completed. They wanted to be able to view data at different levels, top level (all positions and work orders), position, and work order.

When heat map discrepancy data first became an interest it was expected that the quality engineers would collect and organize this data. This is why initially the creation of the heat map display tool was completely manual. Preparing the data manually was found to take a full day of work and was very repetitive so even for a quality engineer to prepare the data as little as every few months would be difficult. This led to the second iteration which automated much of process and allowed the tool to be created in the background as the quality engineer was working on other tasks. However, it did require some knowledge of how to collect and lay out the raw data to be input into the tool. Managers and

supervisors were not familiar enough with the data collection process, nor did they have the time to do this themselves. To really allow the tool to be useful to them the tool would have to be able to be created, essentially, with the push of a button. This led to the third iteration of the tool which fully automated all data collection and formatting. Finally, throughout the creation process, speaking with managers and supervisors to determine what features would be useful to them, two major additional features were added. The first was a map which showed the location of defects on the aircraft and the second was ability to scroll through the data aircraft by aircraft to determine if any major changes in discrepancy had occurred between aircraft.

5.3.1 The first iteration: a manually made data display tool

The first step in designing the data display tool was creating its basic layout. The data was displayed in individual tabs for the top level data containing data from all work orders in all the line positions, each line position containing the data from all work orders from that position, and finally, each of the top five work orders alone. Each tab contained the heat map for the related data and the detailed discrepancy data below the heat map so it was all easily accessible. This formed the basic layout of the tool.

To create this tool was very time consuming. A query for all the discrepancies written for the past twenty completed aircraft was performed and the data copied to Excel. This data was then imported to the Microsoft Access Macro which categorized the data and created the organized data table and the heat map table. The heat map table and the organized data table were then both copied back into Excel to a worksheet designated for the “Top Level” data. The heat map table was then sorted by totals of each defect mode (horizontally by columns) and totals of each part type (vertically by rows) to have the highest discrepancy counts in the upper left hand corner of the table. Conditional formatting was then applied to the heat map table to create the heat map coloring. The original raw data was then filtered to only the discrepancies from each position. The steps above were repeated to organize the data and create the heat maps for each position. Next, a pivot table was used on the initial raw data to find the top five highest

driving work orders in each position. Once these work orders were determined the raw data was filtered to each of these work orders and the organized data table and heat maps created for each.

The manual tool also contained hyperlinks on each page to related worksheets. The top level sheet contained links to all positions, and all work orders under each position, each position contained a link to the top level and links to each of the work orders in that position and each work order contained a link to top level and to its position. This allowed for easy navigation between worksheets.

The final feature that was included in the initial manually built tool was the ability to double click on a cell in the heat map and be taken to the organized data filtered to match that cell. For example, double clicking on the cell for the cross section of mode of failure 1 and part 1 would filter the organized data to show only those defects that have words relating to mode 1 and part 1.

Because all the steps for collecting this data and formatting it for display were very repetitive it was clear that this process would be well suited to be performed programmatically.

5.3.2 The second iteration: a semi-automated display tool

The next iteration of the tool automated the creation of the heat maps and the organized data, formatting, hyperlinks, and filtering function. This iteration required that the data for the full set of 20 aircraft and all positions and work orders be queried and stored in an excel file which was referenced by the tool and automatically pulled into Access and copied back into the formatted Excel file with the data. The automation followed almost the exact step that the quality engineer would have taken in creating the heat map tool manually.

The automation allowed for heat maps to be created for every work order that had discrepancies rather than only the top five high driving work orders in each position. The same layout was still used as in the first iteration. Again there were worksheets for top level, each position, and each work order and on each worksheet there was a cross Pareto heat map, hyperlinks to related sheets, the organized data, and

the ability to filter the data by clicking a cell in the heat map. One feature that was added to the layout in this iteration was the ability to use a button to move easily between the heat map and the organized data.

While this iteration allowed the quality engineer to create the tool with much less active time, they still needed to be familiar with using the query program (Hyperion/Brio) and with using Visual Basic Macros in Excel. It was still not simplified enough for any employee to use. Additionally, the tool still took many hours to run and needed further technical improvement.

5.3.3 The third iteration: a fully automated tool

The third iteration focused on simplifying the steps required by the user to create the heat map tool. The user would simply choose the line and enter the effectivities (aircraft numbers) for which they

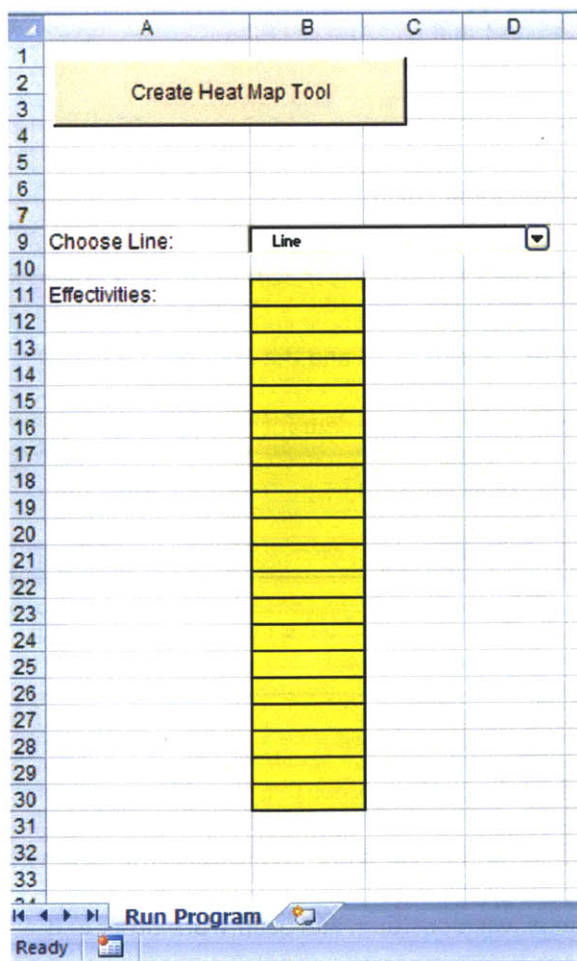


Figure 9: The user interface to create the heat map tool

wanted to display data. This interface from here on will be referred to as the heat map creation program (as opposed to the heat map tool that contains the formatted data and heat maps). See Figure 9 for the heat map creation program interface. The user could enter up to twenty effectivities but was only required to enter at least one. The tool then used this information to automatically run the query in Hyperion/Brio, create the required raw data file, run the Access Visual Basic Macros, create the heat map tool file, and format the heat map tool. The layout and features of the heat map tool remained the same as in the previous iteration. The heat map tool file was automatically named by line and date and stored in the same folder as the heat map creation program. When

the program completed creating the tool the program user interface would close, leaving the tool open for use.

Besides the user interface and simplicity, this iteration also improved upon the speed of creation of the heat map tool from six to eight hours down to one or two hours due to programmatic and technical improvements and it reduced the human interaction time to just a few minutes. Technical development will be discussed further in section 5.4 below. Because the program is creating and formatting hundreds of sheets for each run it would be difficult to reduce its processing time any more while still using a Visual Basic Macro. Instead, a status window was added in this iteration to allow the user to know where in the process of creating the tool the program is, adding to the usability. During the creation of each of the sheet the status box reports, "Performing work order analysis: 1 of 100," for example.

5.3.4 The fourth iteration: added features

As the tool was being developed, managers, supervisors, engineers and others were consulted to learn what additional features would be useful in discrepancy analysis. Through these discussions it was determined that two features should be added to the tool. The first feature is a map of the location on the aircraft where discrepancies occurred. This would help users visualize the problems. The second was the ability to see how discrepancy counts changed over time. This was important because if discrepancy counts for a certain type were increasing managers would be able to focus on this issue before it got worse and, on the flip side, would allow them to see if corrective actions they had taken to improve discrepancy counts were effective.

The final iteration of the heat map tool utilized the heat map creation program interface and used the same general layout (Figure 10) as the second and third iterations of the tool including buttons to move between the heat map, location map, and organized data. It also added a button to enable the time scrolling feature. When the enable time scrolling button is pressed a control window (Figure 11) is shown. This part of the tool allows the user to choose to look at the heat map, location map, and

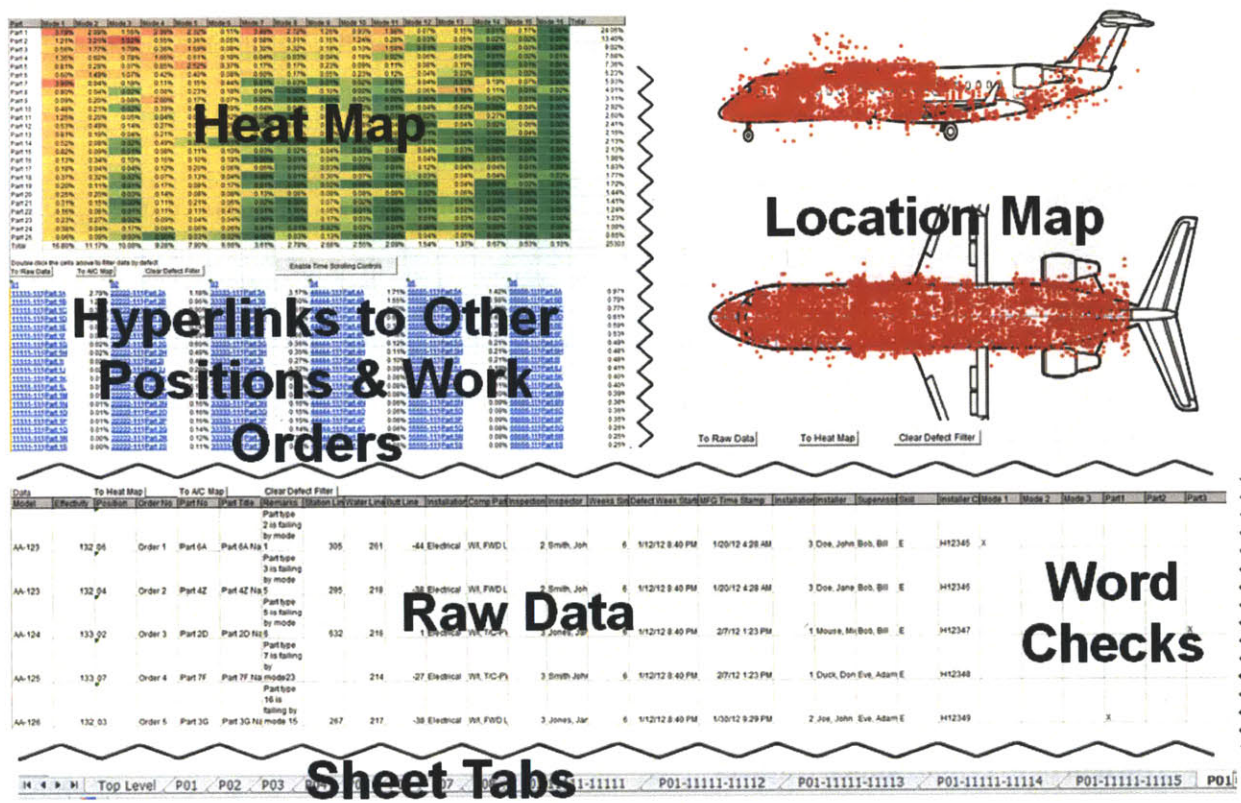


Figure 10: Layout of the heat map tool

organized data for any single effectivity by choosing from a drop down menu. It also allows the user to use back and forward arrow buttons to change the effectivity or jump to the most recent aircraft or oldest

aircraft. Finally, it gives the user the option to have the tool automatically show the heat map effectivity by effectivity in order of oldest to newest and control the speed of this scroll. While the time scrolling is enabled (except during the automatic scroll) the double clicking on a cell will still filter the organized data as well as the defects shown on the location map.

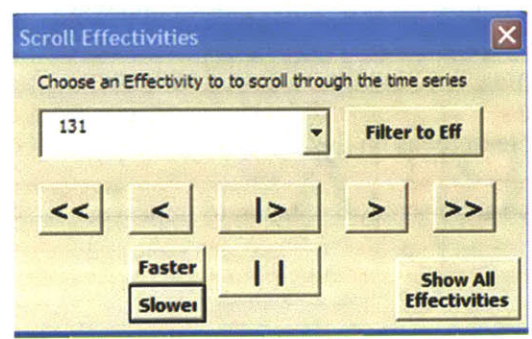


Figure 11: Time scrolling form

5.3.5 Technical Development and Coding of the Tool

The tool was developed using a Visual Basic Macro in Microsoft Excel. The Macro was broken into modules, subroutines, and functions. The key modules were "Create_Tool," "Get_Hyperion_Data," and "Tool_Use." The "Create_Tool" module contains all the subroutines to create the heat map tool

including calling the Access Macro, creating all worksheets, importing data from Access, and formatting the sheets. The “Get_Hyperion_Data” module is used to run the query in Hyperion/Brio and import the tables of raw data into Excel. This was created in a separate module from “Create_Tool” because the “Get_Hyperion_Data” module was a modification of a Macro that had been developed prior to the heat map tool. The “Create_Tool” module did call subroutines from the “Get_Hyperion_Data” module. The “Tool_Use” module was not used in the creation of the tool but contained the subroutines required to enable the functions of the heat map tool such as the sorting with cell double clicks, opening the time scrolling tool, and moving around the sheets with button clicks.

5.3.6 The “Create_Tool” module

First, in the “Create_Tool” module there are some checks to ensure the user has input data correctly before the program calls the sub routines in “Get_Hyperion_Data” which will be discussed further below. Upon beginning the data compilation the program opens a Visual Basic form to display the status of the program which is updated periodically. Once the data has been collected from Hyperion/Brio the program creates the heat map tool file by saving a copy of the heat map creation program. A copy is saved so the coding in the “Tool_Use” module will be in the heat map tool file. The program then sorts work orders by position and by highest to lowest count of discrepancies within each position by using a pivot table. This list of work order numbers is critical because it is the list the program uses to loop through all the positions and work orders to create a sheet for each.

After the list is created, the raw data for all positions and work orders is run through the Access Macro. This Macro which was built prior to the development of the heat map tool program, parses the text in the description of a defect to create counts of common words and organizes these counts and the raw data so that it can be used to create heat maps. The defect descriptions are those written by inspectors and are pulled directly from a database that is part of the computer system that manages the work orders. The organized data table is imported into the heat map tool Excel workbook on a sheet for the top level data. This table contains each defect and whether or not the description of the defect contains certain

common key words describing parts and modes of failure. A simplified example of this table is shown in figure 12. By counting the “X” marks the heat map table is created. The program then performs all the

Work order #	Defect Remarks	Station line	Water line	Butt line	Mode 1	Mode 2	Mode 3	Part 1	Part 2	Part 3
11111	Part 2 failed by Mode 1	90	75	0	X				X	
11112	Other part failed by Mode 2 and Mode 3	60	82	-15		X	X			
11113	Part 3 failed by Mode 3	100	60	12			X			X
11114	Other part failed by other mode	180	90	30						
11115	Part 2 failed by other mode	125	85	-21					X	

Figure 12: Simplified word parse table for creating heat maps

cosmetic formatting on the organized data and heat map, including the sorting and conditional color coding of the heat map. Next the program puts the hyperlinks to each position sheet and work order sheet on the top level sheet. Though the position and work order sheets have not yet been created the hyperlinks and sheet names are both made based on the sorted list referred to above. Since the names are standardized when the sheets are created the links will direct to the proper location.

Once the top level sheet is created, the organized data table for all the other sheets (positions and work orders) is made by filtering the organized data table on the top level sheet. A loop is used to go through each position and create the worksheet for the position. Within the position loop is a loop to create the worksheets for each work order in the position. To create the worksheets the loop uses the sorted position and work order list to add a new named worksheet, filters the organized data table on the top level worksheet to just the position or work order required, copies the filtered data to the newly added position or work order sheet and creates the heat map table for the filtered data. The program then does all the formatting on that sheet.

The last step in the creation of the tool is to create the location maps and buttons for each worksheet. The location maps are created by overlaying a scatter plot of the x-y (station line – butt line) and x-z (station line – water line) location of defects over images of the aircraft. This helps managers easily identify “hot” areas for defects. The tool imports and places the images, creates the scatter plots,

and formats the scatter plots. The plots and buttons are created by looping through each worksheet. They are created at the end of the program rather than within the position and work order loops because the plot feature was added late in the development and it was easier to add the feature by having the program do all the plots at once. Buttons are created with the plots because there are buttons that are located at the plots.

Finally, the program jumps to the top level sheet, saves the heat map tool file, and closes the heat map creation program.

5.3.7 The “Get_Hyperion_Data” module

The “Get_Hyperion_Data” module contains all the subroutines used in opening Hyperion, creating the script that runs the Hyperion query, and importing the Hyperion data to Excel. The query was developed prior to the creation of the tool and was already set up to connect to the proper database and pull the required fields of data. The only part of the query that would change each time the heat map creation program was run were the limits on what data the query would pull. For example, limiting the query to pull only one model line and requested effectivities.

First, the script is written. The program creates a text document to contain the script and writes the required instructions in the text file. The script opens the connection to the database, creates the limits on the model and effectivities, processes the queries, exports the raw data table to Excel, and closes Hyperion. A shell function which is used to run an executable program opens Hyperion with the script to run the query and export the data. Once the query is complete the text file is deleted.

5.3.8 The “Tool_Use” module

The “Tool_Use” module is only used in the heat map tool and is not used to create the tool. Within the tool use module there are subroutines for filtering the organized data as well as event listeners (subroutines that are run when buttons are clicked) for moving around the worksheet and opening the time scrolling form.

There are three filtering subroutines. One subroutine is used when data is to be filtered by a part and a mode, one is used when data is to be filtered by a part or a mode, and one is used to clear the filters and show all data. When a cell is double clicked, depending on the location of the cell (whether it is a part and mode combination, a part or a mode, or a totals cell) one of the three subroutines is invoked. When a part and mode cell is double clicked the cell location is used to look up the name of the part in the row and the name of the mode in the column. This part name and mode name are inputs into the subroutine. The subroutine finds the column of the organized data that contains the part and mode and filters each of those columns to only display rows for which both columns contain Xs. A similar method is used for parts and modes alone. When a totals cell or a “clear filters” button is clicked the subroutine to clear the filters is called.

There are simple subroutines called when the buttons to move around the worksheet are pressed. Each of the subroutines for the three locations (heat map, data, and location map) simply activates the cell that brings that area to the center of the screen. When the “Enable time scrolling” button is pressed the subroutine for that button was the code that shows the form. The time scrolling form will be further discussed in section 5.4.4 below.

5.3.9 Other forms and functions

Additional forms and document modules are used in the heat map tool. To trigger actions when certain workbook events occur the ThisWorkbook module is used. For the heat map tool the ThisWorkbook module is used call events when a cell is double clicked. For each area double clicked in a worksheet there can be a defined action. It is in the ThisWorkbook module that the filtering subroutines are called.

To control the time scrolling a form is used. The time scrolling form not only defines and creates the form but contains code that enables the actual changing of the heat map from effectivity to effectivity. Within the form code there are sub routines to create the pull down menu of effectivities, event listeners

for the different buttons on the form, and a subroutine to clear the effectivity filters if the form is closed. To create the list of effectivities the subroutine finds all unique effectivity values in a set of data (a worksheet), places the effectivities in an array and then sorts them within the array. This array is used to create the drop down menu list of effectivities. This subroutine is run when the form is initialized, when the “Enable time scrolling” button is clicked. The code for each button generally updates the effectivity that is being requested and filters the heat map to that effectivity. For the play and pause buttons the code enters or exits a loop that is repeatedly changing the displayed effectivity. Last, the speed controls update a variable that controls the delay between effectivity changes.

5.3.10 Challenges in development

A number of technical challenges were met with while coding the program. In general the largest challenge to overcome was the length of time it took to run the program to create the tool. In the first semi-automated tool it was taking up to and over eight hours to run a set of data for twenty aircraft. This would be unacceptable as the program would likely have to run overnight. Reduction of the length of the processing time was critical but the amount of data to be processed, hundreds of work orders each with its own sheets, set limits to the processing speed. To try to minimize the length of time to process data first it had to be determined which events were taking the most time.

The first speed driver found was the running of the Access Macro. Initially, the Access Macro was used to create the heat map table because this function had already been developed in Access. Because of this, for each position and work order the Access Macro had to be run on the data for that position or work order. Running the Access program hundreds of times was very slow. Looking at the organized data table showed that the data table contained all the information required to create the heat map table so it could be completed in Excel once the organized data table was imported. Additionally, because the top level contained all the data it could be filtered for each position or work order. With these two factors the Access Macro could be run once to create the organized data table for the top level then

that could be filtered to create organized data tables for each position and work order from which heat map tables could be created. This change cut the processing time in half.

While attempting to speed up code one of the first places to look is at loops. Sometimes a loop can be replaced with a single function by using an array of information rather than stepping through it piece by piece. Originally the program had been written to import the organized data table from Access into Excel cell by cell using a loop. This was incredibly slow. Another method of copying the whole table as a record set was implemented instead which cut the time to copy by about thirty minutes.

Finally, the updating of the screen as each event in the program occurred significantly slows down Excel because the time it takes the program to complete an operation is much less than the time it actually takes to show it on the screen. Similarly, performing calculations for a formula in a cell with each change that occurs takes time at each change. By turning off screen updating and calculating the run time was reduced by about another 25% to bring the time down to about two hours to process the data and create the heat map tool.

5.4 Roll Out of the Tool

After each iteration of the tool was complete it was presented to various managers to get their feedback. This also helped to familiarize people with the tool so that when its development was complete there would be some understanding of the purpose of the tool. In addition, this allowed managers to try making changes based on the heat maps. The manager who initially had asked for the heat map data was the first to utilize the first iteration tool. When he saw the data he noted one cell on the heat map (mode and part) had significantly higher discrepancy counts than any other. Upon seeing this he asked what could be done to help improve that issue. That type of issue had been discovered as a high driver the year prior based on Quality's use of the Access built heat map, and an awareness package to address the issue had already been developed. Within a week of receiving the heap map program data the awareness

package was presented to installers and inspectors from that line. The heat map tool had the impact intended, to encourage managers to take a part in quality improvement.

Once the tool was completed and was determined to be simple enough for everyday use the managers from each line were instructed on the use of the tool. Managers were not only told how to technically use the tool but also how to use the tool to implement change. Very specific step by step instructions both in a written document and presentation format were distributed. Each line was presented to separately, instruction was given, and a demonstration was performed with real data. Seeing the data displayed really shocked some of the managers. It opened their eyes to the severity of some issues and put data to what they may have had a hunch about. After the presentation of the tool managers seemed comfortable with the idea of using the tool in their continuous improvement efforts.

In addition to the managers a few quality engineers were introduced to the tool and taught not only how to use the interface but how the program and code worked. They were also given written directions on how to update parts of the program that were most likely to need to be updated. This was to ensure maintenance of the tool.

5.5 Current Use of the Tool

Though the managers and supervisors initially seemed excited to use the tool, its adoption since then has still been slow. Regardless of the simplicity of the heat map creation program the managers and supervisors have not adopted the use of the interface. Instead, quality engineers have been running the tool and distributing relevant data including the highest driving work orders in each position as well as the defect high drivers. This creates focus on the “vital few” issues using the Pareto principle. The heat maps also are still used to help with visualization of the data.

To address the root cause concerning the difficulty of accessing clear discrepancy data a tool that gathers discrepancy data, runs analysis, and creates a heat map to display data at the push of a button was created. The tool was developed in Microsoft Excel using a Visual Basic Macro and built upon the word parsing, cross Pareto tool developed in 2011. The data display tool was developed in four iterations, each one improving on the ease of use and adding functions based on user needs. The goal of having a simple tool allowing managers and supervisors to easily see discrepancy data is to increase their involvement in the reduction of discrepancies. Due to the data driven culture at X Aerospace employees tend to want to make improvements when they see data indicating areas with opportunities to do so. Prototypes of the tool were used by managers to determine their high driving discrepancies and take corrective actions and line managers and supervisors were instructed on tool use. While they seemed excited about the potential of the tool, currently Quality has been running the tool and passing the data on to the managers. While the initial intent was to have managers run the tool themselves the tool has still made data more easily accessible and is leading to defect reduction efforts.

6 Assessment of Results and Conclusion

This section will discuss the effectiveness of the project. First, the success of the corrective actions taken will be assessed leading to a discussion of the accuracy of the author's initial hypothesis. Next, areas of continuing improvement will be suggested. Finally, the chapter will close with concluding remarks recapping the project and its successes and shortcomings.

6.1 Assessment of Corrective Actions

6.1.1 Standard work for writing inspection instructions

Though the standard work for writing inspection instructions was not adopted as written, its success can be seen by the impact it has made on the way X Aerospace looks at instructions. The development of the standard work inspired further inquiry about the quality of all work instructions and gave X Aerospace the guidelines and tools necessary to develop better instructions. The engineers who develop the work instructions were able to see the potential benefits of some of the features of the improved inspection instructions and have utilized those features such as layout, common syntax, and directive language in improved work instructions.

Despite the standard work not being adopted in its full intended use this corrective action has spurred positive changes in the way X Aerospace writes its inspection and work instructions. These positive changes can be seen as a success in themselves. While corrective actions are developed as methods to mitigate root causes there may be multiple solutions to the problem. In this case, though the large step of implementing the standard work for writing inspection instructions was not taken, smaller steps aiming to correct the poor quality of instructions were.

6.1.2 Data distribution tool

The data distribution tool made immediate impacts. Upon presentation of the first iteration of the heat map tool the manager who had originally requested the heat map data was shocked at how clear it was what the highest driving discrepancies were. Seeing the shocking numbers in such a clearly

displayed format really pushed the manager to want to implement fast corrective actions rather than wait for Quality to lead the corrective action efforts. It achieved getting him invested in the reduction of these discrepancies. He saw one of his highest driving in the heat map and asked Quality if they had any solutions or idea to help reduce this type of discrepancy. There was already an awareness training package in place. The package was presented to one shift of the line's installers and inspectors. The awareness training did reduce the counts of discrepancies on that shift. There was a 19% overall reduction in defects on that shift and a 30% reduction in wire harness security defects. Both reductions were statistically significant ($\alpha=.05$) with t values of .04 and .01 respectively. This demonstrates that using the tool in conjunction with proper corrective actions can create improvements in the numbers of discrepancies.

Knowing that there can be reductions if the tool is used is not enough, however. The managers and supervisors must adopt the tool to make it effective. Initially there was great enthusiasm about the tool because managers and supervisors had never had access to data like this so easily. The difficulty is that managers want the data to be handed to them without having to do any work to get it. Despite the simplicity of the heat map tool creation program managers will not run the program themselves. Instead, quality engineers have been running the tool and handing the data off to managers or using it themselves to suggest corrective actions. Because quality engineers are experienced with data analysis the hand off of information to managers is clearer. While the tool is not being used exactly as intended it has still be successful in getting managers interested in the discrepancy data and encouraged them to get involved in implementing corrective actions on high driving discrepancies.

6.2 Assessment of Hypothesis

Because of the long time scale of building aircraft it is still too early to determine the full accuracy of the author's hypothesis that better data distribution and improved instructions can help reduce the number of discrepancies.

The reduction of discrepancies based on the use of the first iteration of the heat map tool is a good indication that the tool can have a significant impact on defect reduction overall. Additionally, even though the managers and supervisors may not be using the heat map tool creation program themselves they are still requesting the heat map tool (already compiled) which indicates continual utilization of the tool and sustained reduction efforts.

The validity of the assertion that improved instruction will improve defect count is much more difficult to demonstrate. While instructions are currently being improved they have not been fully implemented yet so no data has been collected reporting their effect on defect counts. However, based on the literature reviewed it is clear that such instruction improvements have improved product quality in other companies. If X Aerospace continues working to improve its instructions, even in small steps, they will eventually gain the quality benefits.

6.3 Areas for Continuing Improvement

Though the implementation of improved instructions and the data distribution tool have had some success there is certainly opportunity for continued improvement. Slight changes to installation and inspection instructions and beginning use of the data distribution tool are a good start to make a big impact on discrepancy reduction, but to fully realize these results more positive changes must be made. The opportunities for improvement of instructions still exist in the following:

- Implementation of the standard work for writing inspection instructions for each work order.
- Writing of standard work procedures for writing work instructions.
- Matching of work instructions to inspection instructions in form and content.
- Creation of instructions for performing shake inspections.

If all these points could be achieved it would allow X Aerospace to be more consistent making quality problems more apparent, and allowing for faster corrective actions.

Though the data distribution tool has been used with some success to help reduce high driving discrepancies there is still hesitation from the managers to run the tool creation program themselves. To overcome this some improvements on the tool could be made. The first potential improvement is to have it be fully automated; meaning that it would automatically run every month or other specified time frame for the past 20 aircraft on a line and send a link to the already created heat map tool to managers and supervisors or others who requested the data. This is similar to the method used to deliver other reports to managers such as those on finances and other performance metrics. An additional feature that could be useful for determining whether corrective actions have made a significant difference would be an automated statistical analysis. For example, if a manager performed training on one mode of failure they could put in the date of training and the tool would output a statistical analysis comparing the number of defects before the training and after. Also, the reports of high drivers that quality engineers are creating based on the tool could be automatically created by the heat map creation program. Finally, another area that needs to be further developed is the awareness packages like the one which was used to reduce wire harness securing discrepancies. Currently, these awareness packages are available for some of the high drivers but not all of them.

Additionally, the standard work and data distribution tool were only distributed in one plant and only for the final assembly of the aircraft. Similar instruction improvements and data tools could be implemented in other stages of the aircraft build and at other plants to fully capitalize on improvements.

6.4 Summary and Conclusion

Reworkable discrepancies have been and continue to be a huge opportunity for improvement at X Aerospace. The first step in reduction of these discrepancies is to determine what the discrepancies are and find the root cause of why they are occurring. This project aimed to do that by first understanding each discrepancy that occurred on one aircraft. Based on this analysis root causes were determined and two corrective actions were developed.

The first corrective action was improvement of inspection instructions and the second was a discrepancy data distribution tool. Each of the corrective actions taken had both successes and shortcomings in its implementation. The improved inspection instructions were not implemented, but their creation encouraged the improvement of the current inspection instructions and work instructions. The data distribution tool has been put into use by managers and supervisors to determine where the high driving discrepancies occur and has seen some success with reductions. However, the managers and supervisors still are not comfortable creating the tool directly and go through quality engineers to acquire the tool. Overall, though neither corrective action was perfect in its execution both have paved the way for significant improvements in discrepancy counts with continued development and use.

7 References

- Berk, Joseph. *Cost Reduction and Optimization for Manufacturing and Industrial Companies*. Wiley - Scrivener, 2010.
- Boehm, B., and V. R. Basili. "Defect Reduction Top 10 List." *Computer* (2001): 135-7.
- Burati, J., J. Farrington, and W. Ledbetter. "Causes of Quality Deviations in Design and Construction." *Journal of Construction Engineering and Management* 118.1 (1992): 34-49.
- Dostaler, I. "Avoiding Rework in Product Design: Evidence from the Aerospace Industry." *The International Journal of Quality & Reliability Management* 27.1 (2010): 5-26.
- Gramopadhye, AK, CG Drury, and PV Prabhu. "Training Strategies for Visual Inspection." *Human Factors and Ergonomics in Manufacturing & Service Industries* 7.3 (1997): 171-96.
- Gramopadhye, Anand, et al. "The use of Advanced Technology for Visual Inspection Training." *Applied Ergonomics* 29.5 (1998): 361-75.
- Gryna, F. M. *Quality Planning and Analysis: From Product Development through use*. McGraw-Hill, 2001.
- Harrington, H. J. "Cost of Poor Quality." *International Journal of Strategic Cost Management*. 2.1 (1999):
- Hegazy, Tarek, Mohamed Said, and Moustafa Kassab. "Incorporating Rework into Construction Schedule Analysis." *Automation in Construction* 20.8 (2011): 1051-9.
- Inman, Robert R., et al. "Designing Production Systems for Quality: Research Opportunities from an Automotive Industry Perspective." *International Journal of Production Research* 41.9 (2003): 1953-71.
- Krishnan, Suresh Kumar. "Increasing the visibility of hidden failure costs." *Measuring business excellence* 10.4 (2006): 77-101.
- Lee, Jim, and Srinath Unnikrishnan. "Planning Quality Inspection Operations in Multistage Manufacturing Systems with Inspection Errors." *International Journal of Production Research* 36.1 (1998): 141-56.
- Love, Peter E. D., and Heng Li. "Quantifying the Causes and Costs of Rework in Construction." *Construction Management & Economics* 18.4 (2000): 479.
- Love, Peter E. D. "Auditing the Indirect Consequences of Rework in Construction: A Case Based Approach." *Managerial Auditing Journal* 17.3 (2002): 138-46.
- . "Auditing the Indirect Consequences of Rework in Construction: A Case Based Approach." *Managerial Auditing Journal* 17.3 (2002): 138-46.

- Nickles III, George M., Brian J. Melloy, and Anand K. Gramopadhye. "A Comparison of Three Levels of Training Designed to Promote Systematic Search Behavior in Visual Inspection." *International Journal of Industrial Ergonomics* 32.5 (2003): 331-9.
- Ockerman, Jennifer J., and Amy R. Pritchett. "Preliminary Investigation of Wearable Computers for Task Guidance in Aircraft Inspection". *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE , 1998. 33-40.
- Ockerman, Jennifer J. "Over-Reliance Issues with Task Guidance Systems". *Human Factors and Ergonomics Society Annual Meeting, 43 rd, Houston, TX*. 1999. 1192-1196.
- Ohno, Taiichi. *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, 1988.
- Patel, Swapnesh, Colin G. Drury, and Jay Lofgren. "Design of Workcards for Aircraft Inspection." *Applied Ergonomics* 25.5 (1994): 283-93.
- Scheffler, M., et al. "Modeling and Optimizing the Costs of Electronic Systems." *Design & Test of Computers, IEEE* 15.3 (1998): 20-6. Print.
- Suresh Kumar Krishnan. "Increasing the Visibility of Hidden Failure Costs." *Measuring Business Excellence* 10.4 (2006): 77-101. Print.
- Viswanadham, N., Shashi M. Sharma, and Mukesh Taneja. "Inspection Allocation in Manufacturing Systems using Stochastic Search Techniques." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 26.2 (1996): 222-30. Print.
- Wenner, Caren, Floyd Spencer, and Colin G. Drury. "The Impact of Instructions on Aircraft Visual Inspection Performance: A First Look at the overall Results". *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. SAGE Publications , 2003. 51-55. Print.

8 Appendix

8.1 Defect Analysis Worksheet

Section 1

Model #:		Effectivity:		Positon:	
----------	--	--------------	--	----------	--

Defect #:		WO #:	
Defect Time Written:		Oper #:	

Location:					
Station Ln:		Water Ln:		Butt Ln:	
Type:			Component Part #		
Inspector:					

Remarks:					

MFG Stamp:		MFG UserID:	
------------	--	-------------	--

Section 2

Is this a repeat discrepancy?:		Defect Type:	
How did the discrepancy occur?:			
Who was the installer?:			
What shift?:		What Position?:	
When did the discrepancy get created?:			
What is the responsible unit?:			
What is the responsible function?:			
What is the build criteria?:	(Attach documents if necessary)		
What is the inspection criteria?:	(Attach documents if necessary)		

Section 3

Why is discrepancy occurring?

--

Why?:

--

Why?:

--

Why?:

--

Why?: If more whys are needed continue on back of sheet.

--

Root Cause:

--

How can this be avoided in the future?:

--

8.2 Installer/Inspector Interview Questions

Are you able to look at the blueprint as you are actually doing the installs? Do you?

When you do rework to correct any defect do you find that the descriptions of the defect are detailed enough to allow you to make the correction?

When different Work Orders for similar work have different things included in the instructions does this cause confusion? For example, some Work Orders include the references to blueprint views while others do not.

Have you ever been taught about reuse of hardware or what constitutes damaged hardware? What are the criteria for a “damaged” piece of hardware?

Do you feel that the tools you are given allow you to do your job effectively? If not what improvements could be made to those tools? Do bad tools sometimes cause the need for rework?

When an uncommon problem or install comes up is it difficult to figure out what to do? Do you try to do the install as best you can and wait to see if it is accepted or do you ask for help?

Are instructions on securing wire harnesses clear? Do the defects written for these types of discrepancies seem subjective or are they different for different inspectors?

Is there anything that could make identification decal installs easier? Are the instructions for decal installs clear?

With paint touch ups how do you know when you have touched up well enough? Are the paint touch-up parameters clear?

How are connector strain reliefs installed?

How do you ensure screws are engaged enough? Do you know what the thread engagement requirement is?

Is it difficult to collect serial numbers? Do you use a scanner or do you write/type them by hand?

Do inspectors read shift notes from installers?

8.3 Design of work cards for aircraft inspection guidelines

1 Information readability

- (a) *Typographic layout*
- (1) Use primary typographic spatial cues like vertical spacing, lateral positioning, paragraphing and heading positioning as far as possible.
 - (2) If space usage is premium, then use secondary cueing; e.g. boldfacing, italics, underlining, colour coding and capital cueing in a decreasing order of preference.
 - (3) Use full justification of both margins of the textual material.
 - (4) Use a consistent typographic layout throughout the document.
- (b) *Sentence, word and letter*
- (5) Use of sentence conventions:
 - Boundary conventions
 - initial capitalization
 - final punctuation marks
 - extra space
 - question mark at end of question
 - (6) Use of word conventions:
 - Do not use all-capitals format, use both upper and lower case.
 - Hyphen indicates word division at end of line.
 - Space before and after word.
 - Initial capitalization for proper nouns.
 - (7) Use of letter conventions:
 - Use a typeface like Helvetica that has no redundant features.
 - Avoid using a generic dot-matrix typeface.
- (c) *Printing quality standards*
- (8) Develop and implement standards for changing printer ribbons, toner boxes etc to ensure a consistent print quality at all times.

2 Information content

- (a) *Appropriate content*
- (9) Information provided should be supportive of the inspector's personal goal to 'read quickly and also understand the information', to ensure its usage and eliminate personal biases.
 - (10) It should have certain consistent and common elements to foster generalizations across contexts.
 - (11) It should be accurate.
 - (12) It should be complete, i.e. it should include information regarding what is to be done? where? how? in what order/sequence? which specific terms to pay attention to? references to additional sources of information?
 - (13) It should be up to date with revisions and updates.
 - (14) It should be easy to use and comprehend and hence should be clear and unambiguous.
 - (15) It should be specific and contextual, i.e. pertaining to the particular aircraft being inspected.
 - (16) It should be written in a consistent and standardized syntax.
 - (17) It should be flexible for both expert as well as novice inspectors.
 - (18) Eliminate use of all illogical and self-contradictory statements.
 - (19) Use only certain approved acronyms and proper nouns and provide a glossary if called for.
 - (20) Try to achieve a balance between brevity, elaboration and redundancy of information.
- (b) *Graphic information*
- (21) All spatial information is to be presented in graphical format. Avoid use of textual format for presenting spatial information.
 - (22) The text should assist the graphics and not vice versa.
 - (23) Avoid use of high-fidelity graphics to eliminate clutter. Simple line drawings are superior in most cases.
 - (24) Use a consistent format for figure layout and numbering.
 - (25) Use ordinary numbers, e.g. line 1, 2, 3, etc when referring to figures, and avoid use of complicated reference numbers, e.g. T07-4032-001.
 - (26) Use consistent view-direction information, i.e. use either the UP-AFT icon or the UP-FWD icon, not anything else.
 - (27) The figure views should be as the inspector sees it, from a fixed distance/scale, e.g. 5 ft viewing distance. Avoid use of perspective part drawings as figures.
 - (28) All figures and attachments should have back references to the workcard page/task that originally referred to the figure.
 - (29) Use standard and correct technical drawing terminology, e.g. avoid use of terms 'section' and 'view' interchangeably.
 - (30) Use typographic differentiation between figure titles, part names, crack locations, notes, etc. This differentiation should highlight the importance that one needs to give to each of these, e.g. figure number, crack location, notes, part names, etc in decreasing order of importance calls for boldface cueing for figure numbers.
 - (31) Use standard drawing layout conventions, e.g. location of sectional views with respect to main views.
 - (32) Provide different graphics for spatially mirror-imaged tasks, to reduce the cognitive costs of image inversion, e.g. avoid use of same graphics for both left and right wing inspection tasks.
 - (33) Differentiate close-up views from distant views by giving appropriate scaling information.

3 Information organization

- (a) *Classification of information*
- (34) Distinguish between directive information, reference information, warnings, cautions, notes, procedures and methods.
 - (35) There should be a code for identifying the importance of a particular category of information over others, e.g. warnings, cautions, notes, procedures, methods, directive information, references in decreasing order of importance.
 - (36) Directive information should be broken into the command verb (e.g. check), the objects (e.g. valves, hydraulic lines) and the action qualifiers (e.g. for wear, frays).
 - (37) Each chunk of directive information should not include more than two or three related actions per step to eliminate action slips (e.g. 'remove 10 bolts, remove cover' is acceptable but 'check brake valves, brakes, tires and cables' is not acceptable as one chunk and calls for further breakdown).
 - (38) There should be a clear differentiation between general and specific directive information or tasks, e.g. general tasks usually call for a less detailed inspection over a large but less critical area.
- (b) *Information layering*
- (39) Provide multiple levels of information to cater to the needs of both expert as well as novice inspectors, providing more elaborate information for novices and more concise information for experts performing the same task.
 - (40) Develop a standard framework for distinguishing between and writing multiple layered information. Such a framework should eliminate dependency on fancy phraseology for communication and provide a structure to write into.

(c) *Other organizational issues*

- (41) The task information should be ordered/sequenced in the natural order in which the tasks would be carried out by most inspectors, e.g. according to the spatial location of the tasks as internal tasks and external tasks.
- (42) The page should act as a naturally occurring information module, i.e. it should contain a fixed number of tasks and avoid carryover of tasks across pages. Each task that begins on a page should preferably end on that page too.

4 Physical handling and environmental factors

- (43) The size of the workcard pages should be handy, e.g. avoid using large technical drawings.
 - (44) The entire workcard should not be excessively heavy. It should be such that it can be held continuously for an extended period by any inspector.
 - (45) If the use demands exposure to environmental agents like wind, rain, snow or even harsh and oily floor conditions, adequate precautions should be taken to avoid excessive degradation.
 - (46) It should be compatible with the other tools that an inspector uses in the workplace, e.g. hand tools, boroscope, lighting equipment, etc.
 - (47) Provide a localized reading light in poor lighting conditions.
 - (48) Provide a specialized workcard holder to enable writing in most positions.
 - (49) Provide standard writing tools (pens, pencils, etc) that enable writing in all positions, even against vertical surfaces.
-

From the article "Design of workcards for aircraft inspection" by Swapnesh Patel, Colin G. Drury, and Jay Lofgren published in Applied Ergonomics in 1994

8.4 Inspection Instructions Template/Standard Work

The attached inspection instruction template has been slightly modified to disguise company information but the layout and formatting remain true to that which was presented to X Aerospace.

Contents

Applicability.....	3
Template Use	3
Layout/Contents of the Instructions	3
Information Readability and Organization.....	3
Guide to Use.....	4
Work Order and Aircraft.....	4
Safety	4
Drawing References	5
Spec References	6
Guided Tasks	7
1. Task One: Ex. Harness installs	7
2. Task Two: Ex. Mating connectors	7
3. Task three: Ex. Single part with location	8
4. Continue guided tasks as necessary.....	8
General Inspections.....	9
Grid	9
General Defects	9
1. Previous Order Closed	9
2. FOD (Foreign Object Debris or Damage).....	9
3. General Damage or Hazardous Conditions.....	9
4. Component Part Number and Location with Respect to Blueprint	10
5. ESD (Electrostatic Discharge) Protection.....	10
6. Missing/Incorrect Hardware.....	10
7. Bond Prep and Ground Bonding	10
8. Loose/Poorly installed Hardware.....	11
9. Installation of Connectors/Backshells.....	12
10. Clamps.....	12

11. Sealant/Caulking	13
12. Missing or Damaged Identifications.....	13
13. Wire Terminations	14
14. Paint	15
15. Chafing and Clearance.....	15
16. Grommets.....	16
17. Wire routing	17
18. Dress and Tie	18
Recording Defects	19

Applicability

This template applies to all work orders in final assembly on all lines that do not contain flight safety elements, shake inspections, electrical test inspections, or water test inspections.

Template Use

Parts in **black** can be included as is. Parts in **red** should be replaced with pertinent information. Parts in **blue** are reference notes and should be removed.

Layout/Contents of the Instructions

Guide to Use

Work order and Aircraft

Safety

Drawing References

Spec and Standard Methods References

Guided Tasks

1...

2...

General Inspections

Grid

General Defects

1...

2...

Recording Defects

Information Readability and Organization

Use the font and layout in this template to ensure clear readable text.

Use regular sentence format when possible.

Keep spacing consistent.

If possible keep inspections within a page. Do not split a section between pages.

Use directive language for inspections tasks. For example: →Verify *part* is *mode*.

Try to use levels of information so the top level is least descriptive and under that there is more descriptive information so that novice users will have all the information easily accessible and more experienced users can use the top levels.

Use the symbol → to indicate inspection instructions.

Use a checkbox for guided tasks to be marked as complete.

Put each instruction or reference on a new line.

Use images where clarification is needed.

Be specific but concise.

Keep instructions current and accurate to the work order, drawing, and references.

Follow this template and complete examples for more guidance on how to lay out the inspection instructions.

Each inspection should reference a Spec or Standard Method if at all possible.

Order the guided tasks to match the work order and drawing as much as possible.

Guide to Use

Physical or visual inspection steps are indicated by the symbol →.

The main tasks to be checked off on in the Guided Tasks section below are indicated by a checkbox .

Other information contained in this document is for additional reference and to assist the inspector.

References to specs or other documents are in parenthesis (). These references indicate where the given inspection requirement can be found.

This document is meant to guide inspection but should not be the limit or the act as a specification. Full inspection requirements can be found in the work order, the blueprint, Inspection Instruction Specification document, and the related Specifications and Standard Methods listed in the Spec and Standard Method References section below.

*All Station Line (SL), Water Line (WL), Butt Line (BL) locations are approximate.

Work Order and Aircraft

This Inspection instruction is for Work Order **XXXXXXXX** in Position **XXXX** on the **XXXX** Line.

Safety

Perform this inspection in accordance with Safety Specification X.

Requirements = X		Cutting	Assembly	Sanding	Drilling	Solvents	Inside AC/ Under fixtures/ Top Deck
1	Safety Glasses QEP-865D	X	X	X	X	X	X
2	Safety Shoes ANSI Z41	X	X	X	X	X	X
3	Ear Plugs QNP-400	X	X	X	X	X	X
4	Gloves: Cut Resistant GLV-9700	X					
5	Gloves: Leather Palm – Knit Wrist GLV-50-10		X	X			
6	Gloves: Latex GLV-60					X or	
7	Gloves: Nitrile Disposable GLV-300					X	
8	Dust Mask: QRP-8577			X			
9	Face Shield with Clear Visor				X or		
10	Goggles Vented QEP-502				X		
11	Bump Cap						X

Drawing References

XXXXXX-XXXXX Dash Number -XXX						
View	Sheet	A/C Location*			Drawing Location	References
		SL	WL	BL		
Notes (If not on the plan view sheet)					Grid location (ex. A12-G15)	
Plan View					Grid Location	Section and Detail views found in this view
Section/Detail Views						

Continue for all views referenced in the Work Order or containing parts installed or worked in the Work Order.

All drawings and graphics referenced in the Work Order should have a chart similar to the above.

Views should be in Alphabetical order by hierarchy. For example, if the plan view contains A-A, B-B, and C-C and B-B contains D-D then the views would be listed in the following order.

View	References
Plan View	A-A, B-B, C-C
A-A	
B-B	D-D
D-D	
C-C	

Spec References

A list of all Specifications, Standard Methods, Inspection Instruction Specification documents and other reference documents required for inspection of the Work Order. This includes all documents (other than drawings) listed in the Work Order and all references required for inspection of installs performed in the Work Order. Only include pertinent references.

The documents should be listed with number and title in order of number grouped by Specs first, Standard Methods second, Inspection Instruction Specification documents third and all other documents after Inspection Instruction Specification documents..

Below is a list of documents commonly used for inspections. Only include those that pertain to the work order.

- Spec 1111– Grommets
- Spec 1112 – Electrical Grounding
- Spec 1113 – Wire Terminations
- Spec 1117– Identification of Electrical Components
- Spec 1118– Waterproofing Electrical Components
- Spec 1119– Lacing and Tape Tying
- Spec 1121– Securing Wire Harnesses
- Spec 1122– Connector Strain Relief
- Spec 1125– Mounting of Electrical Connectors
- Spec 1130– Electrical and Electronic System Detail Requirements
- Spec 1131– The Control of Electrostatic Discharge
- Spec 1137– Wire Harness Clamping
- Spec 1143– Wire Chafing
- Spec 1146– Sealing and Insulation of Mating Surfaces
- Spec 1152– Adhesive Bonding
- Spec 1159– Torque, Wrench, and Tightening
- Standard Method 11-12 – Adhesive Bonding per Spec 1152
- Standard Method 11-26 – Wire Harness Assembly
- Standard Method 11-37– Electrical Grounding
- Standard Method 11-43 – Installation of Identification
- Inspection Instruction Specification (IIS) 987 – Inspection Guidelines and Criteria for Electrical/Electronic Installations
- Inspection Instruction Specification (IIS) 102 – Inspection Guidelines and Criteria for Mechanical Installations

Guided Tasks

Guided tasks should include tasks specifically called out in the Work Order with specific part installs. If a Work Order calls out the install of all harnesses between SL 200 and 260 and BL 0-L30 or in specific views you must look at the drawing to determine the harnesses and their installs that will be listed on the guided tasks. If a Work Order says mate connector JXXX, JYYY, JZZZ that would be a guided task. If a Work Order says ensure all grounds are painted but does not call out any specific locations, drawing views or grounds that would not be in the guided tasks.

A good rule of thumb to use is if the Work Order uses the word “ensure” and/or gives a view, location, or part number that task should have a guided task.

You must also include any of the flagged notes on the drawing in the guided tasks. For example, if a Work Order says to install harness A and there is a flag on harness A that says it should be spiral wrapped that should also be included in the guided tasks. As another example if a Work Order says install decals and the drawing gives a list of decals in the notes or flags locations for decal installs those should be included in the guided tasks. Use the sample instructions for more examples and guidance.

1. Task One: Ex. Harness installs

→ Verify that the following harnesses are routed and clamped

Harness #	Harness Name	Applicable Views
XXXXX-XXXXX	Some Wires HA	XXXXX-XXXXX(drawing number){View, View...} (if multiple drawings referenced in Work Order)
XXXXX-XXXXX	Some Wires HA	View, View...(if one drawing referenced in Work Order)

Task Two: Ex. Mating connectors

→ Verify that the following connectors are mated tightly.

Connector	Aircraft Location*			View
	SL*	WL*	BL*	
P/JXXX	XX	XX	L/RXX	View, View

→ Verify connector keyways are at 12 o'clock when the mounting flange is vertical and face the forward of the aircraft when the mounting flange is horizontal unless otherwise specified by the blueprint. (Spec 1125)

→ Verify flange mounting screws or nuts do not interfere with the mating connector plug. (Spec 1125)

→ Verify the flange is mounted with the screw head on the mating side of the connector. (Spec 1125)

3. Task three: Ex. Single part with location

→Verify the installation of part.

<input type="checkbox"/>	Aircraft Location*			View
	SL*	WL*	BL*	
	XX	XX	L/RX X	View, View...

4. Continue guided tasks as necessary...

→Verify

General Inspections

Grid

The grid can be broken down however you see fit. If for example, if most of the install is concentrated in bands from front to back you could use shorter station lengths and wider water and butt line areas. 10"-20" X 10"-20" X 10"-20" is recommended.

The inspection area will be broken up into approximately XX" of station line by XX" of water line by XX" of butt line areas for inspection. Inspect each area for the defects listed in the *General Defects* section below.

Section	Station Line	Water Line	Butt Line
1	XXX-XXX	XXX-XXX	XXX-XXX
2	XXX-XXX	XXX-XXX	XXX-XXX
3	XXX-XXX	XXX-XXX	XXX-XXX
4...	XXX-XXX	XXX-XXX	XXX-XXX

General Defects

The general defects section should pertain to all types of installs performed in the Work Order. Included here are many popular types of defects. Sections 1-7 should be included in all inspections. The sections below are only examples and should not limit inspections to be included. If other general tasks are required add them. Do not include sections that do not pertain to the install. The general tasks can be built from the IIS's but each inspection should reference a Spec or Standard Methods if at all possible.

1. Previous Order Closed

- Check previous inspection operations for completion, for example, if an "OK to Install" is required verify that is complete. (Work Order)
- Verify the Work Order is for the proper aircraft number. (IIS 987, IIS 102)
- Check that all manufacturing operations have been signed off. (IIS 987, IIS 102)
- Check that the component part list is accurate. (IIS 987, IIS 102)

2. FOD (Foreign Object Debris or Damage)

- Verify that the area is free of dirt, particles, tools, uninstalled or loose parts, etc. (IIS 987, IIS 102)

3. General Damage or Hazardous Conditions

- Check the area for damage including but not limited to cut or damaged wires, scratches, burrs, tears, broken parts, sharp edges, dents, etc. (IIS 987, IIS 102)

4. Component Part Number and Location with Respect to Blueprint

→Verify that part numbers match those given on the blueprint and Work Order. (IIS 987, IIS 102)

→Verify that parts are installed in the proper locations per the blueprint. (IIS 987, IIS 102)

Some parts such as clamps may be allowed to be installed in a slightly different location than indicated as long as there are no interference or chafe conditions.

5. ESD (Electrostatic Discharge) Protection

All unmated connectors require ESD protection.

→Verify unmated and unmounted connectors are protected at least by a plastic ESD bag. (Spec 1131)

→Verify that mounted unmated connectors have plastic dust caps installed. (Spec 1131)

Unmated unmounted connectors may have both plastic dust caps and ESD bags as protection, but must have at least an ESD bag.

→Verify that plastic dust caps and/or ESD bags are well secured. (Spec 1131)
ESD bags must be secured with string ties or twist ties.

6. Missing/Incorrect Hardware

→Verify installation of all hardware required by the blueprint. (IIS 987, IIS 102)

→Verify all hardware is of the proper material specified by the blueprint. (IIS 987, IIS 102)

The blueprint indicates all hardware material unless otherwise specified.

7. Bond Prep and Ground Bonding

→Verify all ground bonded areas are repainted in an area around the ground bond and slightly larger than the prepared area. (Spec 1112, SM11-37)

It is allowable to paint the attaching hardware but is not required.

→Verify paint has not wicked under the ground lug. (Spec 1112)

→Verify ground lugs external to the aircraft are sealed. (Spec 1112)

→Verify that each ground bond has a lock washer installed. (Spec 1112)

→Verify ground bond lugs are only bent in the case of space limitations and for ease of paint and sealant application. (Spec 1112)

→Verify ground lugs are bent to a 45 degree angle maximum. (Spec 1112)

→Verify washers are used to separate dissimilar metals surfaces. (Spec 1112)

When ground bonding to steel the intermediary washer is not required.

→Verify ground bonds to steel surfaces are not sealed unless indicated by the blueprint. (Spec 1112)

→Verify connectors have been paint touched up. (Spec 1112)

8. Loose/Poorly installed Hardware

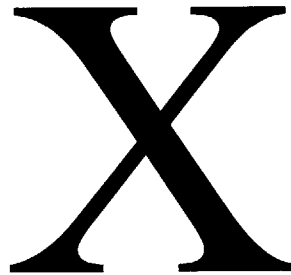
→Verify that hardware, such as nuts, bolts, washers, screws, rivets, etc., are tight. (Spec 1159)

If the assembly torque values are unspecified and the size of the fastener is less than #10 or the tensile strength of the fastener is less than 125,000 psi, then the threaded parts shall be tight. Tight means that the hardware is “firmly secured and that there shall be no relative movement between the attached parts”

→Verify there is no damage to the hardware, including but not limited to stripped screws and burrs on the fastener threads or head. (Spec 1159)

→Verify full engagement (also known as breaking safety).

Full engagement is defined as two threaded pitches, including the chamfer, extending beyond the locking device of the nut. (Spec 1159)



Images showing methods of verifying full
engagement

9. Installation of Connectors/Backshells

- Verify connectors are mated as called out in the blueprint. (IIS 987)
- Verify connector keyways are at 12 o'clock when the mounting flange is vertical and face the forward of the aircraft when the mounting flange is horizontal unless otherwise specified by the blueprint. (Spec 1125)
- Verify flange mounting screws or nuts do not interfere with the mating connector plug. (Spec 1125)
- Verify the flange is mounted with the screw head on the mating side of the connector. (Spec 1125)
- Verify all mated connectors are tight as defined above. (IIS 987, IIS 102)
- Verify all backshells are tight as defined above by applying hand torque in the tightening direction. (Spec 1122)
- Verify Jam Nut connectors are mounted using the "O" ring supplied, torques and safety wired. (Spec 1125)
- Verify the flange of the Jam Nut connector is flush with the structure and the "O" ring is seated in the groove. (Spec 1125)

10. Clamps

- Verify clamps are installed within +/- .5" as specified in the blueprint. (Spec 1137)
- Verify clamp is within +/- 2 sizes and of the same material of that indicated by the Work Order and/or blueprint. (Spec 1137)
- Verify that clamp holds wires securely without crushing them. (Spec 1137)
Wires should be able to slide through the clamp with a pulling force of 2-4 lbs. but should not slide freely through the clamp.
Coax cables should be able to slide through the clamp with a pulling force of 1-2 lbs. but should not slide freely through the clamp.
- Verify clamp wedge touches (preferred overlaps) the opposite end of the clamp. (Spec 1137)
- Verify Mounting holes of the clamp are aligned and the clamp is completely closed. (I1919)
- Verify no wiring is touching an uncushioned part of the clamp band. (Spec 1137)
- Verify no wiring is pinched in the clamp. (Spec 1137)
- Verify proper hardware is used. (Spec 1130)
Steel hardware should be used above flight control systems with moving components unless the Work Order or blueprint indicates otherwise.
A sacrificial washer should be used when no nut plate is installed.
Screw and nut or nut plate should have the same material.
- Verify clamps or primary supports are no more than 24" apart. (Spec 1130)
- Verify spacers are used as required by the blueprint. (Spec 1130)
Spacers may vary $\pm .25$ " from the blueprint requirement to meet anti-chafe conditions.
- Verify spacers are one part (not stacked to achieve desired length). (Spec 1130)

11. Sealant/Caulking

→Verify there is squeeze out around the whole fastener for wet installed fasteners. (Spec 1146)

→Verify that all voids channels and gaps are filled with sealant. (Spec 1146)

→Verify there are no voids or air bubbles on the edges of the sealant. (Spec 1146)

Surface imperfections appearing as pits, dimples or air bubbles shall not be cause for rejection.

12. Missing or Damaged Identifications

→Verify the location of the decal is on the plug (P)/receptacle (R) face side (Spec 1117, SM 11-43)

If both sides of the mounting structure cannot be seen the decal must be mounted on both sides.

If the receptacle (R) is not attached to a mating plug (P) the receptacle decal must be placed on the plug side of the structure.

→Verify the decal is not obscured by the mounting hardware.

→Verify that the decal has no edges or corners lifting from the bonding surface. (Spec 1152)

→Verify that wire harnesses and coax cables have visible identification. (Spec 1117, SM 11-43)

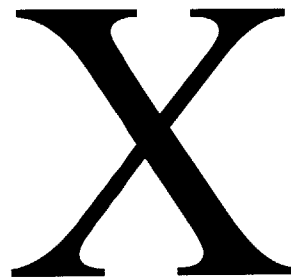
→Verify that there are non-metallic identification bands with the connector "P" or "J" number around the wire group, cable, or harness 6 inches or less from the connector. (Spec 1130)

→Check that connectors are all identified. (Spec 1130)

If the decal is in an area where it could be exposed to aircraft or environmental fluids:

→ Verify that the decal has the proper sealant applied. (Spec 1146)

→Verify that sealant is 1/8"-3/8" wide and continuous around all edges of the decal (Spec 1146)



Images showing proper identification
of wire harness straps

13. Wire Terminations

- Verify terminal board and block screws are breaking safety. (Spec 1113)
- Verify that there is no protrusion of mounting hardware above the terminal board plastic insulation. (Spec 1113)
- Verify screws are mounted from the top of terminal blocks if possible. (Spec 1113)
- Verify there is clearance between terminal board and rivet patterns, etc. Spacers may be used under mounting points to provide this clearance (Spec 1113)
- Verify terminal boards with 12 or less terminals are attached to structure in a minimum of two places, at or near each end. (Spec 1113)
- Verify terminal boards with more than 12 terminals are attached to structure in a minimum of 3 places, one screw at or near each end, and one screw at or near the middle of the terminal board. (Spec 1113)
- Verify Terminal Board insulation strips sit flat on their mounting surface and do not contact the terminal board hardware. (Spec 1113)
- Verify the proper number of holes in the insulation strip.
A "one" hole insulation strip shall be used when the mounting hardware is at the end of the terminal board outboard of the first terminal. A "two" hole insulation strip shall be used when the mounting hardware is between terminals of the terminal board. (Spec 1113)
- If required by the blueprint verify terminal boards are protected by a cover. (Spec 1113)
- Verify no more than four (4) terminals, or three (3) terminals and a bus bar, shall be attached to any single stud except when specifically called for by Work Order and B/P. (Spec 1113)
- Verify that terminals are stacked with the greatest insulation sleeve diameter on the bottom and the smallest on the top. (Spec 1113)
- Verify the top terminal lug locks the terminal lug stack against movement in the direction of loosening. (Spec 1113)
With fingertip, push top lug away from post with 2-3 pounds force to verify stack locking
- Verify there is sufficient slack for two lug reterminations. (Spec 1130)
- Verify terminal lugs are stacked on the terminal as follows:
 - 1 lug per terminal, barrel up
 - 2 lugs per terminal both lug barrels up and fanned
 - 3 lugs per terminal 1st and 2nd lug barrel down, 3rd barrel up, lugs fanned,
 - 4 lugs per terminal 1st and 2nd lug barrel down, 3rd and 4th lug barrel up, lugs fanned (Spec 1113)

X

Images showing proper and improper terminal lug stacking

14. Paint

- Verify that wherever paint has been removed the area is touched up to match the original finish. (Spec 1112)
- Verify no bare metal is showing. (Spec 1112)
- Verify paint has not filled the heads of screws. (Spec 1112)

15. Chafing and Clearance

- Verify all wires are clamped so that at least .5" of clearance are maintained when the wires are pushed toward moving system components or control cables. (Spec 1143)
- Verify wiring is protected from abrasion, excess bending or twisting due to relative motion of attached parts or assemblies. (Spec 1143)
- Verify that the separation between wiring and edges of equipment or structure is at minimum 3/8". (Spec 1143)
If separation of 3/8" cannot be maintained spiral wrap (if acceptable) or grommets must be used.
Wires that can contact a flat surface do not constitute a chafe condition as long as they are not preloaded into the surface.
- Verify parallel wiring is no more than 10 degrees crossed. (Spec 1143)
Greater than 10 degree crossed wire that is preloaded (not occasionally contacting due to slack) constitutes a chafe condition.
- Verify wiring routed parallel to plumbing lines is level with or above the plumbing. (Spec 1130)
- Verify there is at least .5" separation between wires and plumbing lines. (Spec 1130)
- Verify wiring is not attached to plumbing unless the separation is less than 2". (Spec 1130)
- Verify any wiring passing below plumbing line passes at an angle. (Spec 1130)
Presstite may be used to determine preloading.

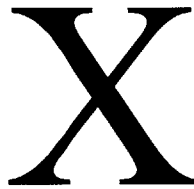
16. Grommets

- Verify split rubber grommets are cut diagonally at a 45 degree angle. (Spec 1111)
- Verify the gap between ends of split rubber grommets and nylon caterpillar grommets is no more than 1/16" (Spec 1130)
- Verify the gap in split rubber and nylon caterpillar grommets is oriented at the top of the hole or in such a way that any wire pressure would not be on the split. (Spec 1130)
- Verify nylon caterpillar grommets are cut only between the teeth. (Spec 1111)
- Verify teeth on nylon caterpillar grommets are removed to clear obstructions such a as clips. (Spec 1130)
- Verify grommets are not lifting or disbonding. (Spec 1152, SM 11-12)
Apply approximately 2 lbs. of pulling force to the ends of the bonded grommet at the gap to ensure the grommet does not lift from the structure.
- Verify nylon caterpillar grommets are not painted over with bonding agent. (Spec 1152, SM 11-12)
- Verify that protective nylon caterpillar grommet strips are installed where 3/8" clearance cannot be maintained between wires and an edge. (Spec 1143)
- Verify strip does not extend more than 1.5" beyond the possible chafe zone unless otherwise indicated by the blueprint. (Spec 1143)

17. Wire routing

→Verify wires are routed parallel or perpendicular to stringers and ribs of the aircraft unless otherwise indicated by the BP. (Spec 1143)

→Verify wires twist rather than bend across hinges. (Spec 1143)



Images showing proper and improper of wire harness routing across hinges

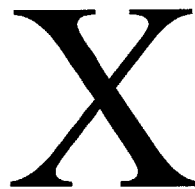
→Verify bend radii of wires and harnesses are not too tight. (Spec 1143)

10x dia. of individual wires, 3x dia. of individual wires at supported areas

6x diameter of coax cables

6x the cross-sectional dimension of the harness in the plane of the bend and at minimum 10x the diameter of the largest single wire in the bundle.

→Verify that wires dressed downward to a termination have a drip loop to prevent fluid from running into the termination. (Spec 1130)



Images showing proper and improper of drip loops

→Verify that with moderate hand pressure a slack of .5" exists between support points. (Spec 1143)



Image showing proper slack measurement

18. Dress and Tie

→Verify ties and tie wraps are at maximum 12" apart and are evenly spaced. (Spec 1130)

→Verify that ties are of approved tape, wrap, and knots. (Spec 1121)

For flat tie tape the Technicians knot or the X Aerospace knot should be used.

→Verify the ends of ties are trimmed to a minimum of 3/8". (Spec 1121)

→Verify plastic tywraps are trimmed with a flat edge and not protrusion from the boss. (Spec 1121)

→Verify that tywrap holds wires securely without crushing them. (Spec 1121)

Wires should be able to slide through the tywrap with a pulling force of 2-4 lbs. but should not slide freely through the tywrap.

Coax cables should be able to slide through the tywrap with a pulling force of 1-2 lbs. but should not slide freely through the tywrap.

→Verify there are no ties or tywraps on the portion of wires contained in conduit. (Spec 1121)

→Verify tywraps are not used in restricted areas. (Spec 1121)

Restricted areas include:

- Locations over flight control moving components such as pivots, pulleys, etc.
- Locations where temperature exceeds 85°C (185°F)
- Locations where strap failure would permit the wiring to move against parts or foul mechanical linkages or permit the strap to fall into moving parts
- Locations where exposure to ultra-violet light may exist (unless straps are ultra-violet resistant)
- Locations where high vibration levels exist such as transmissions or engine compartments
- Locations where the strap is exposed to external environmental conditions, i.e., outside of the fuselage including top deck, pylon, external racks, pylons and wheel wells.

→Verify that spiral wrap is only used where it is authorized by the blueprint. (Spec 1143)

→Verify spiral wrap is not used where wires cross sharp edges at an angle or wires are above control cables. (Spec 1143)

→Verify ends of spiral wrap are secured with ties or tywraps. (Spec 1130)

Recording Defects

When recording defects if possible specify:

- station line
- water line
- butt line
- drawing view
- harness numbers or connector numbers
- component part numbers
- defect mode (ex: loose)
- defect part (ex: clamp)
- if it is a reopened discrepancy include the word “still”
- indication if the defect seems to be from a prior install

Example:

Screw on clamp setup (part) around harnesses XXXXX-XXXXX (part number) is not breaking safety (defect mode).

Drawing View(s): XXXXX-XXXXX, Sheet X, View XX, Grid XXX

SL: XX WL: XX BL: XX

8.5 Heat Map Creation Program and Heat Map Tool Code

8.5.1 ThisWorkbook Module

```
1 Option Explicit
2 Private Sub Workbook_Open()
3     'Resets the program or tool on opening
4     Application.DisplayAlerts = True
5     If ActiveWorkbook.Name Like "*Defect HeatMapProgram*" Then
6         ActiveWorkbook.Sheets("Run Program").Unprotect Password:="heatmapprogram"
7         ActiveWorkbook.Sheets("Run Program").Range("B11:B30").Locked = False
8         ActiveWorkbook.Sheets("Run Program").Range("B11:B30").ClearContents
9         ActiveWorkbook.Sheets("Run Program").Protect Password:="heatmapprogram", UserInterfaceOnly:=True, _
DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowSorting:=True, AllowFiltering:=True
10        ActiveWorkbook.Sheets("Run Program").EnableSelection = xlUnlockedCells
11        ActiveWorkbook.Sheets("Control Sheet").Protect Password:="heatmapprogram", UserInterfaceOnly:=True,
DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowSorting:=True, AllowFiltering:=True
12    End If
13    'Checks that the tool is saved on the hard drive, cannot be run properly from the temp folder
14    If ActiveWorkbook.Path Like "**Temp*" Or ActiveWorkbook.Path Like "**temp*" Or ActiveWorkbook.Path Like
**TEMP*" Then
15        If ActiveWorkbook.Name Like "*_*" Then
16            MsgBox "This file will not work from a temporary folder." & vbCrLf & _
"Download the file to your hard drive then open." & vbCrLf & _
"Work box will close when this box is closed.", _
vbOKOnly, "Save File"
17            Dim wBook As Workbook
18            Dim i As Integer
19            i = 0
20            For Each wBook In Workbooks
21                i = i + 1
22            Next wBook
23            If i = 1 Then
24                Application.DisplayAlerts = False
25                Application.Quit
26            Else
27                Application.DisplayAlerts = False
28                ActiveWorkbook.Close
29            End If
30        End If
31    End If
32    Application.Calculation = xlCalculationAutomatic
33 End Sub
34 Sub Workbook_SheetBeforeDoubleClick(ByVal Sh As Object, ByVal Target As Range, Cancel As Boolean)
35     Dim FilterString1 As String
36     Dim FilterString2 As String
37     Dim FilterString3 As String
38     Dim MainRange As String
39     Dim LastRowRange As String
40     Dim LastColRange As String
41     Dim FirstColRange As String
42     Dim TotalPartRange As String
43     Dim TotalRange As String
44     'Since this code applies to the program and tool the double clicking ranges will only apply to the
tool
45     If Sh.Name <> "Run Program" Then
46         MainRange = "$B$2:$Q$" & Range("A1:A40").Find("Total").Row - 1
47         LastRowRange = "$B$" & Range("A1:A40").Find("Total").Row & ":$Q$" &
Range("A1:A40").Find("Total").Row
48         LastColRange = "$R$2:$R$" & Range("A1:A40").Find("Total").Row - 1
49         FirstColRange = "$A$2:$A$" & Range("A1:A40").Find("Total").Row - 1
50         TotalPartRange = "$A$" & Range("A1:A40").Find("Total").Row
51         TotalRange = "$R$" & Range("A1:A40").Find("Total").Row
52     'Double click the part and mode range
53     If Not Intersect(Target, ActiveSheet.Range(MainRange)) Is Nothing Then
54         With Target
55             'Unprotects to allow for double click
56             ActiveSheet.Unprotect
57             ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Bold = False
```

```

58     ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Italic = False
59     FilterString1 = ActiveSheet.Cells(1, ActiveCell.Column)
60     FilterString2 = ActiveSheet.Cells(ActiveCell.Row, "A")
61     'Bolds and italicizes filtered cell
62     ActiveCell.Font.Bold = True
63     Cells(1, ActiveCell.Column).Font.Bold = True
64     Cells(ActiveCell.Row, 1).Font.Bold = True
65     ActiveCell.Font.Italic = True
66     Cells(1, ActiveCell.Column).Font.Italic = True
67     Cells(ActiveCell.Row, 1).Font.Italic = True
68     'Filters the data
69     Call FilterData(FilterString1, FilterString2)
70 End With
71 End If

72 ''Double Click the Mode totals
73 If Not Intersect(Target, ActiveSheet.Range(LastRowRange)) Is Nothing Then
74     With Target
75         'Unprotects to allow for double click
76         ActiveSheet.Unprotect
77         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Bold = False
78         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Italic = False
79         FilterString1 = ActiveSheet.Cells(1, ActiveCell.Column)
80         'Bolds and italicizes filtered cell
81         ActiveCell.Font.Bold = True
82         Cells(1, ActiveCell.Column).Font.Bold = True
83         ActiveCell.Font.Italic = True
84         Cells(1, ActiveCell.Column).Font.Italic = True
85         'Filters the data
86         Call FilterData2(FilterString1)
87     End With
88 End If

89 ''Double click the part totals
90 If Not Intersect(Target, ActiveSheet.Range(LastColRange)) Is Nothing Then

91     With Target
92         'Unprotects to allow for double click
93         ActiveSheet.Unprotect
94         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Bold = False
95         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Italic = False
96         FilterString3 = ActiveSheet.Cells(ActiveCell.Row, "A")
97         'Bolds and italicizes filtered cell
98         ActiveCell.Font.Bold = True
99         Cells(ActiveCell.Row, 1).Font.Bold = True
100        ActiveCell.Font.Italic = True
101        Cells(ActiveCell.Row, 1).Font.Italic = True
102        'Filters the data
103        Call FilterData2(FilterString3)
104    End With
105 End If

106 ''Double click the mode names
107 If Not Intersect(Target, ActiveSheet.Range(FirstColRange)) Is Nothing Then
108     With Target
109         'Unprotects to allow for double click
110         ActiveSheet.Unprotect
111         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Bold = False
112         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Italic = False
113         FilterString3 = ActiveCell
114         'Bolds and italicizes filtered cell
115         ActiveCell.Font.Bold = True
116         Cells(ActiveCell.Row, 18).Font.Bold = True
117         ActiveCell.Font.Italic = True
118         Cells(ActiveCell.Row, 18).Font.Italic = True
119         'Filters the data
120         Call FilterData2(FilterString3)
121     End With
122 End If

123 ''Double click the part names
124 If Not Intersect(Target, ActiveSheet.Range("$B$1:$Q$1")) Is Nothing Then
125     With Target
126         'Unprotects to allow for double click
127         ActiveSheet.Unprotect
128         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Bold = False
129         ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Italic = False
130         FilterString3 = ActiveCell
131         'Bolds and italicizes filtered cell
132         ActiveCell.Font.Bold = True
133         ActiveCell.Font.Italic = True

```

```

134     ActiveCell.Font.Bold = True
135     Cells(Range("A1:A40").Find("Total").Row, ActiveCell.Column).Font.Bold = True
136     ActiveCell.Font.Italic = True
137     Cells(Range("A1:A40").Find("Total").Row, ActiveCell.Column).Font.Italic = True
138     'Filters the data
139     Call FilterData2(FilterString3)
140     End With
141 End If

142     ''Double click totals areas
143     If Target.Address = "$A$1" Then
144         'Clears filter
145         Call ClearFilterData
146     End If

147     If Target.Address = "$R$1" Then
148         'Clears filter
149         Call ClearFilterData
150     End If

151     If Target.Address = TotalPartRange Then
152         'Clears filter
153         Call ClearFilterData
154     End If

155     If Target.Address = TotalRange Then
156         'Clears filter
157         Call ClearFilterData
158     End If

159     Application.EnableEvents = True

160     If TimeSeries.Visible Then
161         ActiveSheet.Range("S1").Value = TimeSeries.Top
162         ActiveSheet.Range("S1").Value = TimeSeries.Left
163         Application.OnTime Now + (TimeValue("00:00:01")) / 5, "TimeScrolling"
164     End If

165 End If

166 Cancel = True
167 ActiveSheet.Protect UserInterfaceOnly:=True, DrawingObjects:=True, Contents:=True, Scenarios:=True,
AllowSorting:=True, AllowFiltering:=True
168 End Sub

```

8.5.2 Create_Tool Module

```

1 Option Explicit
2 Dim PrgWkBk As Workbook
3 Dim RDWkBk As Workbook
4 Dim DataWkBk As Workbook
5 Dim FinalWkBk As Workbook
6 Dim CurrentPos As String
7 Dim CurrentPosWO As String
8 Dim WrkingSht As String
9 Dim HeatMapSheet As String
10 Dim UpdatedDataSheet As String
11 Dim FolderPath As String
12 Dim LineLetter As String
13 Dim PosLetter As String
14 Dim NumPositions As Integer
15 Dim CurrentPosNumber As Integer
16 Dim Line As String
17 Dim TemplateSheet As String
18 Dim LastSheet As String
19 Dim startTime As Double
20 Dim endTime As Double
21 Global NumWOs As Integer
22 Global CompleteWOs As Integer
23 Global oApp As Object
24 Dim FirstPos As String

25 Sub ConditionalColors()
26     Dim CondColorRange As Range

```



```

27     'Set the range for the heat map
28     Set CondColorRange = FinalWkBk.Sheets(TemplateSheet).Range("$B$2:$Q$" &
FinalWkBk.Sheets(TemplateSheet).Range("A2"). _
    End(xlDown).Row - 1)

29     'Color for heat map
30     CondColorRange.FormatConditions.AddColorScale ColorScaleType:=3
31     CondColorRange.FormatConditions(CondColorRange.FormatConditions.Count).SetFirstPriority
32     CondColorRange.FormatConditions(1).ColorScaleCriteria(1).Type = _
xlConditionValueLowestValue
33     With CondColorRange.FormatConditions(1).ColorScaleCriteria(1).FormatColor
34         .Color = 8109667
35         .TintAndShade = 0
36     End With
37     CondColorRange.FormatConditions(1).ColorScaleCriteria(2).Type = _
xlConditionValuePercentile
38     CondColorRange.FormatConditions(1).ColorScaleCriteria(2).Value = 50
39     With CondColorRange.FormatConditions(1).ColorScaleCriteria(2).FormatColor
40         .Color = 8711167
41         .TintAndShade = 0
42     End With
43     CondColorRange.FormatConditions(1).ColorScaleCriteria(3).Type = _
xlConditionValueHighestValue
44     With CondColorRange.FormatConditions(1).ColorScaleCriteria(3).FormatColor
45         .Color = 7039480
46         .TintAndShade = 0
47     End With

48     'Sort largest to smallest
49     Call ParetoSort

50 End Sub

51 Sub ContinuationAfterHyperion()

52     Dim ws As Worksheet
53     Dim j As Integer
54     Dim LastPos As String
55     Dim PivotRange As Range

56     'Update Status Box
57     Status.WOComp.Value = "Running: Preparing Workbooks"
58     Status.Repaint

59     'Open Hyperion Output Workbook
60     Set DataWkBk = Workbooks.Open(Filename:=FolderPath & Line & ".xls")

61     'Check if there is data
62     If Application.CountA(DataWkBk.Sheets(Line).Cells(2, 1).EntireRow) = 0 Then
63         MsgBox "Hyperion collected no data. Check your Effectivities and try again."
64         DataWkBk.Close Savechanges:=False
65         Kill (FolderPath & Line & ".xls")
66         Kill (FolderPath & Line & ".txt")
67         Exit Sub
68     End If

69     'Create workbook for raw data
70     Set RDWkBk = Workbooks.Add
71     RDWkBk.SaveAs FolderPath & "raw data.xlsm", FileFormat:=52
72     RDWkBk.Sheets(1).Name = "Raw_Data"

73     'Save as the Line data
74     Dim fs As Object
75     Set fs = CreateObject("Scripting.FileSystemObject")
76     fs.CopyFile FolderPath & PrgWkBk.Name, FolderPath & LineLetter & " Line Heat Maps " + Format(Date,
"MM-DD-YYYY") & ".xlsm"
77     Set fs = Nothing
78     Set FinalWkBk = Workbooks.Open(Filename:=FolderPath & LineLetter & " Line Heat Maps " + Format(Date,
"MM-DD-YYYY") & ".xlsm")

79     'Create Pivot table of Work Orders
80     Set PivotRange = DataWkBk.Sheets(Line).Range(Range(DataWkBk.Sheets(Line).Range("A1"),
DataWkBk.Sheets(Line).Range( _
    "A1").End(xlDown)), Range(DataWkBk.Sheets(Line).Range("A1"),
DataWkBk.Sheets(Line).Range("A1").End(xlDown)).End(xlToRight))

81     DataWkBk.Sheets.Add.Name = "Drivers"
82     DataWkBk.Sheets("Drivers").Activate
83     DataWkBk.PivotCaches.Create(SourceType:=xlDatabase, SourceData:= _
Chr(39) & Line & Chr(39) & "!" & PivotRange.Address, Version:=xlPivotTableVersion12).CreatePivotTable _
TableDestination:="Drivers!r3c1", TableName:="DriverPivot", DefaultVersion:=xlPivotTableVersion12

```

```

84 With DataWkBk.Sheets("Drivers").PivotTables("DriverPivot").PivotFields("Src Dept")
85     .Orientation = xlRowField
86     .Position = 1
87 End With
88 With DataWkBk.Sheets("Drivers").PivotTables("DriverPivot").PivotFields("Part No")
89     .Orientation = xlRowField
90     .Position = 2
91     .LayoutForm = xlTabular
92 End With
93 With DataWkBk.Sheets("Drivers").PivotTables("DriverPivot").PivotFields("Part Title")
94     .Orientation = xlRowField
95     .Position = 3
96     .LayoutForm = xlTabular
97 End With

98 Dim pt As PivotTable
99 Dim pf As PivotField

100 For Each pt In DataWkBk.Sheets("Drivers").PivotTables
101     For Each pf In pt.PivotFields
102         'First, set index 1 (Automatic) to True, so all other values are set to False
103         pf.Subtotals(1) = True
104         pf.Subtotals(1) = False
105     Next pf
106 Next pt

107 DataWkBk.Sheets("Drivers").PivotTables("DriverPivot").AddDataField ActiveSheet.PivotTables( _
"DriverPivot").PivotFields("Part No"), "Count of Part No", xlCount

108 With DataWkBk.Sheets("Drivers").PivotTables("DriverPivot").PivotFields("Part No")
109     .Orientation = xlRowField
110     .Position = 2
111     .LayoutForm = xlTabular
112 End With

113 'Create Raw Data and Driver Sheets and Copy/Paste Data
114 FinalWkBk.Sheets.Add(after:=FinalWkBk.Sheets("Control Sheet")).Name = "Top Drivers"
115 FinalWkBk.Sheets.Add(after:=FinalWkBk.Sheets("Top Drivers")).Name = "All_Raw_Data"

116 'CopyValues copies without monopolizing the clip board
117 Call CopyValues(DataWkBk.Sheets(Line).Range(Range(DataWkBk.Sheets(Line).Range("A1"),
DataWkBk.Sheets(Line).Range("A1").End(xlDown))), _
Range(DataWkBk.Sheets(Line).Range("A1"), DataWkBk.Sheets(Line).Range("A1").End(xlDown)).End(xlToRight)),
-
FinalWkBk.Sheets("All_Raw_Data").Range("A1"))

118 Call CopyValues(DataWkBk.Sheets(Line).Range(Range(DataWkBk.Sheets(Line).Range("A1"),
DataWkBk.Sheets(Line).Range("A1").End(xlDown))), _
Range(DataWkBk.Sheets(Line).Range("A1"), DataWkBk.Sheets(Line).Range("A1").End(xlDown)).End(xlToRight)),
RDWkBk.Sheets("Raw_Data"). _
Range("A1"))

119 RDWkBk.Save
120 RDWkBk.Close

121 Call CopyValues(DataWkBk.Sheets("Drivers").Range("A1:C" &
DataWkBk.Sheets("Drivers").Range("A:A").Find("Grand Total").Row), _
FinalWkBk.Sheets("Top Drivers").Range("A1"))

122 DataWkBk.Close

123 'Sorts the work orders by most to least defects by position
124 Call sortdrivers

125 NumWOs = FinalWkBk.Sheets("Top Drivers").Range("B" & FinalWkBk.Sheets("Top
Drivers").Range("A:A").Find(what:="Grand Total", _
LookIn:=xlValues, Lookat:=xlWhole).Row).Value

126 'Run analysis on each position and work order
127 Call RunPositions

128 FinalWkBk.Save

129 'Update Status Box
130 Status.WOComp.Value = "Running: Saving"
131 Status.Repaint

132 'Update Status Box
133 Status.WOComp.Value = "Running: Moving sheets"
134 Status.Repaint

```

```

135     'Delete unused sheets
136     For Each ws In FinalWkBk.Worksheets
137         If ws.Name = "All_Raw_Data" Or ws.Name = "Top Drivers" Or ws.Name = "Raw_Data" Or ws.Name Like _
            "**Heat Map" Or ws.Name Like "**HM" Or ws.Name Like "**Updated Data" Or ws.Name Like "**UD" _
            Or ws.Name = "Run Program" Or ws.Name = "Control Sheet" Then ws.Delete
138     Next

139     'Update Status Box
140     Status.WOComp.Value = "Running: Copying Buttons"
141     Status.Repaint

142     'Add Control Buttons
143     Call CopyButtons

144     'Update Status Box
145     Status.WOComp.Value = "Running: Making Plots"
146     Status.Repaint

147     'Add Location Maps
148     Call MakePlots

149     'Update Status Box
150     Status.WOComp.Value = "Running: Resetting Tool Sheets"
151     Status.Repaint

152     'Delete Data Sheets from program and save
153     Dim AllWkSht() As Variant

154     PrgWkBk.Sheets("Run Program").Activate
155     PrgWkBk.Save

156     'Protect sheets so that the user does not accidentally change cells when clicking to filter
157     Call ProtectSheets
158     Application.GoTo FinalWkBk.Sheets("Top Level").Range("A1"), True
159     FinalWkBk.Save

160     'Notify of Run Finish
161     Unload Status
162     EndTime = Timer
163     Application.ScreenUpdating = True
164     MsgBox ("Run complete in " & (EndTime - startTime) / 60 & " minutes.")

165     'Close program
166     PrgWkBk.Sheets("Run Program").Range("B11:B30").ClearContents
167     PrgWkBk.Close Savechanges:=True

168 End Sub

169 'Requires reference to Microsoft ActiveX Data Objects xx Library
170 Sub CopyAccTables()

171     Dim Cn As ADODB.Connection
172     Dim Rs As ADODB.Recordset, TableRs As ADODB.Recordset
173     Dim sSQL As String
174     Dim i As Integer
175     Dim iCol As Integer
176     Dim Rw As Long, Col As Long, c As Long
177     Dim MyField As ADODB.Field
178     Dim Loc As Range
179     Dim Tbl As String

180     Set Cn = New ADODB.Connection
181     Cn.Provider = "Microsoft.Jet.OLEDB.4.0"
182     Cn.Open (FolderPath & "HeatMapProgram.mdb")

183     Tbl = "Updated Data"
184     Set Loc = FinalWkBk.Sheets(UpdatedDataSheet).Cells(1, 1)

185     'Create query
186     sSQL = "SELECT * FROM [" & Tbl & "]"
187     Set Rs = Cn.Execute(sSQL)

188     'Write RecordSet to results area
189     Rw = Loc.Row
190     Col = Loc.Column
191     c = Col

192     'Get header row
193     For iCol = 1 To Rs.Fields.Count
194         FinalWkBk.Sheets(UpdatedDataSheet).Cells(1, iCol).Value = Rs.Fields(iCol - 1).Name
195     Next

```



```

196 FinalWkBk.Sheets(UpdatedDataSheet).Cells(2, 1).CopyFromRecordset Rs
197 FinalWkBk.Sheets(UpdatedDataSheet).Columns("A:A").Delete

198 Set Loc = Nothing
199 Cn.Close
200 Set Cn = Nothing
201 End Sub

202 Sub CopyButtons()
203 Dim w As Worksheet
204 Dim HMEndRow As Integer

205 For Each w In FinalWkBk.Worksheets

206     If w.Name <> "Run Program" And w.Name <> "Control Sheet" Then

207         w.Range("400:400").RowHeight = 15
208         HMEndRow = w.Range("A1").End(xlDown).Row

209         'Create buttons below Heat map
210         With w.Buttons.Add(w.Columns("A").Left, w.Rows(HMEndRow + 3).Top, 70, 17)
211             .OnAction = "" & FinalWkBk.Name & "!ToRawData"
212             .Characters.Text = "To Raw Data"
213         End With

214         With w.Buttons.Add(w.Columns("C").Left, w.Rows(HMEndRow + 3).Top, 64, 17)
215             .OnAction = "" & FinalWkBk.Name & "!ToACMap"
216             .Characters.Text = "To A/C Map"
217         End With

218         With w.Buttons.Add(w.Columns("E").Left, w.Rows(HMEndRow + 3).Top, 100, 17)
219             .OnAction = "" & FinalWkBk.Name & "!ClearFilterData"
220             .Characters.Text = "Clear Defect Filter"
221         End With

222         'Create buttons above raw data
223         With w.Buttons.Add(w.Range("C400").Left - 4, w.Rows(400).Top, 70, 17)
224             .OnAction = "" & FinalWkBk.Name & "!ToHeatMap"
225             .Characters.Text = "To Heat Map"
226         End With

227         With w.Buttons.Add(w.Range("E400").Left, w.Rows(400).Top, 64, 17)
228             .OnAction = "" & FinalWkBk.Name & "!ToACMap"
229             .Characters.Text = "To A/C Map"
230         End With

231         With w.Buttons.Add(w.Range("G400").Left, w.Rows("400").Top, 100, 17)
232             .OnAction = "" & FinalWkBk.Name & "!ClearFilterData"
233             .Characters.Text = "Clear Defect Filter"
234         End With

235         'Create buttons below AC map
236         With w.Buttons.Add(w.Range("AH40").Left - 3, w.Rows(40).Top, 70, 17)
237             .OnAction = "" & FinalWkBk.Name & "!ToRawData"
238             .Characters.Text = "To Raw Data"
239         End With

240         With w.Buttons.Add(w.Range("AJ40").Left - 3, w.Rows(40).Top, 70, 17)
241             .OnAction = "" & FinalWkBk.Name & "!ToHeatMap"
242             .Characters.Text = "To Heat Map"
243         End With

244         With w.Buttons.Add(w.Range("AL40").Left, w.Rows("40").Top, 100, 17)
245             .OnAction = "" & FinalWkBk.Name & "!ClearFilterData"
246             .Characters.Text = "Clear Defect Filter"
247         End With

248         With w.Buttons.Add(w.Columns("I").Left + 1, w.Rows(HMEndRow + 2).Top, 200, 30)
249             .OnAction = "" & FinalWkBk.Name & "!TimeScrolling"
250             .Characters.Text = "Enable Time Scrolling Controls"
251         End With

252         w.Range("A" & HMEndRow + 2) = "Double click the cells above to filter data by defect"
253         w.Range("S1:T2").Font.ColorIndex = 2
254         w.Range("T2").Value = 0
255     End If

256 Next w
257 End Sub

```

```

258 Sub CopyValues(rngSource As Range, rngTarget As Range)
259     Set rngTarget = rngTarget.Resize(rngSource.Rows.Count, rngSource.Columns.Count)
260     rngTarget.Value = rngSource.Value
261     If rngTarget.Cells(rngTarget.Rows.Count, 1) = "" And rngTarget.Cells(rngTarget.Rows.Count,
rngTarget.Columns.Count) = "" Then
262         If rngTarget.SpecialCells(xlCellTypeLastCell).End(xlUp).Row < rngSource.Cells(1, 1).Row +
rngSource.Rows.Count - 1 Then
263             rngSource.Copy
264             rngTarget.PasteSpecial
265         End If
266     End If
267 End Sub
268 Sub CreateHyperlinks()
269     'Create hyperlink lists of Positions and Work Orders on Top Level sheet
270     Dim CopyPos1 As String
271     Dim CopyPos2 As String
272     Dim j As Integer
273     Dim i As Integer
274     Dim k As Integer
275     Dim CopyPos1Row As Integer
276     Dim CopyPos1EndRow As Integer
277     Dim CopyCol As Integer
278     Dim CopyRow As Integer
279     Dim CopyRange As Range
280     Dim PasteCol As Integer
281     Dim jFirst As Boolean
282     If CurrentPos = "Top Level" Then
283         j = 1
284         jFirst = True
285     IfNotPos:
286     Do Until j > NumPositions
287         If j < 9 Then
288             CopyPos1 = "P0" & j & PosLetter
289             CopyPos2 = "P0" & j + 1 & PosLetter
290         ElseIf j = 9 Then
291             CopyPos1 = "P0" & j & PosLetter
292             CopyPos2 = "P" & j + 1 & PosLetter
293         ElseIf j > 9 Then
294             CopyPos1 = "P" & j & PosLetter
295             CopyPos2 = "P" & j + 1 & PosLetter
296         End If
297         If jFirst Then FirstPos = CopyPos1
298         'Checks if the position data starts with position 1
299         If FinalWkBk.Sheets("Top Drivers").Columns(1).Find(what:=CopyPos1, LookIn:=xlValues,
Lookat:=xlWhole) Is Nothing Then
300             j = j + 1
301             GoTo IfNotPos
302         End If
303         CopyPos1Row = FinalWkBk.Sheets("Top Drivers").Columns(1).Find(what:=CopyPos1, LookIn:=xlValues,
Lookat:=xlWhole).Row
304         If j < NumPositions Then
305             CopyPos1EndRow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:=CopyPos2,
LookIn:=xlValues, Lookat:=xlWhole). _
Row - 1
306         Else
307             If FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(blank)", LookIn:=xlValues,
Lookat:=xlPart) Is Nothing Then
308                 CopyPos1EndRow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(blank)",
LookIn:=xlValues, _
Lookat:=xlWhole).Row - 1
309             Else
310                 CopyPos1EndRow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(blank)",
LookIn:=xlValues, _
Lookat:=xlPart).Row - 1
311             End If
312         End If
313         'Copy from Drivers sheet to Top Level
314         If CopyPos1 = FirstPos Then k = 1
315         Call CopyValues(FinalWkBk.Sheets("Top Drivers").Range("A" & CopyPos1Row & ":C" & CopyPos1EndRow),
FinalWkBk.Sheets _
(TemplateSheet).Cells(endrow + 5, k * 3 - 2))

```



```

316      'Add Hyperlinks
317      FinalWkBk.Sheets(TemplateSheet).Cells(endrow + 5, k * 3 - 2).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Cells(endrow + 5, k * 3 - 2), Address:="", SubAddress:="" &
FinalWkBk.Sheets(TemplateSheet).Cells(endrow + 5, _
k * 3 - 2).Value & "'!A1"
318      For i = endrow + 6 To endrow + 6 + (CopyPos1EndRow - CopyPos1Row - 1)
319      If FinalWkBk.Sheets(TemplateSheet).Cells(i, k * 3 - 2).Value = "" Then
320      FinalWkBk.Sheets(TemplateSheet).Cells(i, k * 3 - 1).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Cells(i, k * 3 - 1), Address:="", SubAddress:="" & CopyPos1 & "-" &
FinalWkBk.Sheets(TemplateSheet). _
Cells(i - 1, k * 3 - 2).Value & "'!A1"
321      Else
322      FinalWkBk.Sheets(TemplateSheet).Cells(i, k * 3 - 2).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Cells(i, k * 3 - 2), Address:="", SubAddress:="" & CopyPos1 & "-" &
FinalWkBk.Sheets(TemplateSheet). _
Cells(i, k * 3 - 2).Value & "'!A1"
323      FinalWkBk.Sheets(TemplateSheet).Cells(i, k * 3 - 1).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Cells(i, k * 3 - 1), Address:="", SubAddress:="" & CopyPos1 & "-" &
FinalWkBk.Sheets(TemplateSheet). _
Cells(i, k * 3 - 2).Value & "'!A1"
324      End If
325      Next i
326      j = j + 1
327      k = k + 1
328      jFirst = False

329      Loop
330      End If

331      If WrkingSht Like "P##" & PosLetter Then
332      'On Positions copy Work Order lists from Top level sheet
333      CopyCol = FinalWkBk.Sheets("Top Level").Range("A1:AE399").Find(what:=CurrentPos,
Lookat:=xlWhole).Column
334      CopyRow = FinalWkBk.Sheets("Top Level").Range("A1:AE399").Find(what:=CurrentPos,
Lookat:=xlWhole).Row + 1
335      CopyEndRow = FinalWkBk.Sheets("Top Level").Range("A1:AE399").Find(what:=CurrentPos,
Lookat:=xlWhole).End(xlDown).Row

336      'Add link to top level
337      Call CopyValues(FinalWkBk.Sheets("Top Level").Range(Cells(CopyRow, CopyCol).Address & ":" &
Cells(CopyEndRow, CopyCol + 2). _
Address), FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 6))
338      For i = endrow + 6 To FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 6).End(xlDown).Row
339      FinalWkBk.Sheets(TemplateSheet).Cells(i, 1).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Cells(i, 1), Address:="", SubAddress:="" & FinalWkBk.Sheets(TemplateSheet).Name & "-" &
FinalWkBk.Sheets(TemplateSheet). _
Cells(i, 1).Value & "'!A1"
340      FinalWkBk.Sheets(TemplateSheet).Cells(i, 2).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Cells(i, 2), Address:="", SubAddress:="" & FinalWkBk.Sheets(TemplateSheet).Name & "-" &
FinalWkBk.Sheets(TemplateSheet). _
Cells(i, 1).Value & "'!A1"
341      Next i

342      FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 5).Value = "Top Level"
343      FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 5).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Range("A" & endrow + 5), Address:="", SubAddress:="" & "Top Level"!A1"
344      End If

345      If WrkingSht Like "P##" & PosLetter & "-" Then
346      'On Work Orders Add position and top level links
347      FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 5).Value = "Top Level"
348      FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 5).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Range("A" & endrow + 5), Address:="", SubAddress:="" & "Top Level"!A1"
349      FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 6).Value = "Position " + Mid(CurrentPos, 2, 2)
350      FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 5).Hyperlinks.Add
anchor:=FinalWkBk.Sheets(TemplateSheet). _
Range("A" & endrow + 6), Address:="", SubAddress:="" & CurrentPos & "'!A1"
351      End If
352      End Sub

353      Sub CreatePareto()

354      Dim rangeX As String

```

```

355 Dim rngY As String
356 Dim i As Integer
357 Dim j As Integer
358 Dim lastrow As Integer
359 Dim endrow As Integer
360 Dim ColLetter As String
361 Dim wsht As String
362 Dim Mode As String
363 Dim Part As String

364 FinalWkBk.Sheets(TemplateSheet).Cells(1, 1) = "Part"
365 lastrow = WorksheetFunction.CountA(FinalWkBk.Sheets(TemplateSheet).Range("A401:A40000")) + 400
366 wsht = Replace(Replace(FinalWkBk.Sheets(TemplateSheet).Name, " ", "_"), "-", "_")

367 'Insert Row Labels
368 For i = 1 To 16
369     FinalWkBk.Sheets(TemplateSheet).Cells(1, i + 1) = FinalWkBk.Sheets(TemplateSheet).Cells(401, 25 +
i)
370 Next i
371 FinalWkBk.Sheets(TemplateSheet).Range("R1") = "Total"

372 'Insert column labels
373 j = 2
374 For i = 1 To 28
375     If FinalWkBk.Sheets(TemplateSheet).Cells(401, 41 + i) <> "Rejected" And
FinalWkBk.Sheets(TemplateSheet).Cells(401, 41 + i) <> "RRO" And FinalWkBk.Sheets(TemplateSheet).Cells(401, 41 +
i) <> "YSAT" Then
376         FinalWkBk.Sheets(TemplateSheet).Cells(j, 1) = FinalWkBk.Sheets(TemplateSheet).Cells(401, 41 + i)
377         j = j + 1
378     End If
379 Next i

380 'Create Named Ranges for data columns
381 For i = 26 To 70
382     'Change column numbers to letters to use in ranges
383     If i <= 26 Then
384         'Columns A-Z
385         ColLetter = Chr(i + 64)
386     Else
387         'Columns Z to ZZ
388         ColLetter = Chr(Int((i - 1) / 26) + 64) & _
Chr(((i - 1) Mod 26) + 65)
389     End If
390     FinalWkBk.Sheets(TemplateSheet).Names.Add Name:=wsht &
Replace(Replace(Replace(FinalWkBk.Sheets(TemplateSheet).
Cells(401, i).Text, " ", "_"), "( ", "_"), ")", "_"), RefersTo:=FinalWkBk.Sheets(TemplateSheet).Range _
(ColLetter & "401:" & ColLetter & lastrow)
391 Next i

392     FinalWkBk.Sheets(TemplateSheet).Names.Add Name:=wsht & "Effectivity",
RefersTo:=FinalWkBk.Sheets(TemplateSheet).Range _
("B401:B" & lastrow)

393     endrow = FinalWkBk.Sheets(TemplateSheet).Range("A1").End(xlDown).Row
394     FinalWkBk.Sheets(TemplateSheet).Range("A" & endrow + 1) = "Total"
395     endrow = endrow + 1

396     'Insert counts for cross area
397     For i = 2 To endrow - 1

398         Part = Replace(Replace(Replace(FinalWkBk.Sheets(TemplateSheet).Cells(i, 1).Text, " ", "_"), "( ",
" _"), ")", " _")
399         FinalWkBk.Sheets(TemplateSheet).Cells(i, 18) = "=IF($S$2=" & Chr(34) & Chr(34) & ",
SUMPRODUCT((1)*(" & wsht & Part & "=" & Chr(34) &
"X" & Chr(34) & ")), SUMPRODUCT((" & wsht & "Effectivity=$S$2)*(" & wsht & Part & "=" & Chr(34) &
"X" -
& Chr(34) & ")))"

400     For j = 2 To 17
401         Mode = Replace(Replace(Replace(FinalWkBk.Sheets(TemplateSheet).Cells(1, j).Text, " ", "_"), "( ",
" _"), ")", " _")
402         If i = 2 Then
403             FinalWkBk.Sheets(TemplateSheet).Cells(endrow, j) = "=IF($S$2=" & Chr(34) & Chr(34) & ",
SUMPRODUCT((1)*(" & wsht & Mode & "=" & Chr(34) &
"X" & Chr(34) & ")), SUMPRODUCT((" & wsht & "Effectivity=$S$2)*(" & wsht & Mode & "=" &
Chr(34) & "X" -
& Chr(34) & ")))"
404         End If
405         FinalWkBk.Sheets(TemplateSheet).Cells(i, j) = "=IF($S$2=" & Chr(34) & Chr(34) & ",
SUMPRODUCT((1)*(" & wsht & Part & "=" & Chr(34) &

```

```

"X" & Chr(34) & ")*((" & wsht & Mode & "=" & Chr(34) & "X" & Chr(34) & ")), SUMPRODUCT((" & wsht _
& "Effectivity=$S$2)*(" & wsht & Part & "=" & Chr(34) & "X" & Chr(34) & ")*(" & wsht & Mode _
& "=" & Chr(34) & "X" & Chr(34) & ")))"
406     Next j
407     Next i

408     FinalWkBk.Sheets(TemplateSheet).Range("R" & endrow) = "=IF($S$2=" & Chr(34) & Chr(34) & ",
CountA(B402:B" & lastrow & _
",), Countif(B402:B" & lastrow & ", $S$2))"

409     'Delete rows that have no data
410     i = 2
411     j = 0
412     Do While i <= endrow
413         If FinalWkBk.Sheets(TemplateSheet).Cells(i, 18).Value = 0 And Cells(i, 1) <> "" Then
414             FinalWkBk.Sheets(TemplateSheet).Range("A" & i & ":R" & i).Delete Shift:=xlUp
415             i = i - 1
416             endrow = endrow - 1
417             j = j + 1
418         End If
419         i = i + 1
420     Loop

421     'Add deleted rows back to keep Raw Data in the right place
422     i = 1
423     For i = 1 To j
424         FinalWkBk.Sheets(TemplateSheet).Range("A" & 40 + i & ":R" & 40 + i).Insert Shift:=xlDown
425     Next i

426 End Sub

427 Sub Main()
428     Dim NoEfff As Integer
429     Dim i As Integer
430     Dim strTerminateThis As String
431     Dim objWMIcimv2 As Object
432     Dim objProcess As Object
433     Dim objList As Object
434     Dim objCount As Integer
435     Dim msgboxval As Integer

436     Application.IgnoreRemoteRequests = True
437     Application.ScreenUpdating = False
438     Application.DisplayAlerts = False
439     Application.WindowState = xlMinimized

440     ''Makesure the user has entered effectivities
441     NoEfff = 0
442     For i = 11 To 30
443         If ActiveSheet.Range("B" & i).Value <> "" Then NoEfff = 1
444         If NoEfff = 1 Then i = 30
445     Next i

446     If NoEfff = 0 Then
447         MsgBox "First enter at least one effectivity"
448         Exit Sub
449     End If

450     'Time the run to display run time at end
451     startTime = Timer

452     ''Show and Update Status Box
453     Status.Show False
454     Status.WOComp.Value = "Running: Preparing Workbooks"
455     Status.Repaint

456     Call PrepareBook

457 End Sub

458 Sub MakePlots()
459     Dim StaWLRRange As String
460     Dim StaBLRange As String
461     Dim StaBLChart As Chart
462     Dim StaWLChart As Chart
463     Dim w As Worksheet

464     For Each w In FinalWkBk.Worksheets
465         'For all data sheets add A/C images and location for defect plots
466         If w.Name <> "Run Program" And w.Name <> "Control Sheet" Then
467             'Zoom to fit heat map

```



```

468 Application.GoTo w.Range("$A$1:$S$1")
469 ActiveWindow.Zoom = True
470 'Insert A/C Images
471 w.Pictures.Insert(FolderPath & "StaWlMap.jpg").Name = "StaWlImage"
472 w.Shapes("StaWlImage").Top = w.Rows(1).Top
473 w.Shapes("StaWlImage").Left = w.Columns("AH").Left
474 w.Pictures.Insert(FolderPath & "StaBlMap.jpg").Name = "StaBlImage"
475 w.Shapes("StaBlImage").Top = w.Rows(24).Top
476 w.Shapes("StaBlImage").Left = w.Columns("AH").Left
477 'Set plot making ranges
478 StaWlRange = "H401:I" & w.Range("A401").End(xlDown).Row
479 StaBlRange = "H401:H" & w.Range("A401").End(xlDown).Row & ", J401:J" &
w.Range("A401").End(xlDown).Row

480 'Add and format charts
481 Set StaWlChart = w.ChartObjects.Add(1, 1, 500, 500).Chart
482 StaWlChart.ChartType = xlXYScatter
483 StaWlChart.SetSourceData Source:=w.Range(StaWlRange), PlotBy:=xlColumns

484 With StaWlChart
485 .Legend.Delete
486 .Parent.Top = w.Rows(1).Top
487 .Parent.Left = w.Columns("AH").Left
488 .Parent.Height = 4.2 * 72
489 .Parent.Width = 10 * 72
490 .Axes(xlCategory).MinimumScale = 160
491 .Axes(xlCategory).MaximumScale = 800
492 .Axes(xlValue).MinimumScale = 160
493 .Axes(xlValue).MaximumScale = 340
494 .ChartArea.Fill.Visible = False
495 .PlotArea.Fill.Visible = False
496 .Axes(xlValue).MajorGridlines.Delete
497 .Axes(xlValue).Delete
498 .Axes(xlCategory).Delete
499 .ChartTitle.Delete
500 .SeriesCollection(1).MarkerBackgroundColorIndex = 3
501 .SeriesCollection(1).MarkerForegroundColorIndex = 3
502 .SeriesCollection(1).MarkerStyle = xlCircle
503 .SeriesCollection(1).MarkerSize = 3
504 End With

505 StaWlChart.PlotArea.Select
506 With Selection
507 .Height = 1.79 * 72
508 .Width = StaWlChart.PlotArea.Height * 3.72
509 .Left = 147
510 .Top = 85
511 End With

512 Set StaBlChart = w.ChartObjects.Add(1, 1, 500, 500).Chart
513 StaBlChart.ChartType = xlXYScatter
514 StaBlChart.SetSourceData Source:=w.Range(StaBlRange), PlotBy:=xlColumns
515 With StaBlChart
516 .Legend.Delete
517 .Parent.Top = w.Rows(24).Top
518 .Parent.Left = w.Columns("AH").Left
519 .Parent.Height = 2.95 * 72
520 .Parent.Width = 10 * 72
521 .Axes(xlCategory).MinimumScale = 160
522 .Axes(xlCategory).MaximumScale = 800
523 .Axes(xlValue).MinimumScale = -90
524 .Axes(xlValue).MaximumScale = 90
525 .ChartArea.Fill.Visible = False
526 .PlotArea.Fill.Visible = False
527 .Axes(xlValue).MajorGridlines.Delete
528 .Axes(xlValue).Delete
529 .Axes(xlCategory).Delete
530 .ChartTitle.Delete
531 .SeriesCollection(1).MarkerBackgroundColorIndex = 3
532 .SeriesCollection(1).MarkerForegroundColorIndex = 3
533 .SeriesCollection(1).MarkerStyle = xlCircle
534 .SeriesCollection(1).MarkerSize = 3
535 End With

536 StaBlChart.PlotArea.Select
537 With Selection
538 .Height = 152
539 .Top = 4
540 .Width = StaBlChart.PlotArea.Height * 3.154
541 .Left = 148
542 End With

```

```

543     End If
544     Next w
545 End Sub

546 Sub ParetoSort()

547     Dim insertrow As Integer

548     'Sort total column
549     insertrow = FinalWkBk.Sheets(TemplateSheet).Range("A1").End(xlDown).Row
550     FinalWkBk.Sheets(TemplateSheet).Rows(insertrow).Insert Shift:=xlDown,
CopyOrigin:=xlFormatFromLeftOrAbove
551     FinalWkBk.Sheets(TemplateSheet).Sort.SortFields.Clear
552     FinalWkBk.Sheets(TemplateSheet).Sort.SortFields.Add Key:=Range("R2:R" & insertrow - 1),
SortOn:=xlSortOnValues, _
Order:=xlDescending, DataOption:=xlSortNormal
553     With FinalWkBk.Sheets(TemplateSheet).Sort
554         .SetRange FinalWkBk.Sheets(TemplateSheet).Range("A1:R" & insertrow - 1)
555         .Header = xlYes
556         .MatchCase = False
557         .Orientation = xlTopToBottom
558         .SortMethod = xlPinYin
559         .Apply
560     End With

561     'Sort total Row
562     FinalWkBk.Sheets(TemplateSheet).Rows(insertrow).Delete Shift:=xlUp
563     FinalWkBk.Sheets(TemplateSheet).Columns("R:R").Insert Shift:=xlToRight,
CopyOrigin:=xlFormatFromLeftOrAbove
564     FinalWkBk.Sheets(TemplateSheet).Sort.SortFields.Clear
565     FinalWkBk.Sheets(TemplateSheet).Sort.SortFields.Add Key:= _
FinalWkBk.Sheets(TemplateSheet).Range("A" & insertrow & ":Q" & insertrow), SortOn:=xlSortOnValues, _
Order:=xlDescending, DataOption:=xlSortNormal
566     With FinalWkBk.Sheets(TemplateSheet).Sort
567         .SetRange FinalWkBk.Sheets(TemplateSheet).Range("A1:Q" & insertrow)
568         .Header = xlYes
569         .MatchCase = False
570         .Orientation = xlLeftToRight
571         .SortMethod = xlPinYin
572         .Apply
573     End With
574     FinalWkBk.Sheets(TemplateSheet).Columns("R:R").Delete Shift:=xlToLeft

575 End Sub

576 Sub PrepareBook()

577     'Collects user input to create variables for use throughout tool creation.

578     Set PrgWkBk = ActiveWorkbook

579     FolderPath = PrgWkBk.Path + "\"

580     If PrgWkBk.Sheets("Run Program").Range("B9") = 1 Then
581         LineLetter = "A"
582         PosLetter = "A"
583         NumPositions = 8
584         Line = "AA-12"
585     ElseIf PrgWkBk.Sheets("Run Program").Range("B9") = 2 Then
586         LineLetter = "B"
587         PosLetter = "B"
588         NumPositions = 4
589         Line = "BB-34"
590     ElseIf PrgWkBk.Sheets("Run Program").Range("B9") = 3 Then
591         LineLetter = "C"
592         PosLetter = "C"
593         NumPositions = 6
594         Line = "CC-56"
595     ElseIf PrgWkBk.Sheets("Run Program").Range("B9") = 4 Then
596         LineLetter = "C"
597         PosLetter = "C"
598         NumPositions = 7
599         Line = "CC-78"
600     Else
601         MsgBox ("Please Choose a Line then click Run again")
602         Unload Status
603         Exit Sub
604     End If

605     'Update Status Box
606     Status.WOComp.Value = "Running: Collecting Data from Hyperion"

```

```

607     Status.Repaint

608     'Goes to the Get_Hyperion_Data module to query Hyperion
609     Call process_file

610 End Sub

611 Sub ProtectSheets()
612     Dim ws As Worksheet

613     'Protect cells from being changed on double click for filtering
614     For Each ws In FinalWkBk.Worksheets
615         If ws.Name <> "Run Program" And ws.Name <> "Control Sheet" Then
616             ws.Protect UserInterfaceOnly:=True, DrawingObjects:=True, Contents:=True, Scenarios:=True, _
                AllowSorting:=True, AllowFiltering:=True
617         End If
618     Next
619 End Sub

620 Sub RunPositions()

621     Dim i As Integer

622     'Open Access
623     On Error Resume Next

624     Dim LPath As String
625     Dim LCategoryID As Long
626     Dim DATABASE As String

627     DATABASE = FolderPath & "DDRPHeatMapProgram.mdb"

628     Set oApp = GetObject(, "Access.Application")
629     If (Err.Number <> 0) Or (oApp.CurrentDb.Name <> DATABASE) Then
630         Set oApp = Nothing
631         Set oApp = CreateObject("Access.Application")
632         oApp.Visible = False
633         oApp.OpenCurrentDatabase DATABASE
634     End If

635     'Update Status Box
636     Status.WOComp.Value = "Running: Beginning Work Order analysis"
637     Status.Repaint

638     'Create Top Level Data Sheet
639     FinalWkBk.Sheets.Add(after:=FinalWkBk.Sheets("All_Raw_Data")).Name = "Top Level"
640     CurrentPos = "Top Level"

641     'Create temporary data sheets
642     FinalWkBk.Sheets.Add(after:=FinalWkBk.Sheets("Top Level")).Name = "Top Level Updated Data"
643     UpdatedDataSheet = "Top Level Updated Data"

644     'Run Access Code
645     oApp.Run "AutoCode"
646     'Copy the Access tables to Excel
647     Call CopyAccTables

648     'Close Access
649     oApp.Run "Clearstuff"
650     oApp.CloseCurrentDatabase

651     'Close and delete the Raw Data files
652     If Not RDWkBk Is Nothing Then RDWkBk.Close Savechanges:=False
653     Kill (FolderPath & "raw data.xlsm")
654     Kill (FolderPath & Line & ".xls")
655     SetAttr FolderPath & Line & ".txt", vbNormal
656     Kill (FolderPath & Line & ".txt")

657     WrkingSht = "Top Level"

658     'Put the data in a standard format
659     Call UpdateTemplate

660     'Delete temporary Sheets
661     FinalWkBk.Sheets("Top Level Updated Data").Delete
662     LastSheet = "Top Level"

663     'Update Status Box
664     CompleteWOs = 1
665     Status.WOComp.Value = "Running: WO " & CompleteWOs & " of " & NumWOs & " complete."
666     Status.Repaint

```



```

667     'Create Position and Work Order Data Sheets
668     i = 1
669     Do Until i > NumPositions

670         'Set Current Position, Positions are named with 0 before single digits, this accounts for that
671         If i <= 9 Then
672             CurrentPos = "P0" & i & PosLetter
673         Else
674             CurrentPos = "P" & i & PosLetter
675         End If
676         CurrentPosNumber = i

677         'Filter Raw Data to Position
678         FinalWkBk.Sheets("Top Level").Range("$C$401:$C$1000000").AdvancedFilter action:=xlFilterCopy,
Field:=3, CriteriaRange:=CurrentPos, _
CopyToRange:=FinalWkBk.Sheets(CurrentPos).Range("A401")

679         'Check to make sure position has data
680         FinalWkBk.Sheets("Top Level").Range("$A$401:$BU$401").AutoFilter Field:=3, Criterial:=CurrentPos
681         If FinalWkBk.Sheets("Top Level").Range("A402:A1000000").SpecialCells(xlCellTypeVisible).Row >
WorksheetFunction.CountA(FinalWkBk.Sheets("Top Level").Range("A401:A40000")) + 400 Then
682             CurrentPos = "Top Level"
683         Else
684             'Make position sheet
685             FinalWkBk.Sheets.Add(after:=FinalWkBk.Sheets(LastSheet)).Name = CurrentPos
686             'Copy data
687             FinalWkBk.Sheets(CurrentPos).Range("A1").Value = FinalWkBk.Sheets("Top
Level").Range("C401").Value
688             FinalWkBk.Sheets(CurrentPos).Range("A2").Value = "=" & CurrentPos
689             FinalWkBk.Sheets("Top Level").Range("A401:BU40000").AdvancedFilter action:=xlFilterCopy,
CriteriaRange:=FinalWkBk.Sheets(CurrentPos).Range("A1:A2"), _
CopyToRange:=FinalWkBk.Sheets(CurrentPos).Range("A401")
690             WrkingSht = CurrentPos

691             'Put the data in a standard format
692             Call UpdateTemplate

693             If CurrentPos = FirstPos Then
694                 LastSheet = CurrentPos
695             Else
696                 LastSheet = CurrentPosWO
697             End If

698             'Update Status Box
699             CompleteWOs = CompleteWOs + 1
700             Status.WOComp.Value = "Running: Work Order " & CompleteWOs & " of " & NumWOs & " complete."
701             Status.Repaint

702             'Perform analysis on individual Work Orders
703             Call RunWOs
704             LastSheet = CurrentPosWO

705         End If

706         'Clear filters
707         FinalWkBk.Sheets("Top Level").Range("$A$401:$BU$401").AutoFilter
708         i = i + 1
709         LastSheet = CurrentPos

710     Loop
711 End Sub

712 Sub RunWOs()

713     Dim CurrentWO As String
714     Dim PosLoc As Range
715     Dim WOCol As Integer
716     Dim WORow As Integer
717     Dim i As Integer
718     Dim NextPos As String
719     Dim NextPosNumber As Integer
720     Dim EndPosLoc As Range
721     Dim EndWORow As Integer
722     Dim WOCCount As Integer

723     NextPosNumber = CurrentPosNumber + 1

724     'Find Work Orders that pertain to current position in Drivers list, Positions are named with 0
before single digits, this accounts for that

```

```

725     If NextPosNumber <= 9 Then
726         NextPos = "P0" & NextPosNumber & PosLetter
727     Else
728         NextPos = "P" & NextPosNumber & PosLetter
729     End If

730     Set PosLoc = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:=CurrentPos, LookIn:=xlValues,
Lookat:=xlWhole)
731     WOCol = PosLoc.Column
732     WORow = PosLoc.Row + 1
733     CurrentWO = FinalWkBk.Sheets("Top Drivers").Cells(WORow, WOCol).Value
734     CurrentPosWO = CurrentPos + "-" + CurrentWO

735     If CurrentPosNumber < NumPositions Then
736         EndWORow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:=NextPos, LookIn:=xlValues,
Lookat:=xlWhole).Row
737     Else
738         If Range("A:A").Find(what:="(blank)", LookIn:=xlValues, Lookat:=xlPart) Is Nothing Then
739             EndWORow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="Grand Total",
LookIn:=xlValues, Lookat:=xlWhole).Row
740         Else
741             EndWORow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(blank)", LookIn:=xlValues,
Lookat:=xlPart).Row
742         End If
743     End If

744     'Number of Work Orders in current position
745     WOCCount = EndWORow - WORow

746     i = 1
747     Do Until i > WOCCount
748         'Filter Data to Work Order
749         FinalWkBk.Sheets("Top Level").Range("$A$401:$BU$401").AutoFilter Field:=3, Criterial:=CurrentPos
750         FinalWkBk.Sheets("Top Level").Range("$A$401:$BU$401").AutoFilter Field:=5, Criterial:=CurrentWO
751         'Check to make sure Work Order has Data
752         If FinalWkBk.Sheets("Top Level").Range("A402:A1000000").SpecialCells(xlCellTypeVisible).Row <=
WorksheetFunction.CountA
(FinalWkBk.Sheets("Top Level").Range("A401:A400000")) + 400 Then
753             'Make sheet
754             FinalWkBk.Sheets.Add(after:=FinalWkBk.Sheets(LastSheet)).Name = CurrentPosWO
755             'Copy Data
756             FinalWkBk.Sheets(CurrentPosWO).Range("A1").Value = FinalWkBk.Sheets("Top
Level").Range("C401").Value
757             FinalWkBk.Sheets(CurrentPosWO).Range("A2").Value = "=" & CurrentPos
758             FinalWkBk.Sheets(CurrentPosWO).Range("B1").Value = FinalWkBk.Sheets("Top
Level").Range("E401").Value
759             FinalWkBk.Sheets(CurrentPosWO).Range("B2").Value = "=" & CurrentWO
760             FinalWkBk.Sheets("Top Level").Range("A401:BU400000").AdvancedFilter action:=xlFilterCopy,
CriteriaRange:=FinalWkBk.Sheets(CurrentPosWO).Range("A1:B2"),
CopyToRange:=FinalWkBk.Sheets(CurrentPosWO).Range("A401")

761             WrkingSht = CurrentPosWO
762             'Put the data in a standard format
763             Call UpdateTemplate
764             LastSheet = CurrentPosWO
765         End If

766         'Go to next row in Work Order list
767         WORow = WORow + 1

768         'Pass over blank Work Order cells
769         Do While FinalWkBk.Sheets("Top Drivers").Cells(WORow, WOCol).Value = ""
770             i = i + 1
771             WORow = WORow + 1
772         Loop

773         'Go to next Work Order
774         CurrentWO = FinalWkBk.Sheets("Top Drivers").Cells(WORow, WOCol).Value
775         CurrentPosWO = CurrentPos + "-" + CurrentWO
776         i = i + 1
777         CompleteWOs = CompleteWOs + 1

778         ''Update Status
779         Status.WOComp.Value = "Running: WO " & CompleteWOs & " of " & NumWOs & " complete."
780         Status.Repaint
781     Loop

782     CurrentPosWO = LastSheet
783     If WOCCount = 0 Then LastSheet = CurrentPos

784 End Sub

```



```

785 Sub sortdrivers()

786 Dim endrow As Integer
787 Dim j As Integer
788 Dim i As Integer
789 Dim RangePos1 As String
790 Dim RangePos2 As String
791 Dim RangePos1Row As Integer
792 Dim RangePos1EndRow As Integer
793 Dim iFirst As Boolean

794 'Find the last used row
795 endrow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find("Grand Total", Lookat:=xlWhole).Row

796 'Use only the most common part title and delete other part titles
797 For i = 3 To endrow
798     If FinalWkBk.Sheets("Top Drivers").Range("A" & i).Value = "" Then
799         j = FinalWkBk.Sheets("Top Drivers").Range("A" & i).End(xlUp).Row
800         If FinalWkBk.Sheets("Top Drivers").Range("C" & i).Value > FinalWkBk.Sheets("Top
Drivers").Range("C" & j).Value Then
801             FinalWkBk.Sheets("Top Drivers").Range("B" & j).Value = FinalWkBk.Sheets("Top
Drivers").Range("B" & i).Value
802             End If
803             FinalWkBk.Sheets("Top Drivers").Range("C" & j).Value = FinalWkBk.Sheets("Top Drivers").Range("C"
& i).Value + FinalWkBk.
Sheets("Top Drivers").Range("C" & j).Value
804             FinalWkBk.Sheets("Top Drivers").Range("C" & i).EntireRow.Delete
805             If FinalWkBk.Sheets("Top Drivers").Range("A" & j).Value <> "Grand Total" Then
806                 endrow = endrow - 1
807                 i = i - 1
808             End If
809         End If
810     Next i

811     i = 1
812     iFirst = True
813     IfNoPos:
814     Do Until i > NumPositions
815         'Find the beginning and end of the position list
816         If i < 9 Then
817             RangePos1 = "P0" & i & PosLetter
818             RangePos2 = "P0" & i + 1 & PosLetter
819         ElseIf i = 9 Then
820             RangePos1 = "P0" & i & PosLetter
821             RangePos2 = "P" & i + 1 & PosLetter
822         ElseIf i > 9 Then
823             RangePos1 = "P" & i & PosLetter
824             RangePos2 = "P" & i + 1 & PosLetter
825         End If
826         If iFirst Then FirstPos = RangePos1
827         If FinalWkBk.Sheets("Top Drivers").Columns(1).Find(what:=RangePos1, LookIn:=xlValues,
Lookat:=xlWhole) Is Nothing Then
828             i = i + 1
829             'Checks for data
830             GoTo IfNoPos
831         End If

832         RangePos1Row = FinalWkBk.Sheets("Top Drivers").Columns(1).Find(what:=RangePos1, LookIn:=xlValues,
Lookat:=xlWhole).Row
833         If i < NumPositions Then
834             RangePos1EndRow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:=RangePos2,
LookIn:=xlValues, Lookat:=xlWhole).
Row - 1
835         Else
836             If FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(blank)", LookIn:=xlValues,
Lookat:=xlPart) Is Nothing Then
837                 RangePos1EndRow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(Grand Total",
LookIn:=xlValues,
Lookat:=xlWhole).Row - 1
838             Else
839                 RangePos1EndRow = FinalWkBk.Sheets("Top Drivers").Range("A:A").Find(what:="(blank)",
LookIn:=xlValues,
Lookat:=xlPart).Row - 1
840             End If
841         End If
842         'Sort Work Orders most to least common
843         FinalWkBk.Sheets("Top Drivers").Range("A" & RangePos1Row + 1 & ":C" & RangePos1EndRow).Sort _
Key1:=FinalWkBk.Sheets("Top Drivers").Range("C" & RangePos1Row + 1 & ":C" & RangePos1EndRow),
Order1:=xlDescending, _
Key2:=FinalWkBk.Sheets("Top Drivers").Range("A" & RangePos1Row + 1 & ":A" & RangePos1EndRow),

```

```

Order2:=xlAscending, _
    Header:=xlNo
844     i = i + 1
845     iFirst = False
846     Loop

847     'Count the number of positions and Work Orders
848     FinalWkBk.Sheets("Top Drivers").Range("B" & endrow).Value = "=COUNTA(A" & FinalWkBk.Sheets("Top
Drivers").Range("A:A").Find(what:=FirstPos, LookIn:=xlValues, Lookat:=xlWhole).Row & ":A" & endrow & ")"
849     NumWOs = FinalWkBk.Sheets("Top Drivers").Range("B" & endrow).Value

850 End Sub

851 Sub UpdateTemplate()

852     Dim GoToSheet As String
853     Dim ZoomRange As String
854     Dim c As Range
855     Dim CP As String
856     Dim endrow As Integer
857     Dim CopyEndRow As Integer

858     TemplateSheet = WrkingSht

859     'Format Updated data on Top Level Sheet
860     If WrkingSht = "Top Level" Then
861         'Copy from updated data sheet to Top level data sheet
862         GoToSheet = CurrentPos + " Updated Data"
863         Call CopyValues(FinalWkBk.Sheets(GoToSheet).Range(Range(FinalWkBk.Sheets(GoToSheet).Range("A1"),
FinalWkBk.Sheets(GoToSheet).Range("A1"). _
End(xlDown)), Range(FinalWkBk.Sheets(GoToSheet).Range("A1"),
FinalWkBk.Sheets(GoToSheet).Range("A1").End(xlDown)). _
End(xlToRight)), FinalWkBk.Sheets(TemplateSheet).Range("A401"))

864         'Format borders and fill
865         With FinalWkBk.Sheets(TemplateSheet).Range(Range(FinalWkBk.Sheets(TemplateSheet).Range("A401"),
FinalWkBk.Sheets(TemplateSheet). _
Range("A4011").End(xlDown)), Range(FinalWkBk.Sheets(TemplateSheet).Range("A401"),
FinalWkBk.Sheets(TemplateSheet). _
Range("A401").End(xlDown)).End(xlToRight))
866             .Borders.LineStyle = xlContinuous
867             .Borders.Weight = xlThin
868             .Borders.ColorIndex = 15
869         End With
870         FinalWkBk.Sheets(TemplateSheet).Range("A401:BU401").Interior.ColorIndex = 15
871         With FinalWkBk.Sheets(TemplateSheet).Range("A401:BU401")
872             .Borders.LineStyle = xlContinuous
873             .Borders.Weight = xlThin
874             .Borders.ColorIndex = 1
875         End With
876         'Change to date format
877         FinalWkBk.Sheets(TemplateSheet).Range("N401:N" &
FinalWkBk.Sheets(TemplateSheet).Range("A401").End(xlDown).Row). _
NumberFormat = "[$-409]m/d/yy h:mm AM/PM;@"
878         FinalWkBk.Sheets(TemplateSheet).Range("R401:R" &
FinalWkBk.Sheets(TemplateSheet).Range("A401").End(xlDown).Row). _
NumberFormat = "[$-409]m/d/yy h:mm AM/PM;@"
879         FinalWkBk.Sheets(TemplateSheet).Range("S401:S" &
FinalWkBk.Sheets(TemplateSheet).Range("A401").End(xlDown).Row). _
NumberFormat = "[$-409]m/d/yy h:mm AM/PM;@"
880     End If

881     'Create the Cross Pareto Table
882     Call CreatePareto

883     'If there are two dimensions to the Pareto create the heat map, Some Work Orders only have a totals
row
884     If FinalWkBk.Sheets(TemplateSheet).Range("A2").Value <> "Total" Then
885         Call ConditionalColors
886     End If

887     'Add borders and color formatting to heat map
888     endrow = FinalWkBk.Sheets(TemplateSheet).Range("A1").End(xlDown).Row
889     With FinalWkBk.Sheets(TemplateSheet).Range("$A$2:$R$" & endrow)
890         .Borders.LineStyle = xlContinuous
891         .Borders.Weight = xlThin
892         .Borders.ColorIndex = 15
893     End With

894     With FinalWkBk.Sheets(TemplateSheet).Range("A1:R1")
895         .Interior.ColorIndex = 15

```

```

896     .Borders.LineStyle = xlContinuous
897     .Borders.Weight = xlThin
898     .Borders.ColorIndex = 1
899 End With

900 'Create hyperlink lists of Positions and Work Orders on Top Level sheet
901 Call CreateHyperlinks

902 FinalWkBk.Sheets(TemplateSheet).Range("A400").Value = "Raw_Data"

903 End Sub

```

8.5.3 Get_Hyperion_Data Module

```

1 Private run_stat As Integer
2 Private wb_name As String
3 Dim shtcur As Worksheet
4 Dim wkbkcur As Workbook

5 'Libraries required to close instances of Hyperion/Brio with strTerminateThis if query fails
6 Declare Function FindWindow Lib "user32" Alias "FindWindowA" ( _
ByVal lpClassName As String, _
ByVal lpWindowName As String _
) As Long

7 Declare Function FindWindowEx Lib "user32" Alias "FindWindowExA" ( _
ByVal hWndParent As Long, _
ByVal hWndChildAfter As Long, _
ByVal lpszClassName As String, _
ByVal lpszWindowName As String _
) As Long

8 Public Declare Function IsWindowVisible Lib "user32" _
(ByVal hwnd As Long) As Long

9 Function CreateScript() As String

10 Dim script As String
11 Dim fnum As Integer
12 Dim Filename As String
13 Dim Line As String
14 Dim PosNum As Integer
15 Dim LineLetter As String
16 Dim hyperion_path As String
17 Dim x As Integer
18 Dim y As Integer
19 Dim i As Integer
20 Dim EffCount As Integer

21 Line = shtcur.Cells(18, 3)
22 PosNum = shtcur.Cells(18, 5)
23 LineLetter = shtcur.Cells(18, 6)

24 Filename = "\" + Line + ".txt"

25 script = wkbkcur.Path + Filename
26 shtcur.Cells(9, 1) = wkbkcur.Path + "\Query.bgy"
27 hyperion_path = Replace(shtcur.Cells(9, 1), "\", "\\")

28 shtcur.Range("H:H").Clear

29 'Write script to cells
30 shtcur.Cells(1, 8) = shtcur.Cells(1, 9).Text
31 shtcur.Cells(2, 8) = shtcur.Cells(2, 9).Text
32 shtcur.Cells(3, 8) = shtcur.Cells(3, 9).Text
33 shtcur.Cells(4, 8) = shtcur.Cells(4, 9).Text
34 shtcur.Cells(5, 8) = shtcur.Cells(5, 9).Text
35 shtcur.Cells(6, 8) = shtcur.Cells(6, 9).Text
36 shtcur.Cells(7, 8) = shtcur.Cells(7, 9).Text
37 shtcur.Cells(8, 8) = shtcur.Cells(8, 9).Text
38 shtcur.Cells(9, 8) = shtcur.Cells(9, 9).Text
39 shtcur.Cells(10, 8) = shtcur.Cells(10, 9).Text
40 shtcur.Cells(11, 8) = shtcur.Cells(11, 9).Text
41 shtcur.Cells(12, 8) = shtcur.Cells(12, 9).Text

42 'Adds a limit to the query to include each positions

```



```

43     For x = 1 To PosNum
44         If x < 10 Then
45             shtcur.Cells(12 + x, 8).Value = "ActiveDocument.Sections[" & Chr(34) & "qDEFECT Details" &
Chr(34) & "].Limits[" & Chr(34) &
"Src Dept" & Chr(34) & "].SelectedValues.Add(" & Chr(34) & "P0" & x & LineLetter & Chr(34) & ")"
46         Else
47             shtcur.Cells(12 + x, 8).Value = "ActiveDocument.Sections[" & Chr(34) & "qDEFECT Details" &
Chr(34) & "].Limits[" & Chr(34) &
"Src Dept" & Chr(34) & "].SelectedValues.Add(" & Chr(34) & "P" & x & LineLetter & Chr(34) & ")"
48         End If
49     Next

50     'Adds a limit to the query for each requested effectivity
51     EffCount = 0
52     For i = 1 To 20
53         If shtcur.Cells(i + 17, 1).Value <> "" Then
54             Eff = ""
55             For y = 1 To 7 - Len(shtcur.Cells(i + 17, 1))
56                 Eff = Eff & "0"
57             Next
58             Eff = Chr(34) & Eff & shtcur.Cells(i + 17, 1).Text & Chr(34)
59             shtcur.Cells(PosNum + 13 + EffCount, 8).Value = "ActiveDocument.Sections[" & Chr(34) & "qDEFECT
Details" & Chr(34) &
"].Limits[" & Chr(34) & "Effectivity" & Chr(34) & "].SelectedValues.Add(" & Eff & ")"
60             EffCount = EffCount + 1
61         End If
62     Next i

63     x = EffCount + 13 + PosNum
64     y = 13

65     Do While shtcur.Cells(y, 9).Text <> ""
66         If Not shtcur.Cells(y, 9).Text Like "*qSupervisors P4S*" Then
67             shtcur.Cells(x, 8) = shtcur.Cells(y, 9).Text
68             x = x + 1
69         End If
70         y = y + 1
71     Loop

72     'Create script txt file
73     fnum = FreeFile

74     'If the text and xls files exist already delete them
75     If FileExists(script) Then
76         SetAttr script, vbNormal
77         Kill (script)
78     End If

79     If FileExists(wbkbcur.Path & "\" & Line & ".xls") Then
80         Kill (wbkbcur.Path & "\" & Line & ".xls")
81     End If

82     'Open text file and write cells to text file to create the script
83     Open script For Output As fnum
84     x = 1
85     Do While shtcur.Cells(x, 8).Value <> ""
86         Print #fnum, shtcur.Cells(x, 8).Text
87         x = x + 1
88     Loop

89     Close #fnum
90     SetAttr script, vbHidden

91     'hides the txt script file from view in the folder
92     CreateScript = script

93 End Function

94 Function FileExists(ByVal strFile As String, Optional bFindFolders As Boolean) As Boolean
95     'Purpose: Return True if the file exists, even if it is hidden.
96     'Arguments: strFile: File name to look for. Current directory searched if no path included.
97     'bFindFolders. If strFile is a folder, FileExists() returns False unless this argument is True.
98     'Note: Does not look inside subdirectories for the file.
99     'Author: Allen Browne. http://allenbrowne.com June, 2006.
100    Dim lngAttributes As Long

101    'Include read-only files, hidden files, system files.
102    lngAttributes = (vbReadOnly Or vbHidden Or vbSystem)

103    If bFindFolders Then
104        lngAttributes = (lngAttributes Or vbDirectory) 'Include folders as well.

```

```

105 Else
106     'Strip any trailing slash, so Dir does not look inside the folder.
107     Do While Right$(strFile, 1) = "\"
108         strFile = Left$(strFile, Len(strFile) - 1)
109     Loop
110 End If

111 'If Dir() returns something, the file exists.
112 On Error Resume Next
113 FileExists = (Len(Dir(strFile, lngAttributes)) > 0)
114 End Function

115 Sub process_file()

116     Set wkbkcur = ActiveWorkbook

117     'Control sheet is a hidden sheet which contains data for writing the script
118     Set shtcur = wkbkcur.Sheets("Control Sheet")

119     Call run_hyp

120     run_stat = 0
121     Call wait_for_hyperion

122 End Sub

123 Sub run_hyp()

124     Dim script As String
125     Dim CurFol As String

126     'Create Script Txt file to run Brio query
127     CurFol = wkbkcur.Path
128     script = CreateScript
129     briopath = CurFol & "\briogry.exe " & "-nosplash -jscript " & Chr(34) & script & Chr(34)

130     'Run Brio query
131     Shell briopath, vbNormalFocus

132 End Sub

133 Sub wait_for_hyperion()

134     Dim wbtemp As Workbook
135     Dim shttemp As Worksheet
136     Dim Line As String, Pos As Integer, Location As String, Pos_Letter As String
137     Dim dots As String
138     Dim IsErr As Boolean
139     Dim objWMICimv2 As Object
140     Dim objProcess As Object
141     Dim objList As Object

142     Set shttemp = ActiveWorkbook.Sheets("Control Sheet")

143     Line = shttemp.Cells(18, 3)
144     Pos = shttemp.Cells(18, 5)
145     Location = shttemp.Cells(18, 4)
146     Pos_Letter = shttemp.Cells(18, 6)

147     'Stop run if Hyperion is taking too long

148     If run_stat > 45 * 60 Then
149         MsgBox ("Hyperion has been running for 45 minutes and hasn't completed. Free up computation
resources and try again.")
150     ElseIf Dir(ActiveWorkbook.Path + "\" + Line + ".xls") = "" Then
151         Dim hyperion As Long, dialogwindows As Long
152         hyperion = FindWindow("hyperion", vbNullString)
153         dialogwindows = FindWindow("dialog windows", vbNullString)
154         If IsWindowVisible(dialogwindows) = 0 Then
155             Application.OnTime Now() + TimeValue("00:00:01"), "wait_for_hyperion"
156             run_stat = run_stat + 1

157             'Update the status box to indicate Hyperion is still running
158             If (run_stat + 3) Mod 4 = 0 Then
159                 dots = ""
160             ElseIf (run_stat + 3) Mod 4 = 1 Then
161                 dots = "."
162             ElseIf (run_stat + 3) Mod 4 = 2 Then
163                 dots = ".."
164             ElseIf (run_stat + 3) Mod 4 = 3 Then
165                 dots = "..."

```

```

166     End If
167     Status.WOComp.Value = "Running: Collecting Data from Hyperion" & dots
168     Status.Repaint
169     Application.DisplayAlerts = True
170 Else
171     'If Hyperion throws an error end show an error message and end the run

172     strTerminateThis = "brioqry.exe" 'Process to terminate,

173     Set objWMicmv2 = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\.\root\cimv2") 'Connect to CIMV2 Namespace

174     Set objList = objWMicmv2.ExecQuery _
("select * from win32_process where name='brioqry.exe'") 'Find the process to terminate

175     For Each objProcess In objList
176         objProcess.Terminate
177     Next

178     MsgBox "Hyperion has an error (possibly an incorrect password), Try again"
179     Application.ScreenUpdating = True
180     Application.WindowState = xlMaximized
181     End
182 End If
183 Else
184     strTerminateThis = "brioqry.exe" 'Process to terminate,

185     Set objWMicmv2 = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\.\root\cimv2") 'Connect to CIMV2 Namespace

186     Set objList = objWMicmv2.ExecQuery _
("select * from win32_process where name='brioqry.exe'") 'Find the process to terminate

187     Sleep 10000
188     For Each objProcess In objList
189         objProcess.Terminate
190     Next

191     'Go back to the Create_Tool Module
192     Call ContinuationAfterHyperion
193 End If

194 End Sub

```

8.5.4 Script Sample

```

Application.Documents.Open("C:\\ Desktop\\Heat Map Program\\Query.bqy")
ActiveDocument.Sections["qDEFECT Details"].DataModel.Connection.Open("C:\\Desktop\\ Heat Map
Program\\Database.oce")
ActiveDocument.Sections["qDEFECT Details"].DataModel.Connection.Connect()
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].Operator=bqLimitOperatorEqual
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.RemoveAll()
ActiveDocument.Sections["qDEFECT Details"].Limits["Model"].Operator=bqLimitOperatorEqual
ActiveDocument.Sections["qDEFECT Details"].Limits["Model"].SelectedValues.RemoveAll()
ActiveDocument.Sections["qDEFECT Details"].Limits["Position"].Operator=bqLimitOperatorEqual
ActiveDocument.Sections["qDEFECT Details"].Limits["Position"].SelectedValues.RemoveAll()
ActiveDocument.Sections["qDEFECT Details"].Limits["Model"].SelectedValues.Add("AA-12")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P01A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P02A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P03A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P04A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P05A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P06A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P07A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P07A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Src Dept"].SelectedValues.Add("P08A")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000100")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000101")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000102")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000103")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000104")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000105")

```



```

ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000106")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000107")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000108")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000109")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000110")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000111")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000112")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000113")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000114")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000115")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000116")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000117")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000118")
ActiveDocument.Sections["qDEFECT Details"].Limits["Effectivity"].SelectedValues.Add("0000119")
ActiveDocument.Sections["qDEFECT Details"].Process()
ActiveDocument.Sections["qRemove Dups"].Process()
ActiveDocument.Sections["qCombine Supv & Defect"].Process()
ActiveDocument.Sections["rCombine Supv & Defect"].Export("C:\\ Heat Map Program\\AA-
12.xls", bqExportFormatExcel2, true, false)
var myPath="C:\\Heat Map Program\\AA-12.txt"
var oleApp = new JOOLEObject("Scripting.FileSystemObject")
var traceDoc=oleApp.CreateTextFile(myPath)
traceDoc.Close()
Application.Quit()

```

8.5.5 Tool_Use Module

```

1  'The code called when cells are double clicked or buttons pressed.
2  Sub ClearFilterData()
3  'Clears filters from parts and modes
4  Application.ScreenUpdating = False
5  ActiveSheet.Unprotect
6  Rows("401:401").AutoFilter
7
8  'If the the time scrolling is enabled keep the effectivity filter
9  If TimeSeries.Visible And TimeSeries.ComboBox1 <> "" Then
10     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criterial:=ActiveSheet.Range("S2").Text
11     End If
12
13     ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Bold = False
14     ActiveSheet.Range("A1:R" & Range("A1:A40").Find("Total").Row).Font.Italic = False
15     Application.ScreenUpdating = True
16
17 End Sub
18
19 Sub FilterData(FilterString1, FilterString2)
20 'Filter for for part and mode items
21 Dim rngX As Range
22 Dim Filter1 As Integer
23 Dim Filter2 As Integer
24 Dim answer As String
25
26 Set rngX = ActiveSheet.Range("A$401:$BV$401").Find(FilterString1, Lookat:=xlPart)
27 If Not rngX Is Nothing Then
28     Filter1 = rngX.Column
29
30 End If
31 Set rngX = ActiveSheet.Range("A$401:$BV$401").Find(FilterString2, Lookat:=xlPart)
32 If Not rngX Is Nothing Then
33     Filter2 = rngX.Column
34
35 End If
36 Rows("401:401").AutoFilter
37
38 'If the time scrolling is enabled filter to the effectivity too
39 If TimeSeries.Visible And TimeSeries.ComboBox1 <> "" Then
40     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criterial:=TimeSeries.ComboBox1.Value
41     End If
42
43     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=Filter1, Criterial:="X"
44     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=Filter2, Criterial:="X"

```

```

36     RDorMap.Show
37 End Sub

38 Sub FilterData2(FilterString3)
39     'Filter for total rows or columns (parts or modes alone)
40     Dim rngX As Range
41     Dim Filter1 As Integer

42     Set rngX = ActiveSheet.Range("A$401:$BV$401").Find(FilterString3, Lookat:=xlPart)
43     If Not rngX Is Nothing Then
44         Filter1 = rngX.Column
45     End If
46     Rows("401:401").AutoFilter
47     'If the time scrolling is enabled filter to the effectivity too
48     If TimeSeries.Visible And TimeSeries.ComboBox1 <> "" Then
49         ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criteria1:=TimeSeries.ComboBox1.Value
50     End If
51     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=Filter1, Criteria1:=""X"

52 End Sub

53 Sub TimeScrolling()
54     'Used to reopen time scrolling form immediately after the pause button has been clicked
55     Application.DisplayAlerts = True
56     ActiveSheet.Unprotect
57     TimeSeries.Show (False)
58     ActiveSheet.Protect UserInterfaceOnly:=True, DrawingObjects:=True, Contents:=True, Scenarios:=True,
AllowSorting:=True, AllowFiltering:=True
59 End Sub

60 Sub ToACMap()
61     'Jump to the Aircraft Map
62     Range("AN1").Activate
63 End Sub

64 Sub ToHeatMap()
65     'Jump to the Heat Map
66     Range("A1").Activate
67 End Sub

68 Sub ToRawData()
69     'Jump to the Data
70     Range("A401").Activate
71 End Sub

```

8.5.6 TimeScrolling Form Module

FORM:

CODE:

```

1 Dim goscroll As Boolean
2 Dim NumUnique As Integer
3 Dim ScrollSpeed As Integer

```



```

4 Private Sub Back_Click()
5     Dim indexval As Integer

6     'When the user reaches the first effectivity it will then scroll around to the last
7     If TimeSeries.ComboBox1.Value = "" Then
8         indexval = UBound(TimeSeries.ComboBox1.List)
9     ElseIf TimeSeries.ComboBox1.ListIndex = 0 Then
10        indexval = UBound(TimeSeries.ComboBox1.List)
11    Else
12        If ActiveSheet.Range("S2").Text = TimeSeries.ComboBox1.Value Then
13            indexval = TimeSeries.ComboBox1.ListIndex - 1
14        Else
15            indexval = TimeSeries.ComboBox1.ListIndex
16        End If
17    End If

18    Application.Cursor = xlWait
19    TimeSeries.MousePointer = 11
20    TimeSeries.ComboBox1.Value = TimeSeries.ComboBox1.List(indexval)
21    'Save the chosen effectivity in a hidden cell on the sheet
22    ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value
23    'Filter to the chosen effectivity
24    ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criterial:=ActiveSheet.Range("S2")
25    Application.Cursor = xlDefault
26    TimeSeries.MousePointer = 0

27 End Sub

28 Private Sub Faster_Click()
29     'Decreases the pause between changes (only works if the faster button is enabled)
30     ScrollSpeed = ScrollSpeed - 1
31     'If the data is scrolling as fast as computationally possible, disable the faster button
32     If ScrollSpeed <= 0 Then
33         Faster.Enabled = False
34     Else
35         Faster.Enabled = True
36     End If
37 End Sub

38 Private Sub FiltertoEff_Click()
39     TimeSeries.MousePointer = 0
40     'Checks if the user has chosen an effectivity from the pull down menu
41     If TimeSeries.ComboBox1.Value = "" Then
42         MsgBox "First choose an effectivity"
43     Else
44         Application.Cursor = xlWait
45         TimeSeries.MousePointer = 11
46         'Save the chosen effectivity in a hidden cell on the sheet
47         ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value
48         'Filter to the chosen effectivity
49         ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criterial:=ActiveSheet.Range("S2")
50         Application.ScreenUpdating = True
51         Application.Cursor = xlDefault
52         TimeSeries.MousePointer = 0
53     End If
54 End Sub

55 Private Sub Forward_Click()
56     Dim indexval As Integer
57     Application.ScreenUpdating = False

58     DoEvents
59     'When the user reaches the last effectivity it will then scroll around to the first
60     If TimeSeries.ComboBox1.Value = "" Then
61         indexval = 0
62     ElseIf TimeSeries.ComboBox1.ListIndex = UBound(TimeSeries.ComboBox1.List) Then
63         indexval = 0
64     Else
65         If ActiveSheet.Range("S2").Text = TimeSeries.ComboBox1.Value Then
66             indexval = TimeSeries.ComboBox1.ListIndex + 1
67         Else
68             indexval = TimeSeries.ComboBox1.ListIndex
69         End If
70     End If

71     Application.Cursor = xlWait
72     TimeSeries.MousePointer = 11
73     TimeSeries.ComboBox1.Value = TimeSeries.ComboBox1.List(indexval)
74     'Save the chosen effectivity in a hidden cell on the sheet
75     ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value

```

```

76     'Filter to the chosen effectivity
77     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criteria1:=ActiveSheet.Range("S2")
78     Application.ScreenUpdating = True
79     Application.Cursor = xlDefault
80     TimeSeries.MousePointer = 0

81 End Sub

82 Function GetUnique() As Variant
83     'Gets all the effectivities nad puts them in a sorted list in a drop down menu for the effectivity
scrolling
84     ' Accepts an array or range as input
85     ' If Count = True or is missing, the function returns the number of unique elements
86     ' If Count = False, the function returns a variant array of unique elements
87     Dim lastrow As Integer
88     Dim EffArray() As Variant ' array that holds the unique items
89     Dim Element As Variant
90     Dim i As Integer
91     Dim FoundMatch As Boolean
92     Dim NumUnique As Integer
93     Dim sorted As Boolean

94     lastrow = WorksheetFunction.CountA(ActiveWorkbook.Sheets("Top Level").Range("A401:A40000")) + 400

95     'Counter for number of unique elements
96     NumUnique = 0

97     'Loop thru the input array
98     For Each Element In ActiveWorkbook.Sheets("Top Level").Range("$B$402:$B$" & lastrow)
99         FoundMatch = False

100        'Check if the item is in the array
101        For i = 1 To NumUnique
102            If Element = EffArray(i) Then
103                FoundMatch = True
104                GoTo AddItem '(Exit For-Next loop)
105            End If
106        Next i

107        AddItem:
108        'If the item not in list, add the item to unique list
109        If Not FoundMatch Then
110            NumUnique = NumUnique + 1
111            ReDim Preserve EffArray(NumUnique)
112            EffArray(NumUnique) = Element
113        End If

114    Next Element

115    NumUnique = NumUnique - 1

116    'Sort effectivities
117    sorted = False
118    Do While Not sorted
119        sorted = True
120        For i = 1 To UBound(EffArray) - 1
121            If EffArray(i) > EffArray(i + 1) Then
122                Temp = EffArray(i + 1)
123                EffArray(i + 1) = EffArray(i)
124                EffArray(i) = Temp
125            End If
126        Next i
127    Loop

128

129    For i = 1 To NumUnique
130    Next i
131    GetUnique = EffArray

132 End Function

133 Private Sub Newest_Click()
134     Application.ScreenUpdating = False
135     Application.Cursor = xlWait
136     TimeSeries.MousePointer = 11
137     TimeSeries.ComboBox1.Value = TimeSeries.ComboBox1.List(UBound(TimeSeries.ComboBox1.List))
138     'Save the chosen effectivity in a hidden cell on the sheet
139     ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value
140     'Filter to the chosen effectivity
141     ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criteria1:=ActiveSheet.Range("S2")

```

```

142 Application.ScreenUpdating = True
143 Application.Cursor = xlDefault
144 TimeSeries.MousePointer = 0
145 End Sub

146 Private Sub Oldest_Click()
147 Application.ScreenUpdating = False
148 Application.Cursor = xlWait
149 TimeSeries.MousePointer = 11
150 TimeSeries.ComboBox1.Value = TimeSeries.ComboBox1.List(0)
151 'Save the chosen effectivity in a hidden cell on the sheet
152 ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value
153 'Filter to the chosen effectivity
154 ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criteria1:=ActiveSheet.Range("S2")
155 Application.ScreenUpdating = True
156 Application.Cursor = xlDefault
157 TimeSeries.MousePointer = 0
158 End Sub

159 Sub Pause_Click()
160 Application.DisplayAlerts = False
161 goscroll = False

162 'Where the scroll speed is stored
163 Range("T2").Value = ScrollSpeed
164 TimeSeries.MousePointer = 0
165 Application.Cursor = xlDefault
166 'Sets the form location before it closes
167 ActiveSheet.Range("S1").Value = TimeSeries.Top
168 ActiveSheet.Range("T1").Value = TimeSeries.Left
169 'reopens the form when the play loop ends and causes it to exit
170 Application.OnTime Now + (TimeValue("00:00:01")) / 5, "TimeScrolling"

171 End
172 End Sub

173 Sub Play_Click()

174 goscroll = True
175 'If play is clicked scroll until pause is clicked.
176 Do
177 DoEvents
178 'When the scroll reaches the last effectivity it will then scroll around to the first
179 If TimeSeries.ComboBox1.Value = "" Then
180 indexval = 0
181 ElseIf TimeSeries.ComboBox1.ListIndex = UBound(TimeSeries.ComboBox1.List) Then
182 indexval = 0
183 Else
184 If ActiveSheet.Range("S2").Text = TimeSeries.ComboBox1.Value Then
185 indexval = TimeSeries.ComboBox1.ListIndex + 1
186 Else
187 indexval = TimeSeries.ComboBox1.ListIndex
188 End If
189 End If

190 Application.Cursor = xlWait
191 TimeSeries.MousePointer = 11
192 Application.ScreenUpdating = False
193 'Controls the speed of the scroll
194 Sleep ScrollSpeed * 250
195 TimeSeries.ComboBox1.Value = TimeSeries.ComboBox1.List(indexval)
196 'Save the chosen effectivity in a hidden cell on the sheet
197 ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value
198 'Filter to the chosen effectivity
199 ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2, Criteria1:=ActiveSheet.Range("S2")
200 Application.ScreenUpdating = True
201 Application.Cursor = xlDefault
202 TimeSeries.MousePointer = 0
203 'When pause is clicked goscroll=false and play loop exits but this also exits the form
204 If Not goscroll Then Exit Do
205 Loop
206 End Sub

207 Private Sub ShowAll_Click()
208 'Clears the choice on the pull down menu
209 TimeSeries.ComboBox1.Value = ""
210 'Clears the hidden cell on the sheet storing the effectivity
211 ActiveSheet.Range("S2") = TimeSeries.ComboBox1.Value
212 'Clears the effectivity filter
213 ActiveSheet.Range("$A$401:$BV$32000").AutoFilter Field:=2
214 End Sub

```



```

215 Private Sub Slower_Click()
216     'Increases the pause between changes
217     ScrollSpeed = ScrollSpeed + 1
218     'If the data is scrolling as fast as computationally possible disable the faster button
219     If ScrollSpeed <= 0 Then
220         Faster.Enabled = False
221     Else
222         Faster.Enabled = True
223     End If
224 End Sub

225 Sub UserForm_Initialize()

226     Dim i As Integer
227     Dim EffArray As Variant

228     'If the scroll speed is set to slower than the fastest possible enable the faster button
229     ScrollSpeed = Range("T2").Value
230     If ScrollSpeed > 0 Then Faster.Enabled = True

231     ''Set the location of the form
232     TimeSeries.Top = ActiveSheet.Range("S1").Value
233     TimeSeries.Left = ActiveSheet.Range("T1").Value

234     ''Fill the drop down menu with the available effectivities
235     EffArray = GetUnique
236     i = 1
237     For i = 1 To UBound(EffArray)
238         ComboBox1.AddItem EffArray(i)
239     Next i

240     TimeSeries.ComboBox1.Value = ActiveSheet.Range("S2")

241 End Sub

242 Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)

243     Dim okcanmsg As Integer
244     'When the user clicks the exit button
245     'If the data is filtered to aneffectivity warn the user that effectivity filters will be removed when
the form is closed
246     If ActiveSheet.Range("S2") <> "" Then
247         'Lets the user choose to keep the form opened or close it
248         okcanmsg = MsgBox("On form close effectivity filters will be removed.", vbOKCancel, "Close Form?")
249         'If they choose to close it
250         If okcanmsg = 1 Then
251             Application.Cursor = xlWait
252             TimeSeries.MousePointer = 11
253             Application.ScreenUpdating = False
254             'Ends automatic scrolling if it is enabled
255             If goscroll = True Then goscroll = False
256             Rows(401).AutoFilter Field:=2
257             'Clears hidden cells
258             TimeSeries.ComboBox1.Value = ""
259             ActiveSheet.Range("S2") = ""
260             ActiveSheet.Range("S1") = ""
261             ActiveSheet.Range("T1") = ""
262             Cancel = 0
263             Application.ScreenUpdating = True
264             Application.Cursor = xlDefault
265             TimeSeries.MousePointer = 0
266         End If
267         'If they choose to keep it open
268     Else
269         Cancel = 1
270     End If
271 End Sub

272 End Sub

```