



## Case Report

# Optimizing Test and Inspection Operations in Complex Engineering Products

Susan Ferkau, Quinn Risch, Prashanth Bhat, Kevin Lynch\*

Digital Technology, Raytheon Technologies, Waltham, USA

### Email address:

Susan.Ferkau.Neumann@rtx.com (S. Ferkau), Quinn@rtx.com (Q. Risch), Prashanth.J.Bhat@rtx.com (P. Bhat),

Kevin.J.Lynch@rtx.com (K. Lynch)

\*Corresponding author

### To cite this article:

Susan Ferkau, Quinn Risch, Prashanth Bhat, Kevin Lynch. Optimizing Test and Inspection Operations in Complex Engineering Products. *American Journal of Operations Management and Information Systems*. Vol. 6, No. 1, 2021, pp. 9-15. doi: 10.11648/j.ajomis.20210601.12

**Received:** March 16, 2021; **Accepted:** March 30, 2021; **Published:** April 12, 2021

---

**Abstract:** Delivery speed and product cost are critical to both our customers and our shareholders. Test cost has historically represented a third or more of overall product cost. Testing requires considerable time investments as well, especially given the nature of products in the aerospace domain, and their safety demands. In this paper we describe work in use today at a large aerospace manufacturer to optimize test and inspection operations in complex engineering products. We extend Deming's work from the theoretical to application by applying a decision tree and data analytics to test information, resulting in significant savings in dollars and time for test and inspection operations. A bill-of-materials plus operations visualization is employed to initially identify test and inspection operation candidates for removal, and then Deming's work is extended in this paper to determine the business case for removal, resulting in a final approval by experts driven by the underlying data. The decision tree is described, as well as algorithms to estimate failure rate and rework costs that are integral to applying Deming's analysis. A small set of business case results for removing an inspection and a test operation using the applied analysis are shared.

**Keywords:** Test Optimization, Bill-of-Materials Plus Operations, Cost Reduction

---

## 1. Introduction

Test cost represents between 30%-50% of product cost, as detailed in previous works [1-4]. A single test operation may be comprised of over 1000 test measurements. In 2017, we formed an analytics team focused on reducing engineering test cost [5]. We initially found the data quality problematic and challenging, but were able to develop methods for users to improve its quality for analytics use [6]. Maksi et al detail related work to leverage improved test data quality to identify test reduction possibilities [7]. This paper describes our work systematically applying knowledge to remove expensive test operations, while maintaining product integrity, extending the work of Deming [8]. Visualization of test operations is described in a previous study [9], and was used to provide context for the current work. The bill-of-materials plus operations view used as the basis for these visualizations is described by Jiao et al. [10], which employs NIST's (National

Institute of Standards and Technology) CMSD (Core Manufacturing Simulation Data) standard [11]. The d3.js tool suite is described by Bostock [12].

## 2. Approach

We examined the analysis proposed by Deming [8], chapter 15, for reducing incoming lot inspections, and extended his work to test operations within our factory. This was used to create a business case to determine if it made sense to remove a redundant test. The analysis suggested by Deming is simple; if the failure rate is less than the cost of the incoming inspection over the cost of addressing the failure at a later point in the production process then consider removing the incoming inspection. (See Figure 1.)

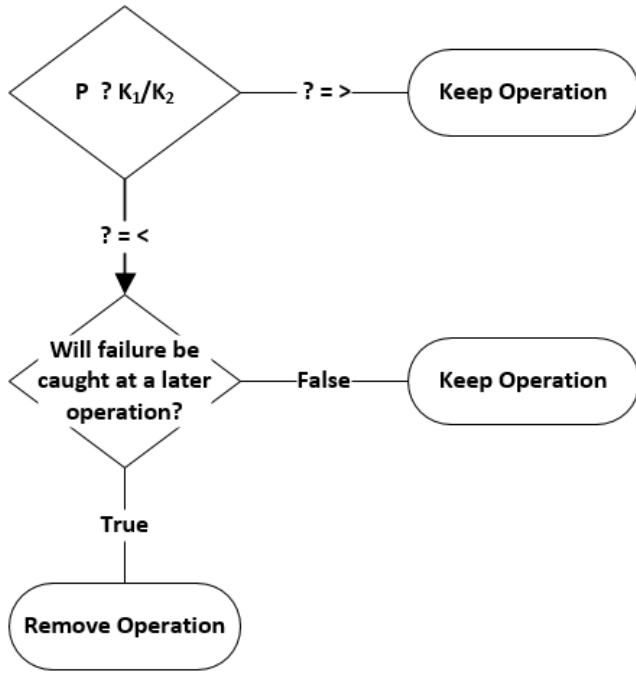


Figure 1. Deming Analysis.

Stated in another way, it costs more to complete a test on all units than it costs to repair failures later in the production process. Deming's analysis assumes omniscience and no error, which of course was not available to us.

From Deming [8], p. 41:

Case 1: If  $P < K_1/K_2$ , no inspection

Case 2: if  $P > K_1/K_2$ , 100% inspection

Essentially we have a straightforward question, is the failure rate,  $P$ , for a test less than the ratio of the cost of the test,  $K_1$ , over the total cost of capturing a defective unit at a later test,  $K_2$ ? Given the realities of the data we were working with, there were five major challenges with even the simple formula that Deming proposed (and these challenges encompass the entire formula):

1. How does one calculate the cost (either of doing an operation, or of performing rework)?
2. What is  $P$  (the true underlying failure rate)?
3. What is  $K_1$  exactly and what are  $K_1$ 's bounds?
4. What is  $K_2$  (the cost of rework associated with failures at the upper level test)?
5. What inspection or test operations are redundant?

$$P_{UB} = 1 - \max\{0, B\}$$

$$B = \frac{2n(1 - P_E) + z^2 - \left[ z \sqrt{z^2 - \frac{1}{n} + 4n(1 - P_E)P_E + (4(1 - P_E) - 2) + 1} \right]}{2(n + z^2)}$$

- (1) Where  $n$  is the number of trials,  $x$  is the number of passes,  $P_{UB}$  is the upper bounds for the failure rate,  $P_E$  is the empirically calculated failure rate  $((n-x)/n)$ ,  $z$  is the  $z$ -score from a Normal distribution that corresponds to the desired confidence (often we choose  $z = 2.97$ ), and  $\alpha$  is the error  $(1 - \text{confidence interval})$ .

In the aerospace and defense industries test failures are well documented. Even issues as a result of test station problems, operator error (e.g. did not plug in the unit), or factory blackouts would be recorded at the unit level, therefore we needed to tease out failures or issues that were actually traceable to hardware problems.

### 2.1. Cost

The most important question that all others predicate upon is whether cost can be mined and or calculated. If the cost data is unavailable or unknowable then all follow on questions are moot. In our case we have plenty of touch labor data (charged hours) available to us and traceable to units, but the hours charged by engineers, support, and maintenance were difficult to derive and trace to a per unit cost. Additionally we could not know the overhead costs associated with security, electricity, and other utilities. It was determined that the touch labor hours were a good proxy for cost and that all other costs were approximately linearly proportional, therefore costs (within the same product line) could be compared and well approximated using touch labor data.

### 2.2. Failure Rate

The failure rate,  $P$ , is calculated empirically as the total number of units that failed on the first test attempt and required rework, divided by the total number of distinct units tested. Where there were a relatively small number of units tested, and very few failures, one will have a large uncertainty as to the true failure rate. A failure rate of 0% will always be less than the ratio of test costs to rework costs, however one can never really be completely certain the true failure rate is actually 0. For instance, no failures in 100 units may not mean there would be no failures in 1000 units (or 10,000 units). In order to ensure we could be certain of any savings we used the upper bound of  $P$ , the failure rate (i.e. the failure rate was overestimated). We considered several different methods of calculating the upper bound of  $P$ , the failure rate, including the Clopper-Pearson interval, an upper confidence bound based on the Beta distribution, the Wilson Score and the Wilson Score with Continuity [13, 14]. The upper bounds calculated by these methods were very similar. We ultimately settled on the Wilson Score with Continuity because we could easily calculate the Wilson Score with continuity in a SQL query. See (1) below.

### 2.3. The Cost of Performing the Test

The cost of performing the lower level test,  $K_1$ , is more difficult to estimate. Our products have varying TAKT times ranging from minutes to tens of hours and therefore  $K_1$  is highly dependent upon the factory and the product specifics.

$K_1$  was estimated/derived by examining the touch labor data for the lower level test (i.e. the one we were attempting to remove). Given the various external factors affecting the actual charge labor time we were very careful about the timeframe in which we used to estimate  $K_1$ . We always consulted factory product engineers and management to ensure there were no process changes or design changes that straddled the timeframe that we used to estimate  $K_1$ .

$K_1$  needs to also account for costs associated with opportunity for failure not associated with hardware, meaning  $K_1$  needs to bundle in the cost of failures associated with test position problems, and operator error. Whatever failure rate that is not associated with hardware, as outlined in the previous section (2.2.), should be taken into account for the rolled up cost of  $K_1$ . The specifics of how to roll these costs in  $K_1$  will be unique to each factory and or product, however the costs for our products were reasonably estimated using the following (2):

$$K_1 \approx \bar{L}_p + \frac{\sum R_{NH}}{F_{NH}}$$

- (2) Where  $\bar{L}_p$  is the average labor cost of units that passed,  $\sum R_{NH}$  is the sum of all rework associated with failures that were not related to hardware (the unit), and  $F_{NH}$  is the number of units that failed non-hardware related issues.

#### 2.4. Rework Costs

$K_2$  is the total cost to capture a failure at a later point in production (see Figure 2). Many aerospace products include a complex assembly process. Testing is conducted at different points and at different stages of assembly. If a defective unit is not tested and the defect is found later in the process, the product must be disassembled, repaired, reassembled and retested. If the defect is found immediately after the deleted test the cost associated with  $K_2$  is minimal. If the failure cannot be detected until much later in the production process the cost associated with  $K_2$  may be very large. In a typical factory scenario there are a very small number of rework scenarios that span multiple assemblies since the failures are found at the lower level.

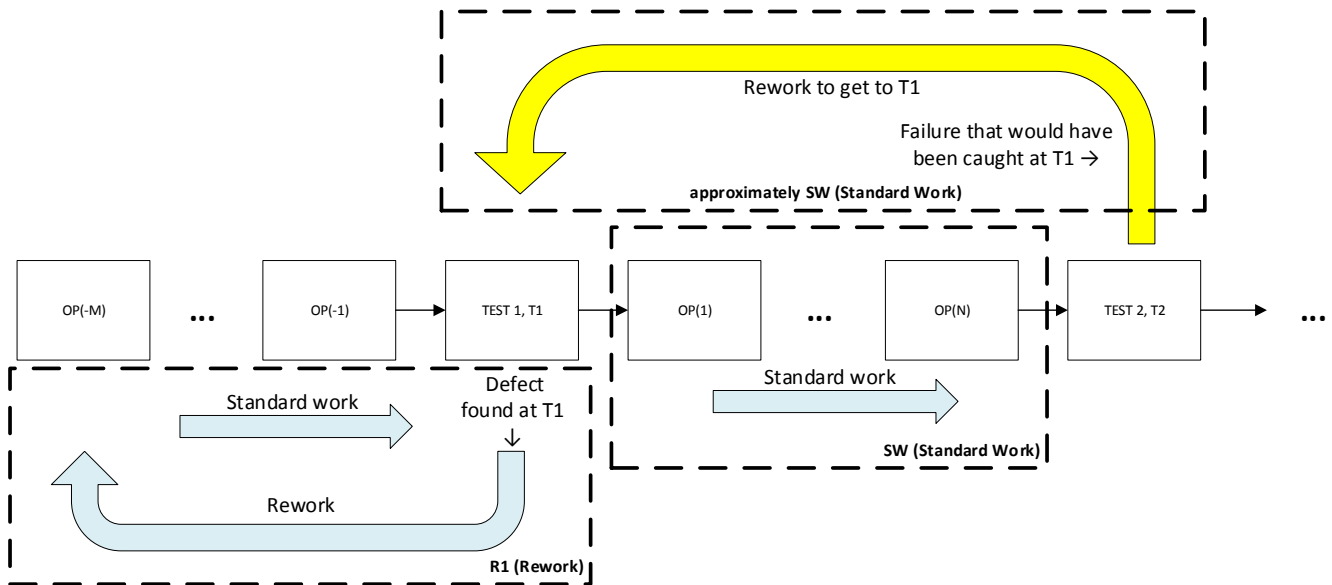


Figure 2. Rework Diagram for Estimation of  $K_2$ .

We developed an algorithm to estimate  $K_2$  based on all historic data (3).

$$K_2 \approx SW + R_1 + SW = 2SW + R_1$$

- (3) Where  $SW$  is the standard work content from the lower level test to the upper level test (where the failure is now found), and  $R_1$  is the historical average of touch labor hours needed go from the lower level test to the upper level test.

#### 2.5. Redundant Operation Identification

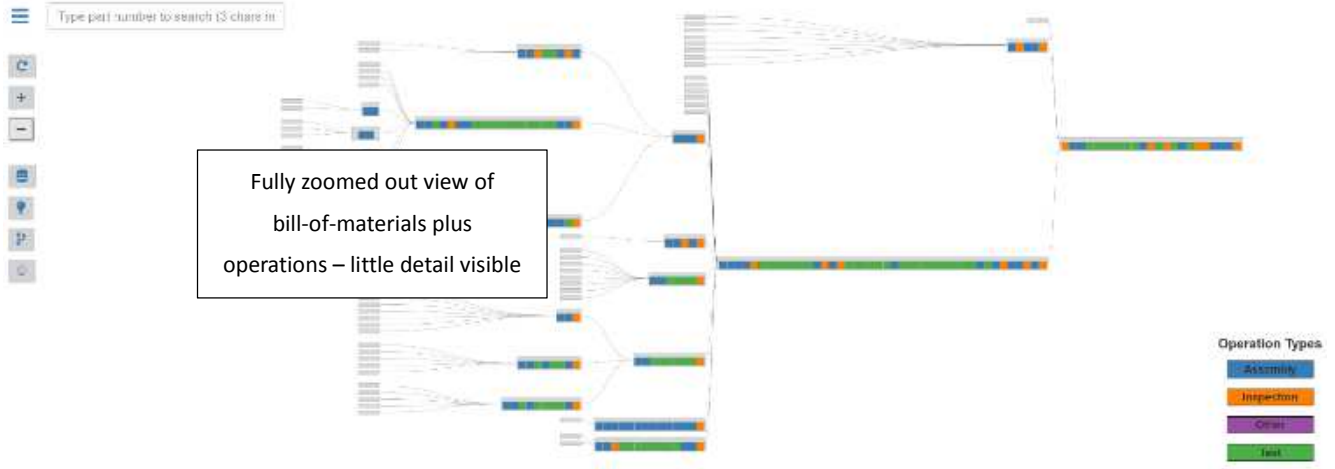
Initially we completed Deming analyses on test operations identified by product engineers. These engineers had intimate knowledge of the test process for specific assemblies and they

identified tests that were potentially redundant. We found instances where there was a sound business case to remove a test operation, and instances where the failure rate and cost of rework precluded removing a test. Our analysis was very successful, but inefficient. This method was constrained to conducting analyses on a single assembly. However our production process involves combining many complex assemblies into a final production unit. It was rare to find an engineer with intimate knowledge of test requirements for multiple assemblies, much less the entire production build. The Production Flow Visualization (PFV) the authors developed, described previously [9], and depicted in Figure 3, provided a complete picture of the entire test process for a production unit which was then leveraged to facilitate conversations about potentially redundant operations with

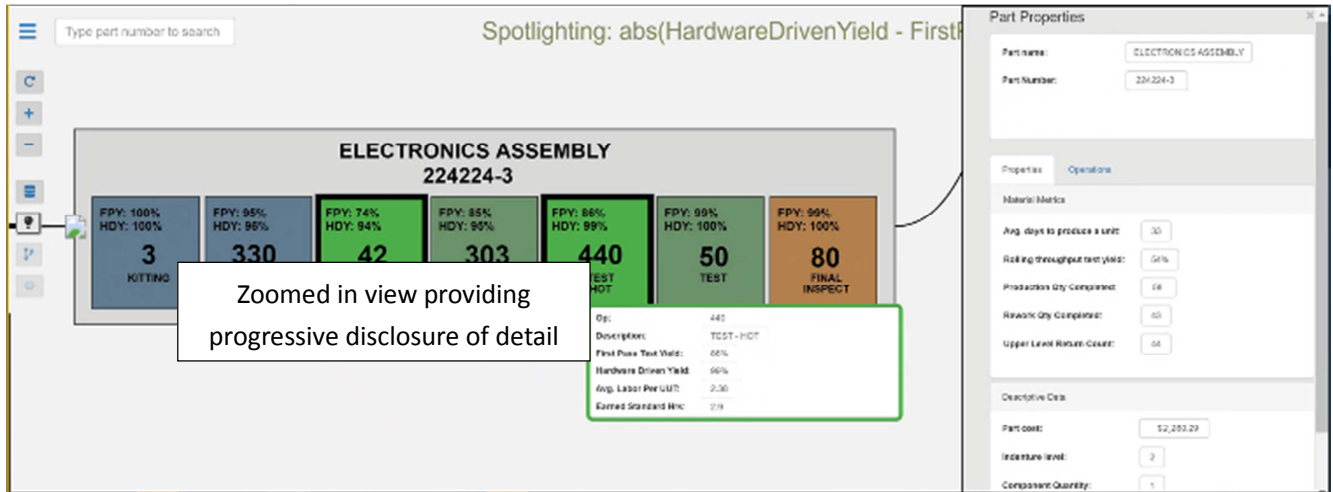
groups of experts.

To further understand the operating context this paper is describing, Figure 4 below is the zoomed in look at a portion of Figure 3, to a specific part and operations, with their

operation number (name), description, first pass test yield, hardware-driven yield, average labor, and earned standard hours information available.



**Figure 3.** Production Flow Visualization (bill-of-materials plus operations view), from [9].



**Figure 4.** Highlighting details in Production Flow Visualization from [9].

Production Flow Visualization clearly shows types of tests completed on sub-assemblies, the tests completed after the sub-assembly is integrated into an assembly, and finally when integrated into the production unit. In some instances a sub-assembly was exposed to a thermal environment or vibration test three (3) or more times. Using Production Flow Visualization (known alternately as bill-of-materials and operations [10]), we looked for instances where a specific type of test was repeated in the production chain. We created a list of tests that were potentially redundant (using our simple analysis). We then completed a Deming analysis and presented our results to a group of production experts.

## 2.6. The Whole Picture

The Deming Analysis for test reduction is based on a key assumption that removal of the lower level test does not increase/change the escape rate at the upper level test (either that the upper level test has 100% perception to lower level defects or

that the introduction of the lower level defects do not affect the escape rate from the upper level test). All final assessments as to whether a test should be removed was a decision made by product experts, and we relied heavily upon test and product engineering assessments of the likelihood of capturing failures at the later test. Additionally, the work that could have transpired in-between the lower level test and the upper level test is generically assumed to be reversible (e.g., no bonding operations) and therefore the forward flow work content is a good estimate of the reverse flow work content (e.g., torqueing and un-torqueing a bolt take roughly the same amount of labor). If the engineers were not confident that the defect would be captured, we did not recommend removing the initial test.

Below is the entire equation (4) that was used to assess the viability of test or inspection operation removal.

$$D = P - \frac{K_1}{K_2} = P_{UB} - \frac{\left(\bar{L}_P + \frac{\sum R_{NH}}{F_{NH}}\right)}{(2SW + R_1)}$$

When  $D > 0$  then Keep the Operation

When  $D < 0$  then proceed to review by a panel of experts to consider removing the operation

### 3. Results

We found three common outcomes to our analysis:

1. removing the test resulted in cost savings with an acceptable increase in risk of passing a bad unit
2. removing the test resulted in minimal cost savings
3. removing the test was not advisable because of high test failure rates, or high cost to address the failure later in the production process

Many of the analyses we completed spanned multiple sub-assemblies and assemblies. This meant that if a test was eliminated on a sub-assembly or component the defect would not be discovered until test at the assembly or production unit level. The total cost to capture a failure at a later point in production,  $K_2$ , would include disassembly of the assembly to

remove the component, and disassembly of the component to correct the defect. The failing component would have to complete all steps in the production flow until it was successfully reinstalled and tested in another assembly.  $K_2$  is defined as the total cost of rework for the defective component to be removed, repaired, retested and reassembled in another assembly. Production flow visualization provided insight into testing at each level of production assembly, and visualized the number of assembly and test operations performed between the test to be removed and the test that would capture any defects.

The Deming algorithm and the production flow visualization allowed us to look for test reduction opportunities that we could present to engineers for final review and approval. We use data driven estimates for all Deming components. We developed SQL based queries to automate the process of calculating  $P$ ,  $K_1$ , and  $K_2$ . We then created a decision process to determine which operations to analyze. The result was a decision tree, depicted below in Figure 5.

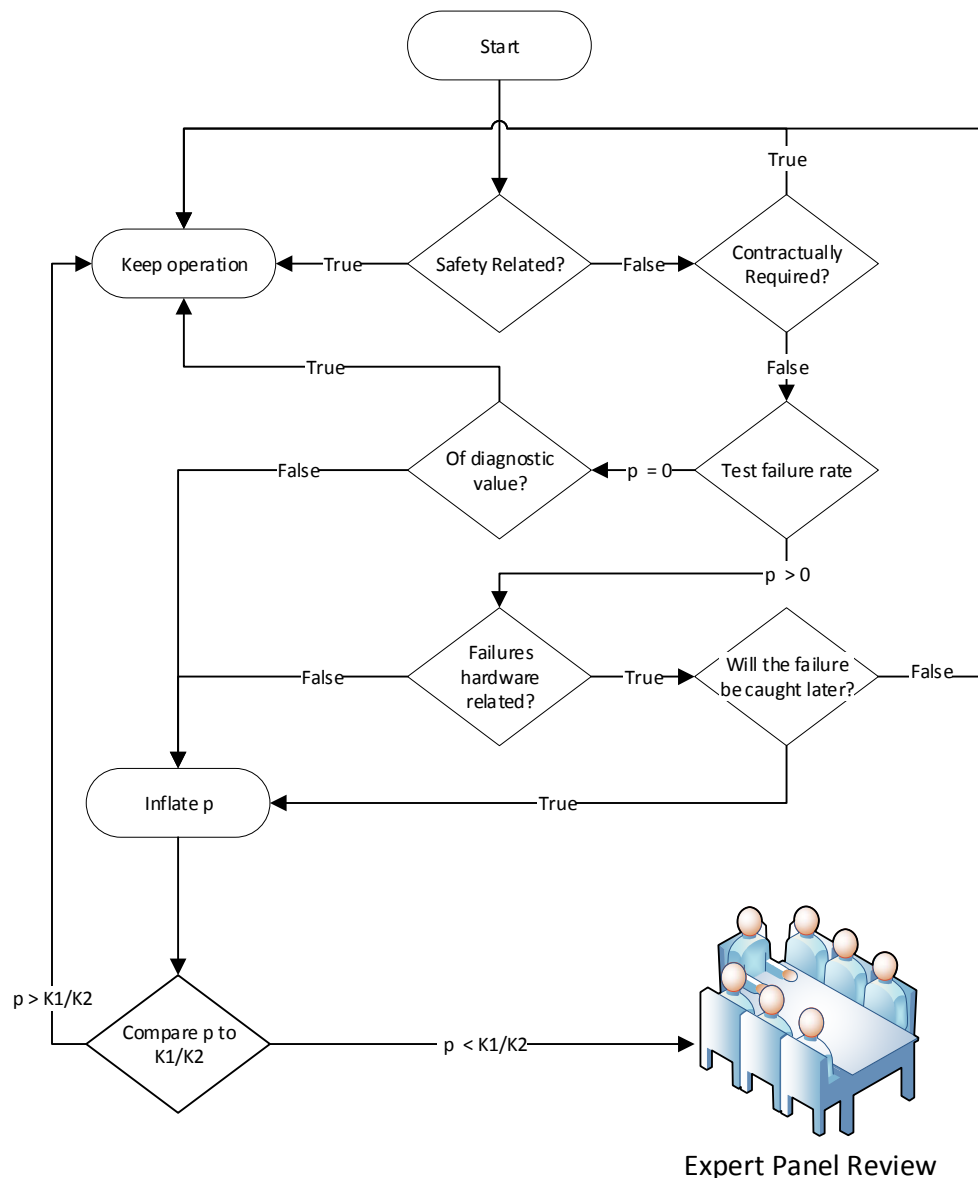


Figure 5. Deming Decision Tree.

The top of the decision tree breaks tests into diagnostic and non-diagnostic or acceptance tests. A diagnostic test is performed early in development to verify the product meets design requirements, or if an unexpected failure becomes prevalent. The test is often intended to be temporary, but becomes difficult to remove. If a test is determined to be diagnostic, and it never fails, the test should be removed. The test is not providing any useful information and may be stressing a component unnecessarily. Likewise, if a diagnostic tests fails very rarely it may not be providing useful information to a test engineer. A Deming analysis helps determine if there is a business case to remove the test. If the process required to remove the test is more expensive than the total savings incurred from removing the test then the test is not removed.

The rest of the decision tree guides a data analyst on when to apply Deming and when not to apply Deming. In many cases  $P$ , the failure rate, would have to be many times lower before a test could be removed. There are many nodes that require engineering input and cannot be assessed with data only. For instance, if the test is contractually required or safety related, or if the failure will be caught at a later operation. An engineering assessment is required to select

the correct branch of the tree. The decision tree in Figure 5 helped us to focus our efforts on test operations that could most likely be removed.

While our initial focus was on test operations, we later found that the Deming analysis also could be applied to inspection operations. There were instances in our factory where an inspection was added to the middle of a component assembly process in response to production issues. The production issue was caught late in the production cycle and triggered expensive rework. Often the production process was improved and the production issues were eliminated, but the new inspection was not. Inspections at our factory are performed by quality engineers. A unit may be held up for minutes to hours waiting for an available quality engineer to complete the inspection. The Deming analysis provided data driven evidence that removing the additional inspection would not increase the cost of production, consistent also with Reinertsen [15].

In Table 1, results for a subset of operations initially identified using production flow visualization described above are included with their actual hours, operation type, failure rate, Deming analysis, and touch labor hours saved per unit.

*Table 1. Business case based on reduced touch labor hours.*

Material	Remove Operation	Type	Capture Failure Operation	$K_1$ (Hrs)	$K_2$ (Hrs)	Failure Rate, Upper Bound	$K_1/K_2$	Deming $P < K_1/K_2$	Touch Labor Hrs Saved per unit
1	150	Inspection	180	0.4	1.30	0.03	0.31	YES	165.6
6	130	Inspection	200	0.4	3.00	0.14	0.13	NO	-
7	70	Test	80	1.2	10.60	0.00	0.11	YES	90.4

## 4. Conclusion

Modern factories are typically awash in data, but availability and quantity of data does not necessarily mean one has the exact data or metric required for analysis. The work herein shows a rigorous application of Edward Deming's theory from [8] and real world results (Table 1). Employing the Production Flow Visualization [9], insight was gained into testing at each level of production assembly to identify potentially redundant test and inspection operations. A decision tree was used to determine which operations should have the Deming analysis applied. The authors' extension of the Deming lot inspection algorithm [8] determined whether a business case existed to remove specific test and inspection operations. Both tests and inspections were successfully removed using the approach detailed in this paper, resulting in lower product cost, and greater factory throughput.

## Acknowledgements

Our colleague Gloria Hubner is remembered for her contributions, her spirit, and her courage.

## References

- [1] Turino, J. (2012). Design to test: a definitive guide for electronic design, manufacture, and service. Springer Science & Business Media.
- [2] Lam, H. (2004, July). New design-to-test software strategies accelerate time-to-market. In *Electronics Manufacturing Technology Symposium, 2004. IEEE/CPMT/SEMI 29th International* (pp. 140-143). IEEE.
- [3] Fox, B., Boito, M., Graser, J. C., & Younossi, O. (2004). Test and evaluation trends and costs for aircraft and guided weapons (No. RAND/MG-109-AF). Rand Corp., Santa Monica, CA.
- [4] Ziomek, C. D., & Jenkins, D. J. (2005, September). The need for embedded intelligence in ATE. In *IEEE Autotestcon, 2005.* (pp. 398-401). IEEE.
- [5] Berryman, S., Brio, A., Burkhardt, A., Ferkau, S., Gharbiah, H., Hubner, G., Lynch, K., & Woudenberg, M. (2017, September). Concept of operations for test cost analytics in complex manufacturing environments. In *2017 IEEE AUTOTESTCON* (pp. 1-8). IEEE.
- [6] Burkhardt, A., Berryman, S., Brio, A., Ferkau, S., Hubner, G., Lynch, K., Mittman, S., & Sonderer, K. (2018, September). Measuring Manufacturing Test Data Analysis Quality. In *2018 IEEE AUTOTESTCON* (pp. 1-6). IEEE.

- [7] Maksi, L., Berryman, S., Brio, A., Burkhardt, A., Elder, S., Ferkau, S., Gharbiah, H., Lynch, K., & Risch, Q., Deriving Common Factory Test Platform Requirements Using Historical Test Data,” in the *Journal of Information Technology and Software Engineering*, October 31, 2018.
- [8] Deming, W. E. (2018). *Out of the Crisis*. MIT press, 411.
- [9] Bhat, P., Berryman, S., Burkhardt, A., Cho, M., Ferkau, S., Gharbiah, H., Johnston, K., Lynch, K., Mittman, S., Risch, Q., & Swansen, M. (2020, December). Manufacturing Bill-of-Materials Plus Operations Visualization Using D3, in the *Journal of Information Technology and Software Engineering*, Vol. 11, Issue 1.
- [10] Jiao, J., Tseng, M. M., Ma, Q., & Zou, Y. (2000). Generic bill-of-materials-and-operations for high-variety production management. *Concurrent Engineering*, 8 (4), 297-321.
- [11] Lee, Y. T. T. (2015). A Journey in standard development: the core manufacturing simulation data (CMSD) information model. *Journal of research of the National Institute of Standards and Technology*, 120, 270.
- [12] Bostock, M., Ogievetsky, V., & Heer, J. (2011). D<sup>3</sup> data-driven documents. *IEEE transactions on visualization and computer graphics*, 17 (12), 2301-2309.
- [13] Clopper, C. J., & Pearson, E. S. (1934). The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26 (4), 404-413.
- [14] Newcombe, R. G. (1998). Two - sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in medicine*, 17 (8), 857-872.
- [15] Reinertsten, D. G. (2009). *The principles of product development flow: second generation lean product development*. Celeritas.