Carryover Parts and New Product Reliability

by

Gokhan Dogan

B.S., Industrial Engineering
Bogazici University, 2002

Submitted to the Sloan School of Management in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Management

at the

Massachusetts Institute of Technology

June 2009

Signature of Author ....
Department of Management
May 28, 2009

Certified by..
Nelson P. Repenning
Associate Professor of Management
Thesis Supervisor

Accepted by..
Ezra W. Zuckerman Sivan
Nanyang Technological University Associate Professor
Chair, Ph.D. Committee

Carryover Parts and New Product Reliability

by

Gokhan Dogan

Submitted to the Sloan School of Management
on May 28, 2009 in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy
in Management

ABSTRACT

By studying a unique data set from a motor vehicle manufacturer, we find that carryover parts, common parts used in successive generations of multi-generational products, are a major source of quality problems, contrary to conventional wisdom. Moreover, the failure rate of carryover parts grows from one generation to the next, a phenomenon known as the carryover spike. Motivated by these results and the need to understand the quality dynamics of multi-generational products, we empirically analyze the field problem-solving process and the new product introduction spike. We attempt to answer the following questions: what factors influence the time required to solve problems? Furthermore, what factors influence the cancellation probability of problem-solving projects? In addition to these questions related to the field problem-solving process, we seek to understand the factors that influence the new product introduction spike. We also investigate various ways to offset the failures of carryover parts. Using a novel simulation model, we test different policies that aim for better prioritization and analysis of carryover problems. Simulation results show that product reliability can be improved drastically using these policies. Our results indicate that managers should expect to witness higher warranty costs related to carryover parts on new products, due to trends in the industry.

Thesis Supervisor: Nelson P. Repenning
Title: Associate Professor of Management

Acknowledgements

Going through the doctoral program at MIT's System Dynamics Group was a fantastic journey. I felt privileged to be in an environment in which we were given the opportunity, time and resources to explore, find, and pursue whatever interests us and, most importantly, to improve ourselves. Every step of the process from finding a thesis topic to writing it, including the several iterations, contributed to my intellectual development. For that, I am most grateful to my thesis advisor, Nelson Repenning. He helped me find a great field site for my thesis project and introduced me to wonderful people there. His support throughout the process, his interest in the phenomenon I was studying, and his confidence in me was extremely valuable. His expertise in product quality and intuition were great resources whenever I had questions. Moreover, he patiently encouraged me to identify the real problem and made sure I focused on it. John Sterman has been a great mentor by setting high standards that always helped me to increase the quality of my work. More importantly, he has been an excellent role model whose principles will guide me whenever I face a professional or personal dilemma. Despite being thousands of miles away, Roger Bohn has been a terrific committee member. He spent several weekends reading my drafts and always challenged me by asking important questions that I never considered.

CS, my collaborator at the field site, was the absolute best collaborator I could have had. My collaboration with someone of his caliber was a blessing. I always admired his commitment to his organization. Writing this thesis would not have been possible without his knowledge in quality and reliability processes and his expertise in data analysis. His excellent technical skills were extremely valuable in helping me build my simulation model. His credibility within the organization allowed me to interview several people. Without their input, I would not have been able to discover the phenomenon I studied.

I am grateful to Yaman Barlas for introducing me to System Dynamics, supporting me for pursuing a PhD, and being the ultimate mentor whenever I needed advice about major decisions in my life.

Throughout the years, I enjoyed the friendship and support of several people. Thanks to Hazhir Rahmandad, Jeroen Struben, Paulo Goncalves, Brad Morrison, Mila Getmansky, Tim Quinn, Dan McCarthy, John Lyneis, Joe Hsueh, Kawika Pierson, Hasan Arslan, Anshuman Tripathy, Felipe Caro, Opher Baron, and Hongmin Li in the OM/SD group for great lunch, dinner, coffee break, and table function conversations. Burak Buke was always there whenever I had a technical question. Onur Kuzucu, Ekrem Esmen, Tim Quinn, Oncel Tuzel and Yildiray Yildiz were great roommates. We shared our passion for Besiktas with Volkan Muslu, Namik Kemal Yilmaz, Alp Simsek, and Ufuk Akcigit, sometimes even by waking up at 7am on the weekends. Thanks to Ufuk for going through all phases of the doctoral program with me, even though we were in different departments. Sharon Cayley was great at giving advice.

Many thanks to my family for their endless love and support. They made many sacrifices for my education and followed me to different cities. My father was the source of my passion for analytical thinking and was my best teacher. My mother surrounded me with her love, from which I take my inspiration. Reyhan has been more than a sister, despite the distance separating

us. I would also like to thank my extended family and Ozge's family for their constant encouragement.

Finally, I dedicate this thesis to my wife Ozge. She supported me in all phases of the doctoral program with her love. We spent so much time discussing the papers that she also deserves the doctoral degree.

To Özge

Table of Contents

Essay 1: Impacts of Carryover Parts on New Product Reliability

Abstract

Conventional wisdom suggests that carryover parts, common parts used in successive generations of multi-generational products, have positive impacts on the reliability and durability of products because "reliability and durability of [carryover] parts have been substantially tested in the market already" (Clark and Fujimoto, 1991). However, this claim has never been tested empirically. In this paper, we make two contributions to the literature. First, by studying a unique data set from a motor vehicle manufacturer, we find that carryover parts are a major source of quality problems, contrary to conventional wisdom, as approximately half of the warranty cost is due to carryover parts. Moreover, the failure rate of carryover parts grows from one generation to the next. Second, we study ways to offset the failures of carryover parts. Using a novel simulation model, we test different policies that aim for better prioritization and analysis of carryover problems. Simulation results show that product reliability can be improved drastically using these policies. We highlight organizational and mechanical barriers to the adoption of these policies. Our results also indicate that managers should expect to witness higher warranty costs related to carryover parts on new products due to the trends in the industry.

1. Introduction

Carryover parts, common parts used in successive generations of multi-generational products, are used substantially in a variety of product types. In a study that spans five industries ranging from electro-mechanical durables to consumer packaged goods, Griffin (1997) found that the average "newness" of 343 new product development projects was 56.6%, corresponding to an average carryover level of 43.4%. According to Muffatto (1996), the desired carryover degree was 50% for Honda Accord's 1993 model.

Usage of carryover parts is hypothesized to have several positive effects on company and product performance. They help companies to introduce their products to the market faster by reducing product development cycle times (Smith and Reinertsen, 1992). Carryover parts increase product variety by increasing research and development capacity (Robertson and Ulrich, 1998). Manufacturing and R&D costs are reduced by spreading fixed costs over several products, and logistics costs are reduced by reducing inventory (Collier, 1982) and delaying product differentiation (Lee, 1996).

In addition to these benefits, usage of carryover parts affects the quality and reliability of products. Clark and Fujimoto (1991) claimed that even though carryover parts might jeopardize product design by putting more constraints on design engineers and leading to suboptimal parts design, carryover parts have positive impacts on the reliability and durability of products because "reliability and durability of [carryover] parts have been substantially tested in the market already, reducing the risk of customer dissatisfaction due to design or manufacturing defects" (Clark and Fujimoto, 1991, pp. 147). Hence, they asserted that the usage of carryover parts is expected to improve the reliability of the product. However, despite the substantial body of literature on other effects of parts commonality, this claim has never been tested empirically and is thus the focus of this paper.

Conventional wisdom suggests that carryover parts have positive impacts on the reliability of products because they provide the opportunity to identify and solve problems associated with them before the introduction of the next generation products. Carryover problems are identified by using field failure information of the current-generation product. This information comes from customers and dealers in the form of warranty claims, surveys, and the like. If a problem identified in the current-generation product is associated with carryover parts, solving this problem will benefit next-generation products, too. Hence, solving carryover problems before the introduction of the next generation will have a major positive impact on next-generation products.

However, field problem solving is not the only process that improves the reliability of next-generation products. The new product development problem solving process also improves the reliability of next-generation products, and these two processes are undertaken simultaneously until the production of the next generation begins. Yet, despite the positive impact of field problem solving on new products' reliability, the problem solving literature on new products mostly focuses on the new product development problem solving process (Brown and Eisenhardt, 1995). The literature is silent about the role of field problem solving on new products' reliability.

Due to differences between the field problem solving process and the new product development problem solving process, they both require investigation but field problem solving, as previously stated, has mostly been neglected in the literature. The main differences between the processes are the information sources and the problem solving costs. Field failure information comes from several machines used by customers in the real product operating environment. On the other hand, the new product development problem solving process uses data from a handful of test machines usually tested in a lab environment. Therefore, field failure information is more representative and provides a better estimate of the significance of problems. However, a problem identified in the upfront stages of a new product development project yields great benefits if solved early because it will be much easier and cheaper to fix at this point (Thomke, 2003). These differences increase the need for a better understanding of the field problem solving process and carryover problems, which have not received sufficient attention in the literature.

This paper contributes to the literature on product development by empirically investigating the impact of carryover parts on product reliability. We find that carryover parts are a major source of quality problems, contrary to conventional wisdom. Moreover, the failure rate of carryover parts grows from one generation to the next. Second, we reveal the importance of field problem solving in improving the reliability of new products, and highlight organizational and mechanical challenges related to the field problem solving process. Third, we study ways to offset the failures of carryover parts. Using a novel simulation that models each problem and failure as a separate agent, we test different policies that aim for better prioritization and analysis of carryover problems. Simulation results show that product reliability can be improved drastically using our policies. Finally, we show that due to shrinking product development lead times managers should expect to witness a higher influence of carryover parts on new product reliability.

The following sections will present the empirical analysis of the field problem solving process and the impact of carryover parts on product reliability, describe the simulation model, and test different policies for managing the process of improving the reliability of carryover parts.

2. Empirical Analysis

2.1 Reliability Dynamics of Multi-Generation Products

In general, product reliability follows these dynamics: field problem solving teams improve the reliability of a product as field failure information accumulates. Then, reliability usually worsens when the next-generation product is introduced, as changes to the product and the production process are made. Figure 1 shows reliability data from successive generations of a representative product in our sample. The y-axis represents defects per machine, a measure of unreliability, with respect to production time. A machine is a single unit produced by the factory. A defect is a part or group of parts in a single machine that will eventually result in a failure, but hasn't resulted in a failure yet. A defect results in a failure after the machine is used by the customer for some time. Conversely, a problem is a flaw in the manufacturing process or design of a group of parts that leads to failures on multiple machines. Defects per machine data in Figure 1 is the ratio of total defects in mature[1] machines produced in a given production month to the number of mature machines produced that month. Our novel dataset includes the total operating hours of each machine to date, production date of each machine, and records of all failures on those machines including failure date. This data allows us to construct Figure 1 by identifying mature machines, failures on mature machines and their production date. Figure 1 shows a gradual improvement in the reliability of the first-generation product. The second-generation product starts with worse reliability compared to the final production months of the first-generation product. This phenomenon is quite typical and is known as the "new product introduction spike." After the introduction of the second-generation product, the field problem solving process improves the reliability of the second-generation product.

[1] Machines that are past 10,000 hours of operation are called mature machines. More than 80% of reported failures occur before 10,000 hours of operation. After 10,000 hours failures are seldom reported. Hence, data is sparse. In Figure 1, only mature machines are represented, as recently produced machines would have less than 10,000 hours of operation and, therefore, would have fewer reported failures. This would bias defects per machine results downward for recent months.

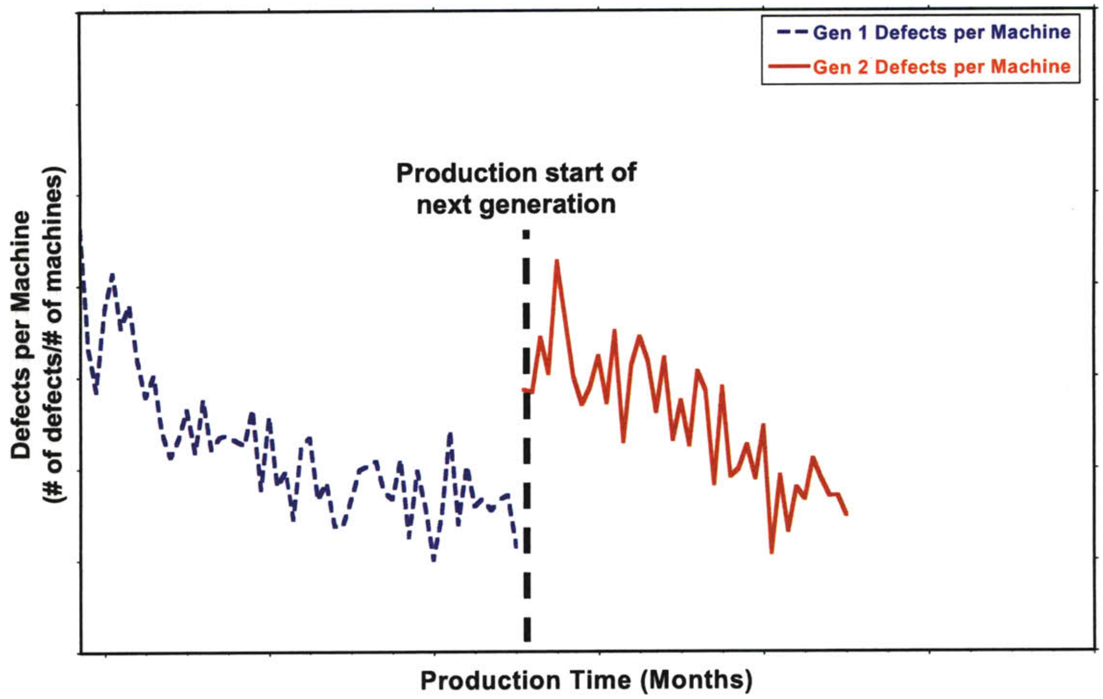


Figure 1: Defects per machine versus production date of products

Field failure information is useful for improving the reliability of both the current-generation product and the next-generation product. Field problem solving improves next-generation reliability by eliminating problems that will be carried over to the next generation. However, field failure information lags behind the production date of defective products. When a machine that will have a failure is produced, engineers are not aware of the situation. The defect is introduced during production, but the failure will not occur until the machine is being used. Engineers learn about the defect only after the machine fails while in use by the customer. Engineers use field failure information to start problem solving projects for problems that are worth being solved. Time delays between the production of a defective product and the failure date can be seen by comparing Figures 1 and 2. Figure 1 shows defects per machine vs. production date, whereas Figure 2 shows failures caused by the defects shown in Figure 1.

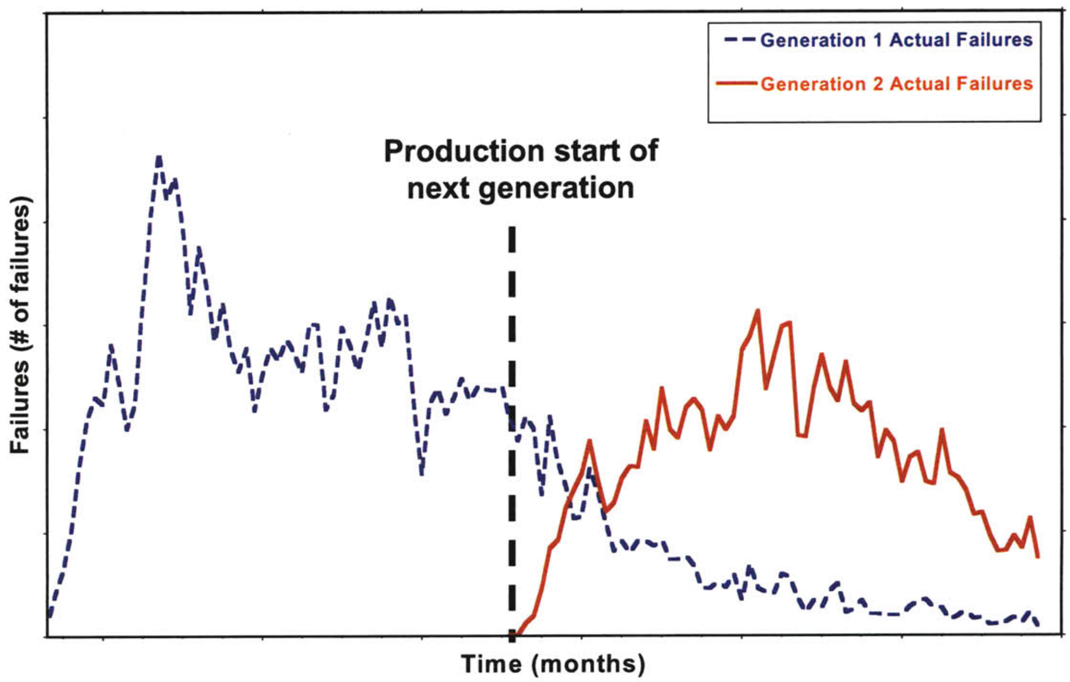


Figure 2: Failures reported to quality and reliability teams versus time

The first delay involved in the process is the sales and shipping delay. Once the machine has been shipped, the customer begins using the machine, and the defect results in failure while the product is being used by the customer. Failure information reaches quality and reliability teams after a reporting delay. Once quality and reliability teams learn about these failures, failure information is analyzed, and problems are identified. Identified problems are investigated to determine whether or not they are worth solving. Problems that are worth solving wait in a queue until engineers are assigned to them, and then they get solved. Most of the time, a solution to the problem will be implemented on machines that will be produced after the solution to the problem is found. Machines produced before the solution to the problem is found are recalled and fixed very rarely and only for very serious problems, since this is a very expensive option.

2.2 Data Analysis

The field problem solving process, new product introductions, and time delays involved in these processes are some of the factors that determine the impact of carryover problems on the reliability of new products. Therefore, we analyzed these three factors. First, we tried to answer the question, "What percent of a new product's warranty is due to carryover parts?" This metric shows us the impact of carryover parts on new product reliability. Conventional wisdom does not expect major reliability issues associated with carryover parts, since they have already been tested in the market and should have been fixed. We also analyzed the time delays in learning about field problems and fixing them. Another question we attempted to answer was, "Do carryover parts contribute to the new product introduction spike?" According to engineers and managers within our case site, the new product introduction spike was only associated with new parts because the design of carryover parts does not change from generation to generation, and there is no obvious reason for their reliability to deteriorate as production of next-generation products begins.

The data we used to answer these questions were collected over a period of three years at a major motor vehicle manufacturer. The manufacturer is a multi-billion dollar company that manufactures a wide range of products at several facilities around the world.[2] Thousands of engineers work on product development projects and solving product problems. They complete dozens of new product development projects every year. Throughout the production life cycle of an average product, thousands of machines are produced. An average product is comprised of tens of thousands of parts. We collected both qualitative and quantitative data from the organization. The qualitative data includes 68 formal, semi-structured interviews with 58 employees ranging from junior engineers to the VP of R&D. We recorded and transcribed all of these formal interviews. Even though we had a generic topic and a list of interview questions, these questions were open-ended, and we changed the focus if the interviewee raised an interesting point. In addition to these formal interviews, we conducted dozens of informal interviews that were not recorded. We also created a unique data set encompassing the following information from 125

[2]To preserve the confidentiality of the company, exact figures are not presented.

products: data on the machine population, parts on machines, new product introductions and the field problem solving process. Machine population data includes production, sales and shipping dates of machines, operating hours per day for each machine, and data on all failures on each machine such as failure date and warranty cost. Parts data include all failures reported on each part and whether the part is carried over or not. Databases have data on field problem solving projects such as important dates and failures related to problems. We also have access to new product introduction data.

2.3 Impact of Carryover Problems on Next-Generation Reliability

We analyzed data from 125 new product introductions to understand the impact of carryover parts on next-generation product reliability. The results in Table 1 below show that about half of the company's warranty costs and failures are due to carryover parts; on average, carryover parts lead to 43% of failures and 47% of the warranty costs. These numbers translate to substantial costs for motor vehicle manufacturers; in 2006, on average, warranty costs of U.S.-based automotive manufacturers comprised 1.68% of their revenues (Warranty Week, 2007). The fact that carryover parts lead to about half of all failures and half of all warranty costs is a counterintuitive result and indicates that carryover parts are a major source of reliability problems. This was contrary to conventional wisdom and the mental models of our research partner. Even though carryover problems were identified in several machines sold to customers, and even though each generation of products are in production for years, leaving engineers plenty of time to fix these problems, they still have a substantial impact on unreliability. Understanding this puzzle is necessary to improve our knowledge of new product development and reliability.

Failures on Carryover Parts/All Failures		
N	Mean	St. Dev.
125	0.43	0.16

Table 1: Summary statistics of the fraction of failures due to carryover parts.

2.4 Time Delays for Identifying and Solving Field Problems

Since carryover problems are identified by field failure information, we analyzed field failure information to understand why approximately half of the warranty costs were due to carryover parts. As previously mentioned, time delays occur before quality and reliability teams solve field problems. Time delays include the time between the start of production and the production of defective products, sales and shipping delays, reporting delays, and problem analysis and problem solving delays. As reported failures of a problem accumulate, engineers create a problem solving project. When a team becomes available to work on the problem, the project becomes activated and is closed when the problem is solved. See Figure 3 below for a breakdown of the delays. These delays reduce the benefits of problem solving. To understand the impact of time delays on carryover problems, time delays should be assessed relative to the time between two generations because problems that are not solved before the introduction of the next generation product will be carried over. The time between two generations is the number of months from the introduction of one generation to the introduction of the next generation.

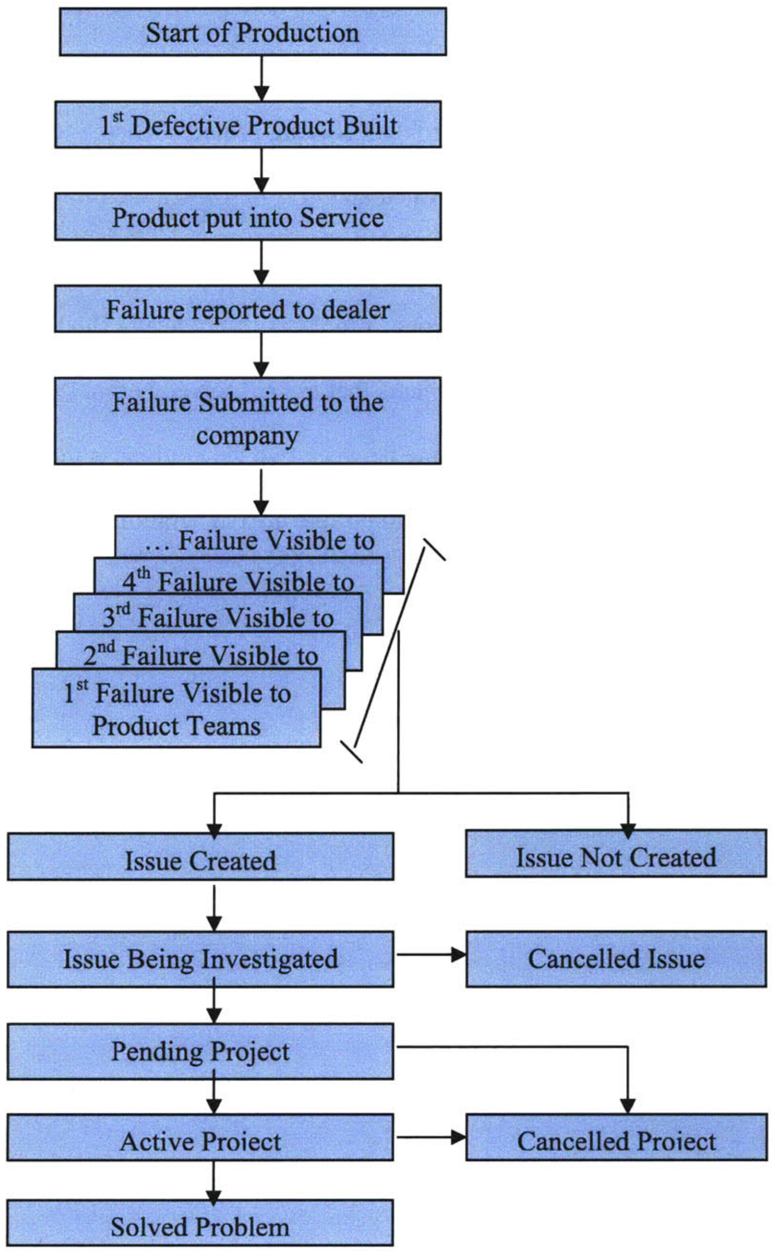


Figure 3: Breakdown of the time delays in problem identification and solution.

We analyzed data from 42 products and 2,024 problems to compare these time delays to the time elapsed between the two generations. The results show that, on average, these delays are very long compared to the time between two generations, and field problems are solved very close to the end of production (Figure 4). The median problem is solved after 91% of the time between two generations, which means that solving the median problem will benefit only 9% of the production life cycle. Given a constant production schedule, this would mean that the solution to the problem would only help 9% of the current generation's products. Figure 5 shows the distribution of these time delays. Sixty-four percent of problems are solved before the end of production for the current generation, meaning that 36% of problems remain unsolved during the production life cycle of the current generation. Problems that are solved before design freeze, which is the target date for stopping any changes to design during new product development project, comprise 41% of solved problems.

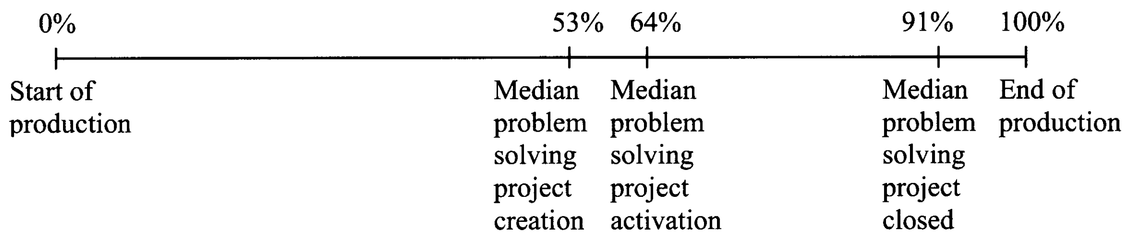
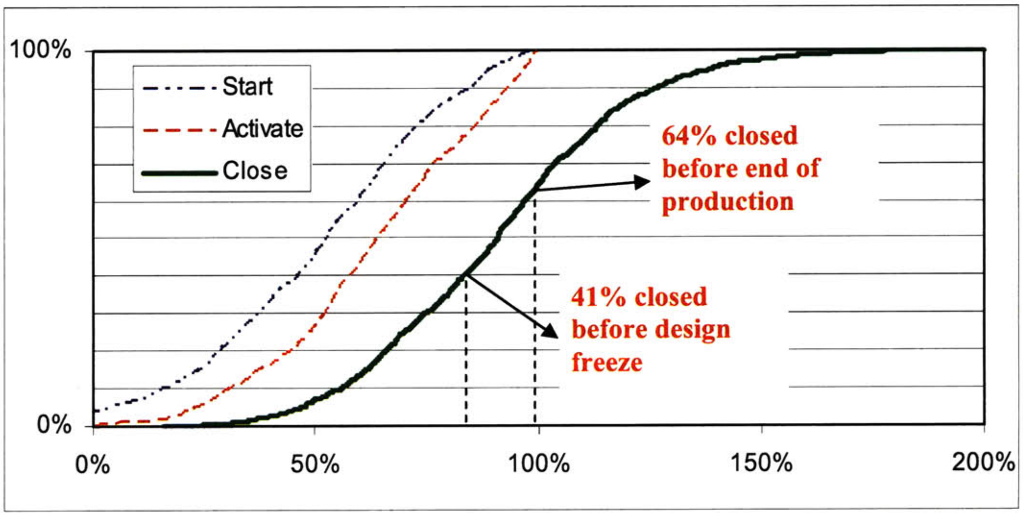


Figure 4: Median field problem solving project creation, activation, and close times relative to the time between start and end of production for product generations

Y-axis: Percent of closed problem solving projects



X-axis: Percent of time between production start and end

Figure 5: Distribution of time to start, activate, and close field problem solving projects relative to the time between start and end of production for product generations

According to these results, problems get carried over to the next generation because of the time delays involved in the system. This increases the need to understand the delays involved in solving a problem. To this end, we analyzed the length of different steps in the process. The results show that the time to learn about the problems is a major time delay. Once engineers learn about the problems, they are able to solve them in a relatively short period of time. Figure 6 shows that the average time it takes to learn about problems, which is the time between the build date of the first defective product and the third failure reported to the company, is approximately 2 years. The average time between the third failure reported and the problem being closed is 9 months. This time frame was recently reduced from 2 years to 9 months as a result of process improvement efforts and continuously improves. Prior to process improvement, the average time that elapsed between the first defective product build date and the problem being closed was 4 years, and was on the order of the time between two generations. Currently, field problems are solved very late because of the "mechanics" of the system – not because field problem solving teams are slow.

Since field problems are identified and solved very late, field problem solving has a small impact on the current generation's reliability, and problems get carried over to the next generation.

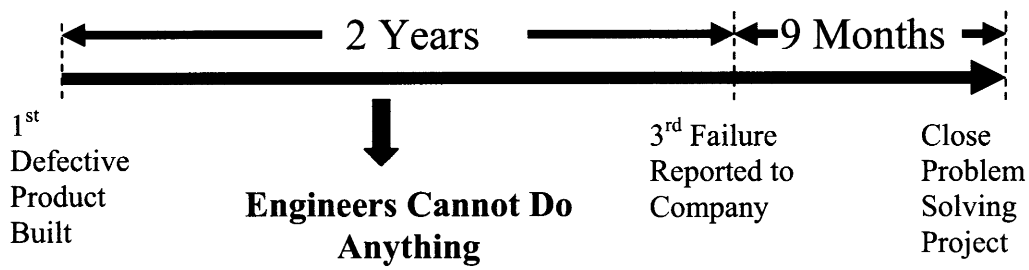


Figure 6: Time from first defective product build date to problem solving project close date

These findings show that long time delays in learning about and solving field problems increase the number of problems that get carried over to the next generation. As previously mentioned, the median problem is solved after 91% of production is already complete and, thus, has little benefit to the current generation product. However, if a carryover problem is solved when 91% of production is already complete, it will benefit not only the remaining 9% of the current generation's production but also the entire (100%) production life cycle of the next generation, making a much greater impact. In this case, solving the carryover problem will yield 12.11 (=109/9) times more benefit than solving a non-carryover problem, assuming a constant production schedule and equal production life cycles for successive generations. This finding shows the need for more emphasis on solving carryover problems. The following section will analyze the impact of carryover parts on new product introduction spike.

2.5 Analysis of the New Product Introduction Spike

We tested the hypothesis that carryover parts do not contribute to the new product introduction spike. The new product introduction spike is defined as the ratio of defects per machine in the next-generation products produced in the first 12 months after their introduction to defects per machine in the previous generation of products produced in the last 12 months of their production. Only mature machines are used for computing the new product introduction spike. The new product introduction spike metric is used by the firm to assess the success of new product introductions in terms of reliability.

The hypothesis that carryover parts do not contribute to new product introduction spikes implies two things: first, that carryover parts have the same defects per machine in successive generations, and second, that when a new generation is introduced, it is only the new parts that are responsible for the deterioration of reliability. To test this hypothesis, we computed the spike for carryover parts only.

The mean value for the carryover spike was 1.51, and the median was 1.32, meaning that the carryover parts fail more often in the next generation (Table 2). Statistical tests strongly reject the hypothesis that there is no carryover spike:

H0: Carryover spike = 1

H1: Carryover spike $\neq$ 1

The p-value for the t-test is less than 0.0001. Therefore, we conclude that carryover parts, which have the same design across successive generations, contribute to new product introduction spikes by having a higher failure frequency in next-generation machines. We also analyzed the spike for new parts by comparing their reliability to the reliability of the parts that they replace. The mean value for the new parts' spike was 1.27, and the median was 1.19. These results show that the mean and median values of the carryover parts' spike are significantly higher than the mean and median values of the new parts' spike (two-tailed paired t-test p-value = 0.0003, Wilcoxon signed-rank test p-value=0.0006). This is a very counterintuitive finding because it shows that carryover parts not only start failing more from one generation to the next but also that their reliability deterioration is worse than that of new parts. The reasons for this finding are being investigated, but preliminary analysis identified the following two reasons:

1. System integration issues: The parts' design is the same, but next-generation machines are more powerful and demanding, so the failure rate of the carryover parts increases.
2. Procurement or manufacturing process issues: The parts' design is the same, but changes in procurement or manufacturing processes increase the failure rate.

	Overall New Product Introduction Spike	Carryover Parts' Spike	New Parts' Spike
N	125	125	125
Average	1.32	1.51	1.27
Median	1.25	1.32	1.19

Table 2: New product introduction spike summary statistics for all parts of products as well as the carryover parts and the new parts.

The findings presented in this section show that carryover problems have a drastic influence on new product reliability due to the time delays involved in identifying and solving them. Moreover, the failure rate of carryover parts grows from one generation to the next. Therefore, policies to prioritize carryover problems and to discover them earlier in the process may significantly improve the reliability of products. Later in the paper, we will present and test three policies that aim at better prioritization and analysis of carryover problems.

3. Qualitative Data Analysis

In addition to the time delays, a second reason for the substantial impact of carryover problems on new product reliability was the lack of communication between problem solving teams. Our research partner had separate teams responsible for solving field problems and for solving problems discovered during new product development projects. These two systems were merged after this study was conducted. Carryover problems were the joint responsibility of these teams because field failure data was reported to field problem fixing teams, but fixing carryover problems ideally required a consideration of the next-generation product's design, too. Otherwise, the solution implemented might not be compatible with new parts designed for the next generation. However, our interviews indicated that there was no well-defined, standard procedure to address carryover problems. When asked about carryover problems, managers responsible for field problem fixing processes defined carryover problems as "problems not fixed by new product introduction teams." On the other hand, managers responsible for new product introduction processes defined carryover problems as "problems not fixed by field problem solving teams." One engineer that had assignments in both the field problem solving domain and the new product introduction domain said, "The 'as is [carryover] process' is fairly weak in my mind…There's really no documented process in terms of how people address carryover [problems]." Therefore, a lack of coordination among these teams was another factor that contributed to the impact of carryover parts on product reliability. Lack of coordination was present despite very strong incentives established by senior management to improve products' reliability. Short-term incentives tied a portion of employees' benefits to products' reliability. Moreover, all business units were asked to improve reliability in the long run following a glide path. In the following section, we present a simulation model to analyze the impact of carryover-related problem solving policies on failures.

4. Model Structure

Empirical analysis shows that carryover problems are a major source of failures and warranty costs. Furthermore, we show that the process of identifying and solving carryover problems involves significant time delays. As a result, it is very important to effectively manage the process of problem identification and resolution. In this section, we present a simulation model in order to quantify and analyze the impact of different problem solving policies on failures.

Our simulation model captures the process of identifying and fixing field problems. It is different than traditional models of problem identification and resolution in the sense that it is agent-based and each problem, defect, and failure is modeled as a separate agent. Traditional models formulate these variables at an aggregate level (Ford and Sterman, 1998; Repenning, 2001) and do not capture the heterogeneity of problems; some problems lead to a much greater number of failures than others. The disaggregated structure of our simulation model allows us to capture differences between problems in several dimensions, such as differences in expected failures, remaining production schedule, whether or not the problem is going to be carried over, etc. Capturing those differences is important in testing different policies regarding prioritization of problems and analysis of failure data. Another distinct feature of the model is that it keeps track of all failures for each problem and uses this failure information to estimate the number of expected failures that will be generated by each problem if it is not solved. Engineers are only aware of failures that are reported to them and do not know how many of the machines already produced will eventually fail. In some policies they use Weibull analysis to estimate the fraction of produced machines that will fail in relation to a specific problem. This analysis gives them an estimate of the economic value of solving each problem. Specifically, they fit Weibull distribution to operating hours until failure date and use Weibull analysis results to estimate the percentage of machines that will fail. Weibull analysis is a method widely used by reliability engineers for this purpose. We also run Weibull analysis within the model periodically for each problem. Figure 7 shows an overview of the simulation model.

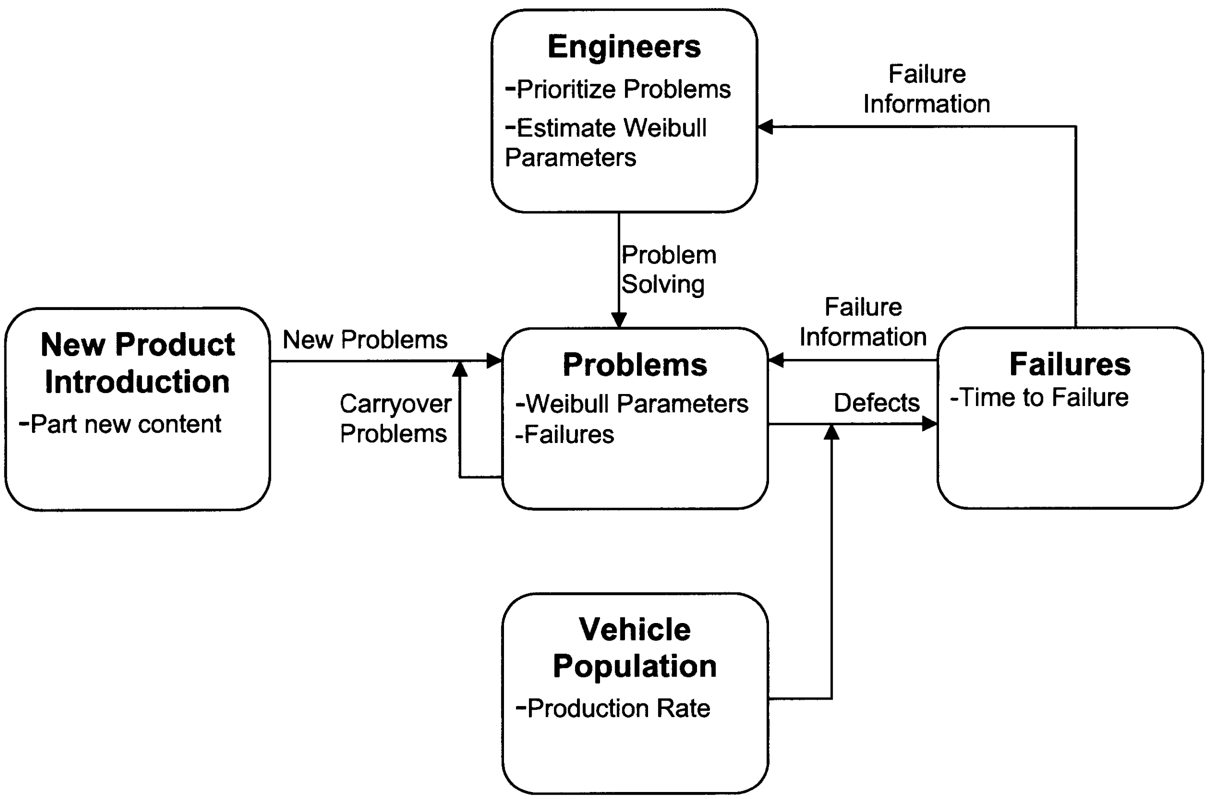


Figure 7: Overview of the simulation model

4.1. Field Problem Identification and Solving

In the model, problems generate defects in some products as they are built, until the problem is solved or the product goes out of production. These defects, and hence the problems, are unknown to engineers until they fail. Once the problem has a certain number of failures, it is investigated. Problems that are considered worth solving are then solved by engineers. The following subsections describe related model structures in detail.

4.2 Defect Creation and Failures

Each problem has different characteristics in terms of the fraction of produced machines that will fail and their operating hours until failure. In the reliability engineering literature, Weibull distributions are used extensively to represent the distribution of operating hours until failure. Since the expected fraction of machines that will fail is equal to the probability of a failure happening within the maturity period, this distribution also models the fraction of machines that will fail. The probability density function of the Weibull distribution is as follows:

Probability density function:

$$f(x;k,\lambda) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}$$

for $x > 0$ and $f(x; k, \lambda) = 0$ for $x \leq 0$, where $k > 0$ is the slope parameter and $\lambda > 0$ is the characteristic value.

Figure 8 shows a Weibull analysis plot of failures within the maturity period of the product, which occurs at 10,000 hours of operation. This plot is used to estimate the fraction of machines that are expected to have a failure before 10,000 hours of operation. According to the plot, 10.95% of the machines are expected to fail.

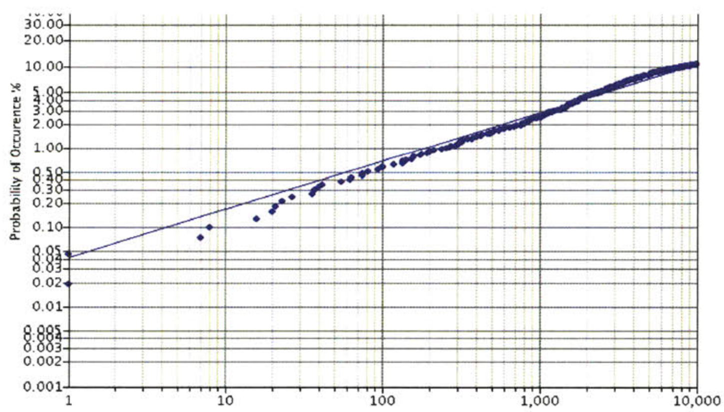


Figure 8: Weibull analysis plot of failures for a sample problem

We use maximum likelihood estimation and ranked regression methods to estimate the parameters of the Weibull distribution. If all machines have failures associated with a problem, an estimation is made using failure data for all vehicles and the operating hours until failure for each machine. However, in many cases, a failure is not observed. In those cases, information indicating that the machine has been in use to date and has not failed is useful information. In reliability engineering terminology, these cases are called suspensions. The maximum likelihood estimation exploits both failure and suspension information to estimate the distribution of operating hours to failure.

The likelihood function for cases that include suspensions is as follows:

$$L = \prod_{i=1}^{n} f(x_i;k,\lambda) \prod_{j=1}^{p} \left[1 - F(y_j;k,\lambda)\right]$$

where n is the number of failures, x_i is the operating hours to failure of ith failure, p is the number of suspended data points, y_j is the operating hours to date of the jth suspension, and $F(y_j;k,\lambda)$ is the cumulative distribution function.

Using empirical data from all problems related to a representative product of our research partner, we estimated the Weibull parameters of these problems. Then, we used the parameter estimates in the simulation model to introduce defects to some machines probabilistically according to the following algorithm: for each produced machine in a given month, we drew a random number for each problem from its estimated Weibull distribution of operating hours to failure. If operating hours to failure drawn for a machine-problem pair is within the maturity period of the product, then we created a defect on that machine associated with that problem. The defect would fail when the machine reached the operating hours to failure. The algorithm for the defect creation process in a given month can be summarized as follows:

```
For i=1 To "Vehicles produced this month"
  For j=1 To "Number of Problems"
     Operating hours to Failure(i,j)= Random number from Weibull(Slope(j),Characteristic Value(j))
     If
       Operating hours to Failure (i,j)<Maturity period of product
     Then
```

Create a defect that will fail after "Operating hours to Failure(i,j)"

Else

Do nothing

Calendar months to failure for a defect is as follows:

$$\text{Sales Delay} + \frac{\text{Hours to Failure}}{\text{Operating Hours per Month of the Vehicle}} + \text{Reporting Delay}$$

There is substantial heterogeneity in terms of the daily operating hours of vehicles used by customers. This heterogeneity is important because it influences the calendar time during which the vehicle will be within its maturity period. There is also considerable heterogeneity in sales delays. Both daily operating hours and sales delays are modeled as random variables that follow the Weibull distribution and are estimated using real daily operating hours data and sales data for products with maximum likelihood estimation.

We ran behavior tests for the Weibull assumption. These tests were conducted by running the model 1,000 times using estimated Weibull parameter values for each problem. No attempt was made to minimize the discrepancy between model output and actual data. Figure 9 shows actual failure data for three generations of a product and maximum and minimum values of the 1,000 simulation runs. Average R^2 values were 0.92, 0.88, and 0.88, respectively, for time series fit of the three generations. We also regressed simulated failures of problems in each run on actual failures of those problems. Figure 10 shows results of one simulation run out of 1,000 runs. Average R^2 values were 0.98, 0.99, and 0.97, respectively, for three generations. A high level of fit between data and model output and the fact that most data points lie within the envelope of minimum and maximum simulated values increases our confidence in the Weibull assumption.

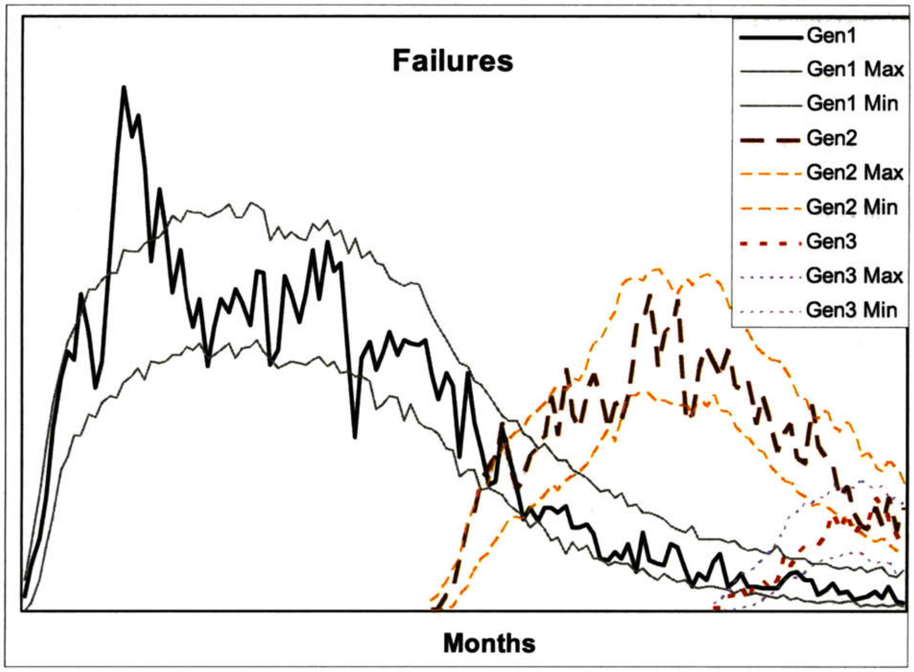


Figure 9: Model calibration results for three successive generations of a product. Thick lines represent actual data and thin lines are the max and min values of the 1,000 simulations run with estimated Weibull parameters.

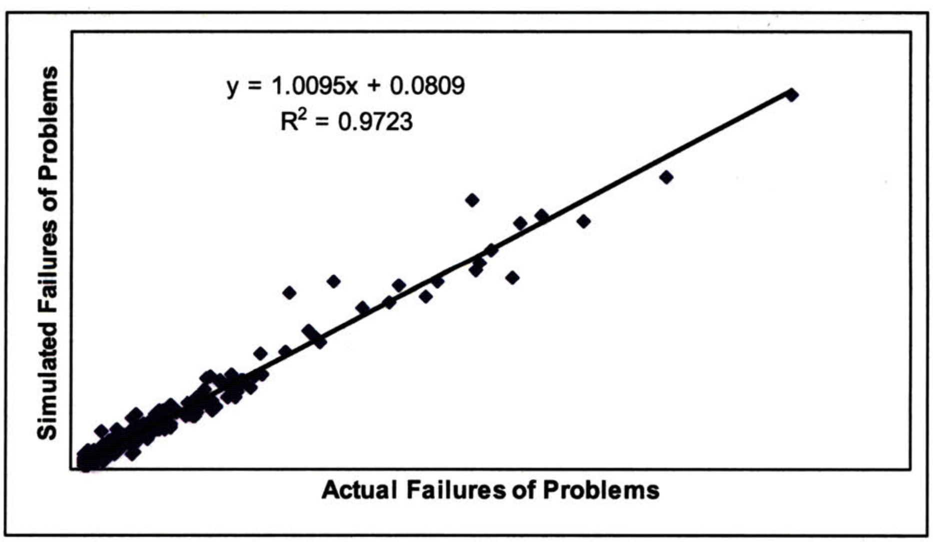


Figure 10: Simulated failures of all problems of generation 1 using estimated Weibull parameters versus actual failures of problems and the best fitting linear model. Data from one simulation run out of 1,000.

4.3 Problem Solving Process

Our research partner follows a systematic process for problem solving. In the model, problems go through the same process as well. It is a queuing system in which problems enter the queue as they lead to failures and are then solved by engineers. A problem is unknown to quality and reliability teams until the first failure, after which it becomes a known problem. After the third failure, it becomes an issue that needs to be investigated. An investigation is conducted to determine if the problem is a "real" problem worth solving. Some investigated issues are canceled because they are deemed not to be real problems or are deemed unworthy of solving. Our research partner initiates a formal project to solve problems that are not canceled. These projects then enter the queue to be worked on and become pending projects. If there is an available problem solving team, a pending project will begin immediately. Otherwise, the project waits in the queue until a problem solving team finishes a project already underway and chooses to work on the queued project. Which problem should be worked on when there is an available team to do so is an important policy issue, and it will be addressed later. Problems chosen from the pending list become active projects. At this point, resources are formally assigned to those projects. Active projects that are not canceled and get solved become "solved problems." Solved problems are useful in improving the reliability of machines that will be produced after the solution is implemented. The following state chart in Figure 3 shows the different states a problem will pass through within the process described above.

4.4 New Problem Introduction and Problems Getting Carried Over

A new generation's problems consist of both new problems due to new design and problems carried over from the previous generation. The number of problems carried over depends on the percentage of new parts in the next generation. If there were no new parts, all problems would be carried over. If all parts were new, all problems would be new problems. We assumed a linear relationship between the fraction of new parts and new problems introduced; thus, the number of new problems introduced increases linearly as the fraction of new parts increases. The fractions of carryover parts and new parts add up to 1, so as the fraction of new parts increases, the fraction of carryover parts decreases. We assumed that the probability of a problem being carried over to the next generation if it was not solved was equal to the fraction of carryover parts. If the fraction of carryover parts is 0, no parts are carried over, and the probability of a problem being carried over to the next generation is 0. On the other hand, if the fraction of carryover parts is 1, all problems are going to be carried over; hence, the probability of a problem being carried over is 1. Engineers do not know if a part, and hence any problem associated with it, will be carried over to the next generation until the bill of materials is released. In the model, some problems are identified as carryover problems with the probability of being carried over equal to the fraction of carryover parts. The earliest time at which engineers can learn whether or not a problem is a carryover problem is the release date of the bill of materials for the next generation. The following equations illustrate the relationships described above:

New Problems = Fraction of New Parts * Problems of a Completely New Product

P(problem(i) will be carried over to the next generation) = Fraction of Carryover Parts = 1 - Fraction of New Parts

The empirical results shown earlier demonstrate that carryover problems result in more failures in next-generation products; this phenomenon is called the carryover spike. We modified the defect creation algorithm to capture this finding. The number of random variables drawn from the Weibull distribution for defect creation was increased to machines produced * carryover spike for carryover problems, but the algorithm ensures that the number of failures does not exceed the number of machines produced:

```
For n = machines produced this month * carryover spike
If
  Operating hours to failure drawn from the Weibull distribution of the problem < Maturity period of
  product
Then
  Create a defect that will fail after the operating hours to failure drawn from the Weibull Distribution
  Created Defects = Created Defects+ 1
  If(Created Defects) = machines produced this month
    Break
```

Else

Do nothing

Data analysis shows that most of the time the carryover spike is due to manufacturing and assembly related issues and these problems get solved by plant employees outside of the scope of field problem solving. On average, the carryover spike's impact vanishes in three years. Therefore, we reduce the impact of carryover spike linearly over time and reduce its impact to zero in three years.

4.5 Problem Prioritization

Engineers in our research site use several tools to prioritize problems including objective metrics such as the number of failures, Weibull analysis results, sophisticated algorithms, and subjective metrics such as the expected difficulty of solving problems and political issues within the organization. Therefore, the decision to choose which problems to work on is a complex one. We analyzed these decisions in essay 2 by using survival analysis to understand the factors that influence the time between issue creation and project activation. Survival analysis also yields hazard rates as a function of time and characteristics of problems. Hazard rate is the probability that the project is activated at time t given that it was not activated until time t. We use results from essay 2 to compute the hazard rate of problems and rank them accordingly. The problem with the highest rank gets activated.

5. Parameter Estimation and Model Validation

Parameters are estimated for a product from our research partner's product portfolio that is representative in terms of its reliability and production volume. Managers in our research site chose this product as a representative one. All parameters used in the model were estimated using the dataset we created. The parameters used in the model are as follows:

- Number of problem solving teams
- Hours to failure distribution estimated for each problem
- Operating hours per month distribution for machines, estimated for the machine population
- Sales delay distribution, estimated for the machine population

- Reporting delays, estimated using process data
- Investigation delays
- Time to solve problems: distribution estimated for problems
- Production volume
- Maturity period of product
- Time between two generations
- Fraction of new parts in new products

Both behavior tests and structural tests such as extreme condition tests were conducted to test the model's validity (Sterman, 2000).

6. Policy Analysis

As noted earlier, the time delays in the process of fixing field problems are quite significant. Therefore, it is very important to leverage all information about problems as much as possible. In this section, we present the results of three different policies aiming at better usage of carryover information. The first policy gives more priority to problems that will be carried over to the next generation. The second policy focuses on problems that are carried over from the previous generation. It uses failure information from the previous generation machines to estimate the importance of problems carried over to this generation as early as possible. The third policy considers solving problems that will be carried over to the next generation only for next-generation products, without solving them for the current generation. This policy will reduce the time it takes to solve these problems, as engineers will have more flexibility in fixing next-generation problems and will not spend time implementing changes in the current generation. Currently, these policies are not implemented by our research partner. We will first explain the three policies and then present the simulation results. Note that all of these policies are tested with the same number of engineers in the model.

6.1 Policy 1: Increasing the Priority of Problems that Will Be Carried over to the Next Generation

The model finds expected defects for all known problems, ranks problems according to expected defects, and begins to solve the highest-ranked problem as soon as a problem solving team becomes available. Expected defects are estimated using the following equations:

Expected Defects = Expected Defects Introduced per Month * Period in which Fixing the Problem will Have an Impact

Expected Defects Introduced per Month = Expected Fraction of Machines that Will Fail * Average Production Rate

Expected Fraction of Machines that will Fail = Prob(HourstoFailure<EconomicLifeofMachines|WeibullParametersEstimated)

Expected fraction of machines that will fail is computed using Weibull parameters estimated by ranked regression. See essay 3 for details of the estimation method. Period in which Fixing the Problem Will Have an Impact is different for carryover problems and non-carryover problems. Carryover problems will be in production until the end of the current generation's production life cycle and the entire production life cycle of the next generation. On the other hand, non-carryover problems will be in production until the end of the current generation's production life cycle. Hence, there is considerable difference between the period during which carryover and non-carryover problems will be in production. All else being equal, the number of expected failures is higher for carryover problems, since they will impact the entire production life cycle of the next generation as well as the remaining production life cycle of the current generation.

Carryover problems will have the following equation under Policy 1:

Period in which Fixing the Problem Will Have an Impact = (Remaining Production Time for this Generation + Time between Generations - Median Time to Fix Problems)

For non-carryover problems, the "period in which fixing the problem will have an impact" is:

Period in which Fixing the Problem Will Have an Impact = (Remaining Production Time for this Generation - Median Time to Fix Problems)

Engineers learn which parts will be carried over to the next generation only after the release of the bill of materials for the next generation. Hence, carryover and non-carryover problems have the same period in which fixing the problem will have an impact before the release of the bill of materials. Note that even though carryover problems have a longer "period in which fixing the problem will have an impact" under Policy 1, some non-carryover problems will have higher priority than some carryover problems if their number of expected defects introduced per month is large enough.

6.2 Policy 2: Using Previous Generation's Failure Data to Predict the Importance of Problems Carried over from the Previous Generation

Usually, a new generation product contains carryover parts with problems that were not solved before the introduction of the new generation. It is also possible that failures occur on carryover parts after the introduction of the new generation. Typically, these are the problems resulting in failures after very long time delays. Policy 2 aims to use failure information from the previous generation machines to estimate what might be coming for the current generation. Since these problems cause failures after a very long time delay, it will be too late if engineers wait until failures are seen in the new generation. This policy aims to solve these problems as soon as possible, even if they have not yet caused failures in the current generation.

If this policy is implemented in the model, when a problem is carried over to the next generation, we keep track of all failures that occur in the previous generation and combine them with failures of the current generation to run the Weibull analysis. The Weibull analysis results are then used to rank problems. If this policy is not implemented in the model, a new problem is created for each carryover problem when the product transitions from one generation to the next. The new problem has the same Weibull parameters, but the engineers are not aware of the association between the carryover problem in the new generation and the one in the old generation. Therefore, they do not use failure data from the old generation while running the Weibull analysis for a problem in the new generation.

Empirical analysis shows that this policy immensely increases the amount of available information. Figure 11 shows failure data for identical parts in two successive generations. The blue curve shows the information available from the previous generation. The pink curve shows the information available from the new generation. The red curve is the combined failure information from the two. The combined information is much richer and provides much more abundant data compared to second-generation information only.

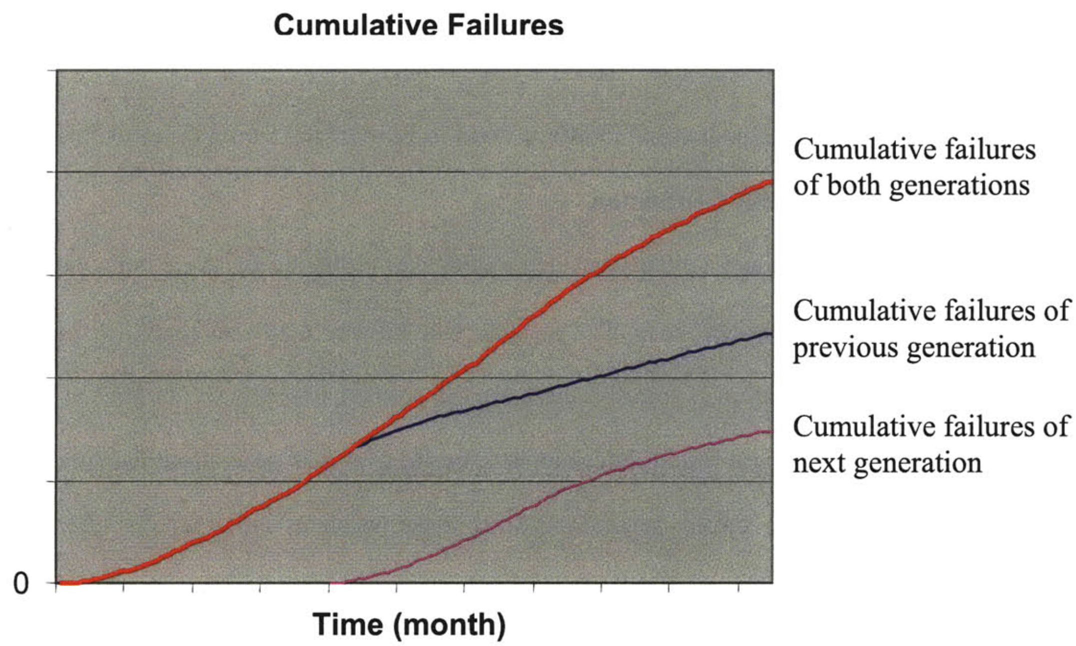


Figure 11: Cumulative failures of two successive generations of a product. The blue curve represents the previous generation and the pink represents the next generation. The red represents the total failures of the two generations.

Not considering the previous generation's information regarding carryover parts corresponds to using only the failure data indicated by the pink curve to estimate the Weibull parameters of problems. On the other hand, using information from both the previous generation and the new generation corresponds to using the failure data shown by the red curve. Figure 11 shows that the difference in information availability is dramatic.

6.3 Policy 3: Solving Carryover Problems for Only Next Generation

In this policy, engineers are given the option to design and implement the fix for a problem for the next generation only and not for the current generation. The obvious disadvantage of this policy is that it will not improve the reliability of the current generation. The advantage of this policy stems from decreased effort in designing and implementing the fix. Since the fix does not need to take into account the design of the current generation, engineers will have a greater degree of freedom in designing the fix. A change in the current generation requires designing the fix so that it functions well with the interacting parts of the current generation, which is challenging, since the current generation is already in production. It might even lead to a redesign of the interacting parts of the current generation or changes in manufacturing processes.

Designing the fix only for the next generation will be easier, especially if the next generation's design is not complete, since engineers will have fewer constraints. Based on interviews with expert reliability engineers, we assumed that the policy of fixing carryover problems only for the next generation would lead to a 25% reduction in the time it takes to solve problems. The experts stated that 50% of the time is spent on problem identification and root cause analysis, and that would be the same in both cases. However, they estimated a 50% reduction in the actual problem solving activity, resulting in a 25% reduction in the time required to solve the problem. In the model, this policy is implemented by ranking problems according to the number of expected defects per "effort for solving the problems." We find the expected number of defects for problems and then divide that number by the average time to solve problems. If this policy is implemented, the average time to solve carryover problems would be 75% of the average time to solve non-carryover problems.

7. Simulation Results

We ran scenarios that tested these policies one by one, as well as their combinations. The base scenario, which represents the practice of our research partner, is obtained from the survival analysis results of essay 2. In the base case, carryover problems were not prioritized, failure information from the previous-generation products was not used, and carryover problems were solved for both the current generation and the next generation. The time between generations was assumed to be 50 months, in line with the data, and we ran the simulation for 250 months. Since the simulation model has probabilistic formulations, 100 simulations were run for each scenario and averages were compared. Table 3 summarizes the average values of total defects created in different scenarios.

Simulation results of all scenarios are presented in Table 3. Using carryover information for prioritization reduces the number of defects by 12%. When combined with the option to solve problems for the next generation only, total reduction in defects increases to 17%. Interestingly, using carryover information from the previous generation increases the number of defects by 5%. However, when this policy is combined with the policies of using carryover information for prioritization and having the option to solve problems for the next generation only, the combined benefit of the three policies is a 25% reduction in defects, showing that there are synergies. This is a drastic improvement in quality obtained without increasing resources. These improvements translate into a drastic increase in profitability because, in many motor vehicle companies, warranty costs are on the order of hundreds of millions or billions of dollars (Warranty Week, 2007).

Use Carryover Information For Prioritization	Have the Option to Solve for Next Generation Only	Use Carry Over Information From Previous Generation	Total Defects in 250 Months
No	No	No	100%
Yes	No	No	88%
No	No	Yes	105%
Yes	Yes	No	83%
Yes	No	Yes	78%
Yes	Yes	Yes	75%

Table 3: Results of the policies tested with the simulation model. All policies are significantly different than the base case scenario, with significance levels less than 10^{-4}.

8. Impact of Industry Trends on Carryover Parts' Reliability

In several industries, including our research partner's industry, product development lead times are getting shorter (Smith and Reinertsen, 1998) due to the adoption of different organizational structures, project management tools, and design and testing technologies. This leads to more frequent product introductions and less time between two generations of a multi-generation product. In this section, we attempt to quantify the impact of this trend on carryover defects. Less time between two generations of a multi-generation product could have two potential impacts on carryover problems. First, fewer problems will be identified before the next generation is introduced due to the time delays in learning about these problems. Second, engineers are going to have less time to fix carryover problems.

The simulation results presented in the previous section assume that the time between two generations is 50 months. Our research partner reduced the time between generations considerably over time. We tested the implications of this change on the benefits of our policies by comparing the previous simulation results to simulations in which the time between two generations is 25, 75, or 100 months. Note that engineers learn which problems will be carried over to next generation 12, 36, or 48 months before the start of production for the next generation respectively. The fraction of defects due to carryover parts is shown in Figure 12, which illustrates that carryover problems will become increasingly important in the future as product development lead times continue to shrink across industries.

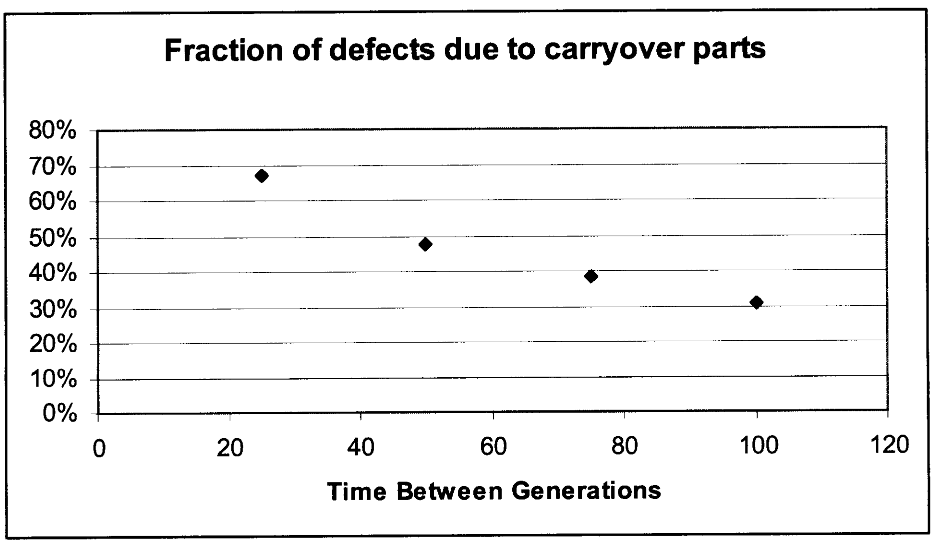


Figure 12: Simulation results when the time between generations is 25, 50, 75, and 100 months.

9. Relevance for Practitioners

Our research partner's managers found the empirical results surprising. One product group checked the validity of the results and reached the same conclusions. Earlier drafts of this paper were read by managers, and they described the results as an "accurate representation" of the company.

Efforts to put more emphasis on carryover problems are underway based on our empirical findings and simulation results. Initial attempts include efforts by a large team of managers and engineers to define a standard process for addressing carryover problems across the company, stopping some new product development projects until carryover problems are solved, preparing the IT infrastructure to make carryover problems more visible, and assigning engineers to keep track of and be accountable for carryover problems.

10. Conclusion

Our data analysis showed that carryover parts used in successive generations of multi-generation products are a substantial source of unreliability. Organizational coordination challenges and long time delays involved in learning about these problems increase the extent and number of problems carried over to the next generation. Since carryover problems have a significant impact on the reliability of next-generation products, the field problem solving process that solves carryover problems is an important determinant of the reliability of new products. Carryover problems and field problem solving have not received enough attention in the literature, and our results show that research on new product development will benefit from a better understanding of these issues.

We tested different policies to employ field problem information more effectively using a simulation model. The model's novel structure, which represents each problem separately and keeps track of all failures, allowed us to capture substantial differences in problems in terms of total failures. This structure also allowed us to estimate the Weibull parameters periodically and to predict the expected failures for all problems. These features of the model were instrumental in testing prioritization policies aimed at better selection of problems to work on and testing policies designed to identify upcoming problems earlier.

The managerial implications of our findings indicate the need for more attention to be focused on problems associated with carryover parts. Better coordination is needed between engineers working on field problems and new product development engineers. Our results show that these teams should work together to minimize problems carried over to the next generation. This requires targets for reducing carryover problems, clear communication channels, joint responsibility for carryover problems, and transparent IT systems that give both teams access to problem information, production plans, and bills of materials for new products.

Other implications of our findings involve problem prioritization and identification. For prioritization, managers should consider both the percentage of products that are expected to fail and the number of machines that are going to be produced while that problem is open. Since carryover problems are in production for a much longer period of time than non-carryover problems, carryover problems should have higher priority. On the other hand, problems that have caused failures in previous-generation products should be investigated to determine whether or not they will lead to failures in the current generation. If these problems are expected to lead to failures in the current generation, teams should solve these problems without waiting for them to lead to failures in the current generation in order to prevent as many failures as possible.

Our data was collected at only one company, and this is a limitation of our paper. However, our research partner has dozens of products that span a wide spectrum of motor vehicles, from relatively basic machines to very complicated products. These products are exposed to vastly different environments. Moreover, there are several subsidiaries within the company with individual profit and loss responsibility, leading to different practices and cultures. These factors increase the generalizability of our results. Furthermore, we show that as product development lead times get shorter, carryover problems will have a greater impact on new product reliability, which strengthens our conclusion that managers should pay greater attention to carryover problems in a wide range of industries. In fact, our qualitative analysis revealed that one of the reasons why our host organization lacked effective processes and policies for dealing with carryover problems is that carryover is a relatively new phenomenon. When product development lead times were much longer, teams had more time to solve field problems; hence, carryover was a less important issue. Thus, we expect that companies in several other industries adjusting to shorter product development lead times will also benefit by improving their carryover processes and policies.

References

Brown SL, Eisenhardt KM. 1995. Product development: past research, present findings, and future directions. Academy of Management Review, 20(2) 343-378.

Clark KB, Fujimoto T. 1991. Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. Harvard Business School Press: Boston, MA, USA.

Collier DA. 1982. Aggregate safety stock levels and component part commonality. Management Science, 28(11) 1296-1303.

Ford DN, Sterman JD. 1998. Dynamic modeling of product development processes. System Dynamics Review, 14(4) 309-340.

Griffin A. 1997. The effect of project and process characteristics on product development cycle time. Journal of Marketing Research, Special Issue on Innovation and New Products, 34(1) 24-35.

Lee HL. 1996. Effective inventory and service management through product and process redesign. Operations Research: Special Issue on New Directions in Operations Management, 44(1) 151-159.

Repenning NP. 2001. Understanding fire fighting in new product development. The Journal of Product Innovation Management, 18 285-300.

Robertson D, Ulrich K. 1998. Planning for product platforms. Sloan Management Review, Summer, 19-31.

Smith PG, Reinertsen DG. 1992. Shortening the product development cycle. Research-Technology Management, May-June, 44-49.

Smith PG, Reinertsen DG. 1998. Developing Products in Half the Time: New Rules, New Tools. John Wiley & Sons: New York, NY, USA.

Sterman JD. 2000. Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin/McGrawHill: Boston, MA, USA.

Thomke S. 2003. Experimentation Matters: Unlocking the Potential of New Technologies for Innovation. Harvard Business School Press: Boston, MA, USA.

Warranty Week. 2007. Retrieved September 10, 2008, from http://www.warrantyweek.com/archive/ww20070515.html

Essay 2: New Product Introductions, Field Problem Solving and Product Reliability

Abstract

In this paper, we empirically analyze the field problem-solving process and the new product introduction spike. More specifically, we address the following question: what factors influence the time required to solve problems? The problem-solving process is composed of two main parts: the time taken before dedicating resources, meaning the time spent in the queue while the problem awaits resources, and the time between dedicating resources and the problem resolution date. Different factors might have different impacts on these two parts of the process. Therefore, we analyze these two parts separately. Interestingly, regression results show that the importance of a problem has little influence on the time spent between dedicating resources to a problem and the resolution date. Its influence on the time spent before dedicating resources is greater. A problem's importance also exerts considerable influence on the cancellation probability. More important problems have lower chances of being cancelled. In addition to these questions related to the field problem-solving process, we also analyze the new product introduction spike and find that the amount of late changes significantly correlates to the new product introduction spike.

1. Introduction

Multi-generational products are used in several industries ranging from durables to consumer packaged goods (Griffin, 1997). Since multi-generational products are produced under the same brand name for a long time, customer perceptions about their quality and reliability changes slowly. Therefore, the quality and reliability of these products has a considerable influence on their market share. Figure 1 shows warranty costs as a percentage of revenue for major automotive manufacturers from 2003 to 2006. The lead of Toyota and Honda in terms of reliability and durability as shown in Figure 1 is one of the reasons for their steadily increasing market share.

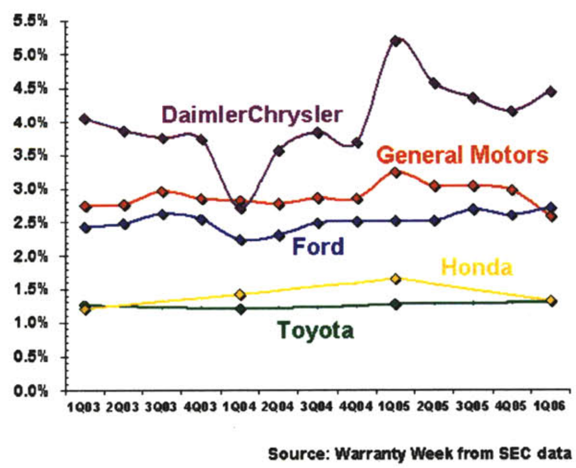


Figure 1: Warranty costs as a percentage of revenue for major automotive manufacturers from 2003 to 2006

Quality improvement processes for multi-generational products include activities conducted during new product development projects and after new product introduction. During new product development, project teams try to prevent and solve problems through analysis and testing. After new product introduction, failures that occur in the "field" while being used by the customers are reported to the manufacturers and the manufacturer then tries to solve these problems. This process is called field problem solving. Field problem solving for the current generation of a product is undertaken simultaneously with the new product development project for the next generation product. When the new product development project is completed, production starts for the next generation product.

Figure 2 shows the number of defects per manufactured product versus production time, a metric for the reliability of products. After the introduction of a new generation, reliability is improved through field problem solving. When the next generation is introduced, reliability deteriorates, a phenomenon known as a "new product introduction spike," and is then improved again through field problem solving. Reliability dynamics, as seen in Figure 2, are representative of most multi-generation products. There is a time delay between the production date of a defective product and its failure date while being used by the customer. Figure 3 shows the number of failures reported over time.

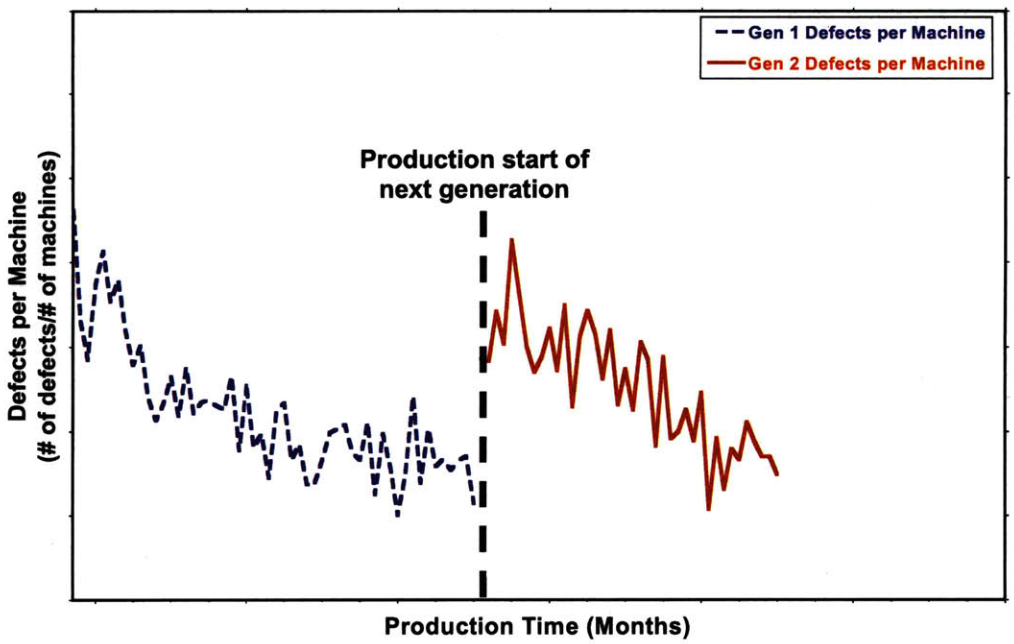


Figure 2: Defects per machine versus production date of products

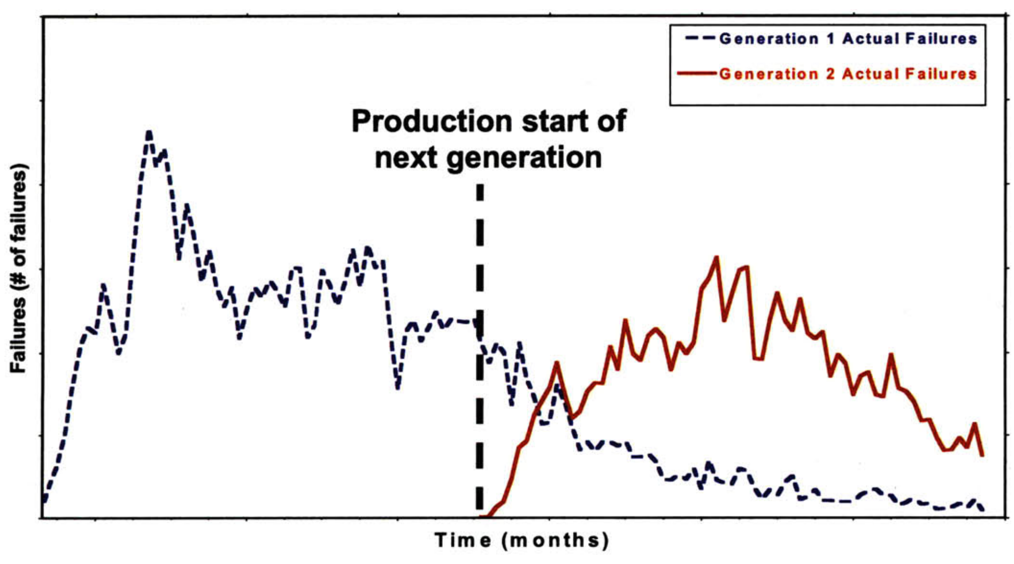


Figure 3: Failures reported to quality and reliability teams versus time

Even though there is a large body of literature that focuses on problem-solving activities during new product development projects (Clark and Fujimoto 1991, Thomke 2003; Wheelwright and Clark 1992), the literature is mostly silent about the field problem-solving process and the new product introduction spike. However, both the new product introduction spike and the field problem-solving process are crucial for the quality and reliability of multi-generational products and are the foci of this paper.

The field problem-solving process deserves more attention in the literature since it is crucial for many companies and has important differences compared to the new product development problem-solving process. First, data about field problems come from a much larger sample, all products sold to the customers, than new product development projects. During new product development projects, test sample sizes are usually limited due to testing costs. Second, since field failure information comes from customers, it represents the entire spectrum of applications the products are exposed to. In most product development projects it is not feasible to replicate all operating environments of products. Third, unlike new product development projects, the pace of problem identification during the field problem-solving process is not controlled by the manufacturer. During new product development projects, the project team decides on the analysis and testing schedule, and hence controls the problem identification rate. Contrarily, after the start of production, field problems are identified by evaluating customer complaints.

Each problem addressed by our research partner is a formal problem solving project, and the problem-solving teams follow a well defined process. Therefore, our data set gives us the opportunity to analyze a scientific approach to problem solving. The field problem-solving process starts with the reporting of the first failure of a problem to the manufacturer. After a number of failures are reported, a problem attracts the engineers' attention, depending on the nature and timing of the problems and an "issue" is created. First, the problem is investigated to understand if it is a real problem that should be addressed. If the answer is yes, the problem waits for resources. Otherwise it gets cancelled. A problem becomes an "active project" when a team of employees becomes available to work on it. When the problem-solving project is activated, the project team first tries to understand the root cause by examining failed product parts returned by customers. Then, they try to find a solution to the problem, validate the proposed solution by analysis and testing, and close the project if they conclude that the proposed solution is indeed effective. If the solution is effective, products produced after the implementation of the fix will not have failures related to the solved problem. During the process, a project might be cancelled if the team was late in solving it and the problem goes out of production, if further investigation indicates that there is no need to address it, or if a cost/benefit analysis shows that the problem fixing cost will exceed the benefits.

Figure 4 shows the process for solving field problems. Note that the process has two main parts. In the first part, resources are not yet assigned to the problem; it is waiting for resources in a queue. In the second part, resources are assigned to the problem and the project is active.

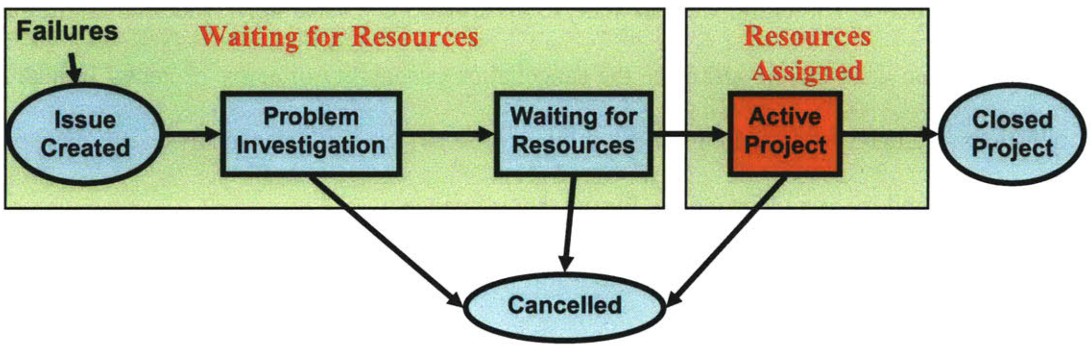


Figure 4: Process for solving field problems

The speed of this process and the efficiency of the use of resources are crucial for improving the product reliability. If problems are solved faster, fewer products will fail. On the other hand, minimizing the cancellation rate of active projects improves the efficient use of resources. A cancelled active project is a waste of resources since a team of employees already expended effort on a project that did not yield benefits.

In addition to the field problem-solving process, we will also analyze the new product introduction spike. The new product introduction spike is an important metric used by many companies for assessing the performance of new product development projects in terms of reliability. In this paper, the spike is defined as the ratio of the average reliability of products manufactured within the first year after the introduction of a new generation to that of products manufactured within the last year prior to the introduction of the new generation.

This paper contributes to the literature by empirically analyzing the field problem-solving process and new product introduction spike. We will first try to answer the following question: what factors influence the time to solve problems? As mentioned before, the problem-solving process has two main parts; the time between issue creation and project activation, meaning the time spent in the queue while the problem is waiting for resources, and the time between project activation and the problem close date in which resources are actively working on the problem. Different factors might have different impacts on these two parts of the process. Therefore, we will analyze these two parts separately. The second question we will try to answer is: what factors influence the cancellation probability of active projects? In addition to these questions related to the field problem-solving process, we will also analyze the new product introduction spike and try to understand the factors that influence it.

In the following sections, we will summarize our data and report analysis results.

2. Data

The data was collected over a period of three years at a major motor vehicle manufacturer. The manufacturer is a multi-billion dollar company that has several subsidiaries manufacturing a wide range of products at several facilities around the world.[1] Thousands of engineers work on product development projects and solving product problems. Throughout the production life cycle of an average product, thousands of machines are produced. We collected both qualitative and quantitative data from the organization. The qualitative data includes 68 formal, semi-structured interviews with 58 employees ranging from junior engineers to the VP of R&D. We recorded and transcribed all of these formal interviews. Even though we had a generic topic and a list of interview questions, these questions were open-ended, and we changed the focus if the interviewee raised an interesting point. In addition to these formal interviews, we conducted dozens of informal interviews that were not recorded.

To analyze the field problem-solving process and new product introduction spikes, we created a unique data set consisting of thousands of problems and more than a hundred new product introductions. The next section describes the data in more detail and presents empirical analysis results.

[1]To preserve the confidentiality of the company, exact figures are not presented.

3. Empirical Analysis

In this section, we will first analyze the field problem-solving process and then new product introductions.

3.1 Field Problem Solving Process

Our dataset consists of 5872 activated projects. Some of these projects were closed, some were cancelled, and the rest were still active at the time of data collection. We will use this data set to analyze factors that influence the time it takes to solve problems and the factors that influence the probability of canceling problems. As mentioned before, the time it takes to solve problems has two parts. The first part is the time between issue creation and problem activation. During this period, the problem is waiting for resources. The decision by engineers to work on a problem terminates this period. On the other hand, from project activation to problem close, engineers are actively working on the problem, trying to find the root cause, design a fix, validate it, and implement it. This part is terminated when the team finds the fix to the problem and implements it. Since these two parts have important differences, they will be analyzed separately.

We identified the factors that might influence the time it takes to solve problems and the cancellation probability of an active problem by interviewing employees involved in the quality and reliability of products. The following factors are used:

Subsidiaries: Our research partner has several subsidiaries. Which subsidiary is responsible for solving the problem might influence the process due to differences in their circumstances. We used subsidiary dummies to represent this factor. Also, different product and component types (e.g. engine, hydraulics etc.) might have different impacts on the process. For each problem, we know the fraction of failures associated with different product and component types. We included these fractions in our analysis.

Team characteristics: The experience level of the engineer that leads the project with the problem solving process and the size of the team might influence the problem-solving process. The number of projects solved by the lead engineer before starting a project is our proxy for the experience level of the lead engineer in the problem-solving process. Team size is the number of team members. As number of team members increases, it might be more difficult to coordinate, increasing the time to activate and close problems.

For each problem, we used an average workload proxy at the subsidiary level in addition to the subsidiary fixed effect. Workload is not a fixed effect and changes over time for each subsidiary. Our data set includes only the number of projects the lead engineer is working on simultaneously, and we do not have data on the projects of other team members. Since our project data is on lead engineer level but the problems are solved by teams, we used an aggregate measure for workload at the subsidiary level. This metric is the average active projects per lead engineer in each subsidiary during the period in which the teams are actively working on a problem. We also used the average number of projects the lead engineer of works on as a proxy for lead engineer workload. Note that some lead engineers do not spend all of their time on problem solving.

Problem Complexity: Complexity of a problem is another factor that might influence the problem-solving process. One metric we used is the difficulty level defined by the team members as Easy, Medium, or Hard. Other metrics we used are related to the analysis and testing of problems and the implementation of the solution. The number of failures that occurred before project activation might influence the analysis process since it might be easier to find the root cause with more failures. Testing of a solution might be more expedient if the problem has a lower average operating hours to failure rate, since test results can be obtained faster. Implementation might be more difficult if the fix needs to be implemented in more plants. Also, if the number of parts involved in a problem is higher, it might be more difficult to find the root cause, and it may take more time to implement the design changes required to implement the fix. In addition to making it more difficult to solve active projects, problem complexity might increase the time between issue creation and project activation because the teams might not be very eager to work on difficult problems.

Importance of a Problem: The importance of a problem might influence the problem-solving process since the problem-solving teams might activate more important problems sooner and expedite the problem-solving process after project activation. They might also be more hesitant to cancel more important projects. The metrics used to represent the importance of a problem are the following: our research partner's internal score assigned to each problem for ranking the problem priority, the number of failures and warranties at the time of project activation, and the expected benefits of solving the problem on warranty costs, sales, and prices.

Correlation coefficients of the variables are presented in Appendix 1. Coefficients are not high between variables, except with failures at activation and warranty costs at activation. Note that the correlation between the subsidiary workload between project creation and activation and subsidiary workload between project activation and close is very high but these variables are not used in the same regression model.

3.1.1. Time Between Issue Creation and Project Close: The faster a project is closed, the more benefits the effort will yield because the fix will be implemented on more products. We analyzed the impact of the factors explained in the previous section on the time between issue creation and project closure. As mentioned before, the time between issue creation and project activation, and the time between project activation and problem close will be analyzed separately. The model we will use for testing hypotheses regarding the impact of the before mentioned factors on time between issue creation and project activation and time between project activation and close has the following form:

Duration = f(Subsidiary, Team Characteristics, Problem Complexity, Importance of a Problem, Other Factors).

Ordinary Least Squares (OLS) has several drawbacks in this setting. First, our data set contains censored observations. If the project was not closed at the time of data collection, that data point is censored because we do not know for how long it will be open. However, we do know that it has been open for a certain amount of time, which is useful information to consider for the analysis. OLS cannot handle these cases. Second, the normality assumption of OLS is usually violated in cases similar to ours. Consequently, we used survival analysis, also known as event history modeling (Sentas et. al. 2008). The advantage of survival analysis is to be able to accommodate censored data points and a wide range of distributions that allow different hazard functions.

Since we are interested in the predictions regarding the time it takes to close projects, we used parametric duration models, namely accelerated failure time models. Accelerated failure time models have the following specification:

$$\ln(t_j)=x_j\beta_x+\varepsilon_j$$

Models are named according to the distribution assumed for $e^{\varepsilon j}$. Different distributions have different implications for the underlying hazard rate. Hazard function is the probability that the failure happens at time t, given that it did not happen until time t. Exponential has a constant hazard rate whereas Weibull is monotonic and lognormal, loglogistic and gamma are variable. We used the log-likelihood values and AIC metrics to choose among these distributions. AIC is a metric that rewards a high likelihood value, but rewards parsimony as well. Also, lognormal, weibull, and exponential are special cases of gamma. Therefore, we tested hypotheses regarding the appropriateness of these three distributions using gamma parameter estimates. We will present the results in the following two subsections.

3.1.1.1 Time Between Issue Creation and Project Activation: We tested several distributions with different implications for the hazard rate. Log-likelihood and AIC values are almost identical for gamma, Weibull, and exponential distributions, and are better than that of lognormal and loglogistic (Table 1). All coefficient values and significance levels are very close for all of these regressions. Exponential is a special case of gamma distribution, and a kappa value of 1 and sigma value of 1 of gamma distribution corresponds to exponential. In the gamma regression, confidence intervals for kappa and sigma include 1; therefore, we cannot reject the hypothesis that the model is exponential. Hence, we used exponential regression in this case.

A project is activated as soon as any lead engineer starts to work on that problem. Assignment of a lead engineer to the problem is done right before project activation by the manager responsible for the field problem-solving process. Since the assignment of the lead engineer to the project is done at the end of the period between issue creation and project activation, we did not include lead engineer related variables such as the average number of projects or experience level.

Detailed exponential regression results are presented in Table 2. Subsidiary dummy, fraction of failures associated with different product types, fraction of failures associated with different component types, and start year dummies are not reported for brevity. According to likelihood ratio test results, they are all significant with p-values less than 0.0001, except fraction of failures associated with different component types which has a p-value of 0.0018. Note that summary statistics of variables and constant terms in regressions are not presented to preserve the confidentiality of our research partner.

	Gamma	**Exponential**	**Weibull**	**Lognormal**	**Loglogistic**
Hard Problems	0.196***	0.196***	0.196***	0.191***	0.187***
MediumProblems	0.180***	0.180***	0.179***	0.229***	0.210***
Team Members	0.0211***	0.0211***	0.0210***	0.0179***	0.0199***
Plants	0.0189*	0.0189*	0.0189*	0.016	0.0181*
Parts	0.00024	0.000243	0.000246	-0.000188	-0.000252
Average Hours to Failures	0.000142*	0.000141*	0.000141*	0.000181*	0.000182*
Warranty Benefits	-3.28E-08	-3.30E-08	-3.32E-08	-1.29E-08	-1.33E-08
Sales Benefits	1.21E-08	1.22E-08	1.22E-08	1.28E-08	6.60E-09
Score	-0.000123***	-0.000122***	-0.000122***	-0.000151***	-0.000153***
Subsidiary Workload	0.0865***	0.0868***	0.0872***	0.0538*	0.0454*
ln(sigma)	-0.00771			0.201***	
Kappa	0.981***				
ln(p)			0.0122		
ln(gamma)					-0.381***
N	5026	5026	5026	5026	5026
Log-likelihood	-7861	-7861	-7861	-8140	-8115
AIC	15868	15863	15864	16423	16373

Table 1: Results for lognormal, loglogistic, gamma, weibull and exponential regressions. The dependent variable is the time between issue creation and project activation. * denotes $p<0.05$, ** $p<0.01$, *** $p<0.001$.

```
Exponential regression -- accelerated failure-time form
No. of subjects =         5026                 Number of obs    =       5026
No. of failures =         5026
Time at risk    =  947224.8556
                                               LR chi2(73)      =    2489.83
Log likelihood  =   -7861.7257                 Prob > chi2      =     0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
HardProblems	.196233	.0459775	4.27	0.000	.1061188	.2863472
MediumProbs	.1798233	.0365587	4.92	0.000	.1081695	.251477
teammembers	.0210508	.0043128	4.88	0.000	.0125977	.0295038
plants	.0188708	.0075741	2.49	0.013	.0040258	.0337157
parts	.0002431	.0009621	0.25	0.801	-.0016427	.0021288
avghrstoflrs	.0001414	.0000582	2.43	0.015	.0000274	.0002554
warrbnfts	-3.30e-08	4.23e-08	-0.78	0.435	-1.16e-07	4.99e-08
salesbnfts	1.22e-08	1.10e-08	1.10	0.271	-9.48e-09	3.38e-08
score	-.0001225	.000012	-10.25	0.000	-.0001459	-.0000991
subsworkload	.0868481	.0206943	4.20	0.000	.046288	.1274082

Table 2: Exponential regression results

Easy problems are used as the base line for complexity dummy and hence coefficients for medium and hard problems represent incremental duration compared to easy problems. Regression results show that hard and medium problems are activated significantly later than easy problems. All else being equal, lead engineers might be tempted to work on easier problems since they receive the same benefits with less effort. The number of team members increases the time to activate projects, probably due to coordination challenges of larger teams. If more plants are involved in a problem, the project is activated later. Again, the reason might be coordination challenges between engineers in different plants. Higher average operating hours to see failures of problems delays the activation date of problems. The longer time delay to learn about failures might lengthen the investigation process conducted prior to activating the project. Problems with a higher score are activated sooner, since it will be more beneficial to solve those problems as soon as possible. Lower subsidiary level workload during the period between issue creation and project activation leads to faster project activation.

Exponential regression results are used in essays 1 and 3 to simulate the decision-making process of our research partner. Hazard rates of problems are calculated in the simulation model using the significant factors in Table 2, and the problem with the highest hazard rate is activated.

In addition to survival analysis, we also ran OLS and compared the results (Appendix 2). The dependent variable was the natural logarithm of the time between issue creation and project activation.

Since it has the same functional form with the survival analysis, coefficients are interpreted in the same way. OLS results are consistent with survival analysis results in terms of the significance of the coefficients and the estimates for significant coefficients.

We tested the parametric specification of our survival analysis model by comparing exponential regression results to the Cox proportional hazards model. The Cox proportional hazards model is a semi-parametric model and has the following form for the hazard function:

$$h(t, x_j)= h_0(t)\exp(x_j\beta_x)$$

The reason that the Cox proportional hazards model is semi-parametric is that it does not make any assumptions about the shape of the underlying hazard rate $h_0(t)$. As mentioned previously, exponential regression assumes that $h_0(t)$ is fixed. Note that the accelerated failure time and proportional hazard metrics for exponential regression yield the same information. The difference is that the accelerated failure time metric shows the impact of variables on the time to activate a problem whereas the hazard rate metric shows the impact on the probability to be activated. Parameters have the same absolute value but the opposite sign between the two metrics for exponential regression. Results of the Cox proportional hazards model are compared to the exponential regression in Table 3. Note that the parameter values and significance levels are very close.

	Cox	**Exponential**
Hard Problems	-0.208***	-0.196***
MediumProblems	-0.183***	-0.180***
Team Members	-0.0224***	-0.0211***
Plants	-0.0203*	-0.0189*
Parts	-0.000406	-0.000243
Average Hours to Failures	-0.000142*	-0.000141*
Warranty Benefits	3.31E-08	3.30E-08
Sales Benefits	-1.35E-08	-1.22E-08
Score	0.000122***	0.000122***
Subsidiary Workload	-0.106***	-0.0868***

Table 3: Exponential regression results and the Cox proportional hazards model results. The dependent variable is the time between issue creation and project activation. * denotes $p<0.05$, ** $p<0.01$, *** $p<0.001$.

Next, we will assess the practical impact of each variable on the time it takes to close projects and analyze the practical significance of these results. Due to model specification:

$$\ln(t_j)=x_j\beta_x+\varepsilon_j$$

$$t_j=\exp(x_j\beta_x)\exp(\varepsilon_j)$$

$$t_j=\exp(x_1\beta_1+ x_2\beta_2+\ldots+ x_k\beta_k)\exp(\varepsilon_j)$$

the marginal impact of a variable (say x_1) is to multiply the time it takes to close the project by $\exp(\beta_1)$:

$$t_j^*=t_j\exp(\beta_1)$$

Table 4 shows the percent impact of one standard deviation increase in significant factors on the time it takes to close the projects.

	Impact of One StDev Increase	Impact of 50% Increase compared to Average
Hard Problem	22%*	
Medium Problem	20%*	
Score	-15%	-4%
Subsidiary Workload	11%	20%
Number of team members	9%	9%
Number of plants	5%	2%
Average operating hours to see failures	5%	4%

Table 4: Percent impact of one standard deviation and a 50% of average increase in significant factors on time between issue creation and project activation. Note that the impact values reported for hard and medium problems in this table correspond to an increase from 0 to 1.

Problem complexity, subsidiary workload, and the score has the highest impact on the time between issue creation and project activation.

3.1.1.2 Time Between Project Activation and Problem Close: Gamma regression has the highest log-likelihood value and the lowest AIC (Table 5), so we used gamma regression for analyzing the time between project activation and project close.

	Gamma	Exponential	Weibull	Lognormal	Loglogistic
Hard Problems	0.427***	0.419***	0.375***	0.504***	0.489***
Medium Problems	0.336***	0.330***	0.289***	0.409***	0.392***
Team Members	0.0743***	0.0792***	0.0713***	0.0772***	0.0771***
Plants	0.0563***	0.0613***	0.0540***	0.0559***	0.0593***
Parts	0.00591***	0.00634***	0.00603***	0.00548***	0.00586***
Average Hours to Failures	0.000325***	0.000347***	0.000328***	0.000300***	0.000317***
Warranty Benefits	-2.21E-08	-2.11E-08	-1.90E-08	-2.42E-08	-2.40E-08
Sales Benefits	2.71E-09	7.06E-09	5.29E-09	-2.43E-09	-6.89E-10
Score	-0.000036**	-0.000040**	-0.000037***	-0.000036**	-0.000038**
Warranty at Project Activation	2.59E-08	2.46E-08	2.11E-08	3.55E-08	3.31E-08
Failures at Project Activation	-0.000068**	-0.000068**	-0.000065**	-0.000077**	-0.000072**
Subsidiary Workload Between Project Activation and Close	0.103***	0.121***	0.113***	0.0802***	0.0905***
Engineer Experience Level at Activation	-0.0155***	-0.0170***	-0.0147***	-0.0166***	-0.0164***
Engineer Average Number of Problems Between Project Activation and Close	-0.00720*	-0.00965*	-0.00870*	-0.00374	-0.00469
ln(sigma)	-0.160***			-0.0137	
Kappa	0.618***				
ln(p)			0.264***		
ln(gamma)					-0.593***
Log-likelihood	-5628	-5823	-5658	-5720	-5681
AIC	11381	11767	11440	11563	11486

Table 5: Log-likelihood and AIC values for lognormal, loglogistic, gamma, weibull and exponential regressions

Detailed Gamma regression results are presented in Table 6. Subsidiary dummy, fraction of failures associated with different product types, and activation year dummies are not reported for brevity. According to likelihood ratio test results, the fraction of failures associated with different component types is not significant and hence is not included in the model. Others are all significant with p-values less than 0.0001, except activation year dummy which has a p-value of 0.04

Gamma regression -- accelerated failure-time form

No. of subjects =	5028	Number of obs =	5028
		No. of failures =	3479
		Time at risk =	1345585.18
		LR chi2(59) =	1287.93
Log likelihood =	-5628.4354	Prob > chi2 =	0.0000

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
HardProblems	.4268792	.0441817	9.66	0.000	.3402846	.5134738
MediumProbs	.3359656	.0341373	9.84	0.000	.2690577	.4028735
teammembers	.0743067	.0046062	16.13	0.000	.0652788	.0833346
plants	.056252	.008341	6.74	0.000	.039904	.0726
parts	.0059147	.0011588	5.10	0.000	.0036435	.008186
avghrstoflrs	.0003251	.0000535	6.07	0.000	.0002202	.0004301
warrbnfts	-2.21e-08	2.54e-08	-0.87	0.384	-7.18e-08	2.76e-08
salesbnfts	2.71e-09	1.16e-08	0.23	0.816	-2.01e-08	2.55e-08
score	-.0000363	.0000115	-3.16	0.002	-.0000588	-.0000138
warractivtn	2.59e-08	2.48e-08	1.05	0.296	-2.26e-08	7.44e-08
flrsactivtn	-.0000675	.0000213	-3.17	0.002	-.0001094	-.0000257
subsworkload	.102807	.0219338	4.69	0.000	.0598176	.1457964
engrexprnc	-.0155061	.0011554	-13.42	0.000	-.0177706	-.0132416
engavgproj	-.007198	.0036149	-1.99	0.046	-.0142831	-.0001129
/ln_sig	-.1602716	.0178755	-8.97	0.000	-.195307	-.1252362
/kappa	.6179674	.046759	13.22	0.000	.5263214	.7096134
sigma	.8519124	.0152284			.8225821	.8822885

Table 6: Gamma regression results. The dependent variable is the time between project activation and project close.

As mentioned previously, Weibull, lognormal, and exponential models are nested in the gamma model. A kappa value of 1 corresponds to Weibull and a kappa value of 0 corresponds to lognormal. Since the 95% confidence interval for kappa does not include 0 and 1, the hypotheses that the model is weibull or lognormal are rejected. A kappa value of 1 and a sigma value of 1 correspond to exponential, which is also rejected.

Regression results show that the factors that increase the difficulty of solving problems, such as complexity level, number of plants involved, number of parts, and average operating hours to see failures tend to increase the time it takes to solve the problem. The number of plants involved might slow the projects down due to the higher effort required to implement the fix in more plants. Average operating hours to see failures also increase the time to close projects, probably due to the longer testing and validation times required. Failures at the time of activation decrease the time to solve the problems. The reason might be that it is easier to find the root cause for problems with a higher number of failures. All of these factors that were considered so far are the ones that are related to the physics of the problem in the sense of influencing the time it takes to find the problem, fix it, and implement the fix. On the other hand, the score of the problem, a metric for the importance of the problem, is also significant. However, as we will see below, even though the score is statistically significant, its practical impact is not that high (Table 9).

The number of team members increase the time it takes to solve problems; possibly due to coordination issues. Lead engineer experience reduces the time it takes to close the problems, and it is significant. Subsidiary workload increases the time to close projects. The average number of projects the lead engineer worked on while the project was active is borderline significant and it is not significant when robust standard errors are used (Table 8).

We also ran the regression model by adding lead engineer fixed effects (Appendix 3). The results are very similar. In addition to these two models, we also ran OLS (Appendix 4). To alleviate the problem of OLS not handling censoring, we only used projects that were activated within the first year of our data set. 95% of the projects activated in that year were either closed or cancelled at the time of data collection. In the OLS model, the dependent variable was the natural log of the time between activation date and project closure. Since it has the same functional form with the survival analysis, coefficients are interpreted in the same way. Results are very similar to Gamma regression results.

By definition, Gamma regression does not have a corresponding proportional hazards representation. However, exponential regression has a corresponding proportional hazards representation, and its coefficients and significance levels are very close to Gamma regression. Therefore, we compared the Cox proportional hazards model results to the exponential regression results (Table 7). The results are very close, showing that the parametric hazard function assumption is plausible.

	Cox	Exponential
Hard Problems	-0.485***	-0.419***
Medium Problems	-0.373***	-0.330***
Team Members	-0.0917***	-0.0792***
Plants	-0.0692***	-0.0613***
Parts	-0.00773***	-0.00634***
Average Hours to Failures	-0.000413***	-0.000347***
Warranty Benefits	2.42E-08	2.11E-08
Sales Benefits	-6.69E-09	-7.06E-09
Score	0.0000480**	0.0000402**
Warranty at Project Activation	-2.56E-08	-2.46E-08
Failures at Project Activation	0.0000824***	0.0000681***
Subsidiary Workload Between Project Activation and Close	-0.145***	-0.121***
Engineer Experience Level at Activation	0.0189***	0.0170***
Engineer Average Number of Problems Between Project Activation and Close	0.0112*	0.00965*

Table 7: Exponential regression results and the Cox proportional hazards model results. The dependent variable is the time between project activation and close. * denotes p<0.05, ** p<0.01, *** p<0.001

To account for the possibility that the projects of a lead engineer are correlated due to some unobserved factors, we computed robust standard errors by clustering projects at the lead engineer level (Cleves et. al. 2004). In Table 8, we compare the results of Table 6 with the results clustered at the lead engineer level. Results are almost identical except for warranty benefits and "engineer's average number of problems between project activation and close". Warranty benefits become significant with clustered results, whereas "engineer's average number of problems between project activation and close" becomes insignificant.

	Gamma Regression	Cluster
Hard Problems	0.427***	0.427***
Medium Problems	0.336***	0.336***
Team Members	0.0743***	0.0743***
Plants	0.0563***	0.0563***
Parts	0.00591***	0.00591***
Average Hours to Failures	0.000325***	0.000325***
Warranty Benefits	-2.21E-08	-2.21E-08*
Sales Benefits	2.71E-09	2.71E-09
Score	-0.0000363**	-0.0000363**
Warranty at Project Activation	2.59E-08	2.59E-08
Failures at Project Activation	-0.0000675**	-0.0000675***
Subsidiary Workload Between Project Activation and Close	0.103***	0.103**
Engineer Experience Level at Activation	-0.0155***	-0.0155***
Engineer Average Number of Problems Between Project Activation and Close	-0.00720*	-0.0072

Table 8: Gamma regression and clustered regression results. Clustered results are at the lead engineer level.

Table 9 shows the impact of one standard deviation increase and a 50% increase compared to the average in significant factors in Table 8 on the time it takes to close projects.

	Impact of One StDev Increase	Impact of 50% Increase compared to Average
Hard Problem	53%*	
Medium Problem	40%*	
Number of team members	32%	35%
Lead Engineer Experience	-21%	-9%
Number of plants	13%	5%
Subsidiary Workload	13%	26%
Average operating hours to see failures	12%	9%
Number of part numbers	10%	2%
Number of failures at the time of activation	-6%	-1%
Score	-5%	-1%
Warranty Benefits	-1%	0%

Table 9: Impact of one standard deviation increase and a 50% of average increase in significant factors on time between project activation and problem close. Note that the impact values reported for hard and medium problems in this table correspond to an increase from 0 to 1.

Table 9 shows that the importance factors that are supposed to expedite the process such as the score, warranty benefits of solving the problem, and the number of failures at the time of activation have a limited effect compared to other factors. Even increasing the score from the lowest possible value to the highest possible value decreases the time it takes to close the project by 17%, a small amount compared to the one standard deviation increase of other factors that are related to the physics of solving the problem. The score has a much larger impact on the time between issue creation and project activation. Increasing it from the lowest possible value to highest possible value decreases the time between issue creation and project activation by 46%, a much more dramatic reduction in time compared to the time between project activation and problem close.

Regression results regarding the time between project activation and close are used in essays 1 and 3 to simulate the problem-solving process by influencing the time to solve problems.

3.1.2. Probability of Canceling an Active Project: We analyzed the impact of the factors used in the previous section on the probability of canceling an active project using logistic regression. To minimize the impact of censoring on our analysis, we limited the data set to projects that were activated within the first year of our data set. 95% of the projects activated in that year were either closed or cancelled at the time of data collection. We also ran the same analysis using all data points, and the qualitative results did not change. The logistic regression results are shown in Table 10:

```
Logistic regression                               Number of obs   =       1835
                                                  Wald chi2(47)   =          .
                                                  Prob > chi2     =          .
Log pseudolikelihood = -746.75381                 Pseudo R2       =     0.1892

                          (Std. Err. adjusted for 351 clusters in engineerid)
------------------------------------------------------------------------------
             |               Robust
   cancelled |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
HardProblems |   .0418577   .2522084     0.17   0.868    -.4524617     .536177
 MediumProbs |   .3150806   .1601631     1.97   0.049     .0011668    .6289945
 teammembers |  -.0984675   .0557255    -1.77   0.077    -.2076875    .0107525
      plants |   .0659073   .0285396     2.31   0.021     .0099707    .1218439
       parts |   .0078633   .0043496     1.81   0.071    -.0006618    .0163883
avghrstoflrs |   .0015229   .0002135     7.13   0.000     .0011045    .0019414
   warrbnfts |  -5.93e-07   4.53e-07    -1.31   0.190    -1.48e-06    2.94e-07
  salesbnfts |   2.45e-07   1.85e-07     1.32   0.185    -1.18e-07    6.09e-07
 warractivtn |  -8.75e-09   1.61e-07    -0.05   0.957    -3.24e-07    3.07e-07
 flrsactivtn |  -.0000743   .0000888    -0.84   0.402    -.0002483    .0000996
       score |  -.0004367   .0001189    -3.67   0.000    -.0006698   -.0002037
subsworkload |   .3425303   .1169961     2.93   0.003     .1132221    .5718384
```

Table 10: Logistic regression results. Robust standard errors. Clustered at the lead engineer level.

According to the likelihood ratio test results, subsidiary and complexity dummies are significant (p-values<0.0003). The fraction of failures associated with different component types is also significant (p-value<0.0001). The fraction of failures associated with different product types is not significant (p-value=0.27), and hence is not included in the model. Coefficients of these dummy and control variables are not reported for brevity.

Interestingly, the coefficient for medium problems is significant, showing that there is a significantly higher chance of canceling a medium project compared to an easy project; however, the coefficient is not significant for hard problems. The coefficients for number of team members and number of parts are marginally significant. More team members decrease the chance of project cancellation; possibly due to increased peer pressure or an increased perception of sunk cost. More parts, more plants, and a longer time to see failures, increase the chance of project cancellation. A higher score decreases the cancellation probability, whereas the impact of failures and warranty at activation and expected warranty and sales benefits of fixing problems are not significant. Finally, subsidiary workloads increase the chance of cancellation.

Table 11 shows the additive impact of one standard deviation and a 50% of the average increase of significant variables on the probability of cancellation.

	Impact of One StDev Increase	Impact of 50% Increase compared to Average
Average operating hours to see failures	0.07	0.06
Score	-0.07	-0.02
Number of team members	-0.05	-0.04
Medium Problem	0.04*	
Subsidiary Workload	0.04	0.09
Number of plants	0.02	0.01
Number of part numbers	0.02	0.00

Table 11: Impact of one standard deviation and a 50% of average increase of significant variables on the probability of cancellation. Note that the impact value reported for medium problems in this table correspond to an increase from 0 to 1.

3.2 New Product Introduction Spike

Using data from 125 new product introductions, we analyzed the factors that influence the new product introduction spike. The new product introduction spike is the ratio of the average reliability of products manufactured within the first year after the introduction of a new generation to that of products manufactured within the last year before the introduction of the new generation. The number of failures in a representative operating hour range from all products produced in a certain period is divided by the number of products produced to calculate average reliability. Products that passed the operating hour range used for reliability calculations are called mature products. For the new product introduction spike calculations, we focused on mature products to get rid of biases that might occur due to products that did not pass the operating hour range for reliability.

We analyzed the impact of different factors on new product introduction spike using regression analysis. The factors we included in the analysis were the following:

Product Type Dummy: Which product family the product belongs to.

Average Production Volume: Monthly data showing the number of products built in that month.

Pilots per volume: The number of pilot machines used for testing the new design and new production processes normalized by the average production volume of the product.

Part New Content: Percentage of new parts in the next generation of the product. The remaining ones are carried over from the previous generation.

Late Change Index (LCI): is a proxy for the amount of late engineering drawing changes.

Arrangements: Slightly different versions of a product.

Table 12 shows regression results.

```
Linear regression                                     Number of obs =       125
                                                      F( 20,    103) =        .
                                                      Prob > F      =         .
                                                      R-squared     =    0.3439
                                                      Root MSE      =    .35991
------------------------------------------------------------------------------
             |               Robust
NPISpike     |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
PartNewContnt|    .677286   .4247097     1.59   0.114    -.1650255    1.519598
LCI          |   1.276715   .3896036     3.28   0.001     .5040276    2.049401
PilotsPerVol |   .0151565   .0428242     0.35   0.724    -.0697752    .1000882
AvgPrdctnVol |   .0028207    .001611     1.75   0.083    -.0003745    .0060158
Arrangements |   .0049332   .0085399     0.58   0.565    -.0120037    .0218701
```

Table 12: Regression results for the new product introduction spike. Robust standard errors. Product type dummy is not reported for brevity.

According to our regression results, LCI and product type dummy are the only significant variables. The likelihood ratio test for product group dummy has a p-value of 0.02.

There is strong evidence that the LCI has an impact on new product introduction spike. Part new content is correlated with LCI (Table 13) but does not have a significant impact on the new product introduction spike when both variables are included in the model. When LCI is excluded from the regression model, part new content becomes significant (Appendix 5). This shows that it influences the new product introduction spike through LCI.

Linear regression

Number of obs =	125
F(1, 16) =	6.46
Prob > F =	0.0218
R-squared =	0.1353
Root MSE =	.12561

LCI	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf. Interval]	
PartNewContnt	.2971001	.1168915	2.54	0.022	.0493012	.544899
Constant	.6257831	.0461616	13.56	0.000	.5279249	.7236414

Table 13: LCI regressed on part new content. Robust standard errors used and clustered at the product group level.

This is an interesting empirical finding that supports Repenning's (2001) analytical finding that quality and reliability suffer when product development teams are engaged in firefighting.

In Essay 1 of the thesis, we showed that the new product introduction spike is composed of two spikes; new content and carryover spikes. The new content spike represents the deterioration in reliability when obsolete parts are replaced with new parts. Carryover parts are common parts used in successive generations of multi-generational products. The carryover spike represents the change in reliability of carryover parts as the production of a new generation starts. One hypothesis we tested is that these two spikes are correlated. If these two spikes are negatively correlated, it means that there is a tradeoff between carryover reliability and new content reliability and policies that improve one hurts the other. If they are positively correlated, it means that there is no tradeoff between the two, and best practices that improve the overall spike positively impact both the new content and carryover spikes. Regression results show that the carryover spike and the new content spike are significantly and positively correlated (Table 14), supporting the argument that there is no tradeoff between carryover parts' reliability and new content reliability.

Linear regression

Number of obs =	125
F(1, 16) =	8.75
Prob > F =	0.0093
R-squared =	0.1981
Root MSE =	.7153

CO Spike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf. Interval]	
NC Spike	.8122322	.2745764	2.96	0.009	.2301562	1.394308
Constant	.4764657	.3117901	1.53	0.146	-.1844996	1.137431

Table 14: Carryover spike regressed on new content spike. Robust standard errors.

We also analyzed the impact of different factors on both spikes. Regression results are shown below.

Linear regression

Number of obs = 125
F(20, 103) = .
Prob > F = .
R-squared = 0.3704
Root MSE = .37954

NCSpike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
PartNewContnt	.5242687	.4825132	1.09	0.280	-.4326824	1.48122
LCI	1.274004	.4826082	2.64	0.010	.3168648	2.231144
PilotsPerVol	.0030986	.0375842	0.08	0.934	-.0714408	.0776379
AvgPrdctnVol	.0022669	.0015923	1.42	0.158	-.0008911	.0054249
Arrangements	.0068203	.0087546	0.78	0.438	-.0105424	.024183

Table 15: New content spike regressed on factors mentioned above. Robust standard errors. Product type dummy is not reported for brevity.

Linear regression

Number of obs = 125
F(20, 103) = .
Prob > F = .
R-squared = 0.2570
Root MSE = .7524

COSpike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
PartNewContnt	1.575449	.7819182	2.01	0.047	.0246982	3.126199
LCI	1.883835	.6393794	2.95	0.004	.6157769	3.151893
PilotsPerVol	.0680913	.0577264	1.18	0.241	-.0463954	.182578
AvgPrdctnVol	.0039428	.0026993	1.46	0.147	-.0014107	.0092963
Arrangements	-.0033177	.0137318	-0.24	0.810	-.0305516	.0239161

Table 16: Carryover spike regressed on factors mentioned above. Robust standard errors. Product type dummy is not reported for brevity.

The late change index is significant in both regressions. An interesting finding is that the part new content is significant for the carryover spike but not for the new content spike. This might be because a higher part new content leads to more designing and testing of new parts, increasing work pressure for both new parts and carryover problems, lowering the reliability of both but hurting the carryover parts more due to relatively fewer resources for carryover parts. Another interesting finding is that the coefficient for the LCI is bigger for the carryover spike. This might be because late changes have a more negative impact on carryover parts because carryover parts are ignored when the new product development team is engaged in firefighting and trying to solve new part problems under time pressure. However, the statistical significance of these findings should be tested. We tested these hypotheses by comparing the impact of all factors on the new content and carryover spikes. The null hypothesis was that the coefficients in the two regressions are equal. The following regression was run:

$$\begin{bmatrix} NCSpike \\ COSpike \end{bmatrix} = \begin{bmatrix} X \\ X \end{bmatrix} \beta_1 + \begin{bmatrix} 0 \\ X \end{bmatrix} \beta_2 + \varepsilon$$

If a coefficient in β_2 is significantly different than 0, it means that the variable associated with that coefficient has a significantly different impact on the new content and carryover spikes. The regression results are shown below. We do not reject the hypothesis that the factors have an equal impact on the new content and carryover spikes. This finding shows that the relationship between the factors and new content and carryover spikes are similar.

```
Linear regression                                      Number of obs =      250
                                                       F( 41,    206) =        .
                                                       Prob > F      =        .
                                                       R-squared     =   0.3072
                                                       Root MSE      =   .59588
```

Spike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
PartNewContnt	.5242687	.4825132	1.09	0.279	-.4270286	1.475566
PartNewContnt2	1.05118	.9188118	1.14	0.254	-.7603004	2.86266
LCI	1.274004	.4826082	2.64	0.009	.3225198	2.225489
LCI2	.6098309	.8010722	0.76	0.447	-.9695203	2.189182
Pilots	.0030986	.0375842	0.08	0.934	-.0710004	.0771975
Pilots2	.0649927	.0688833	0.94	0.347	-.0708139	.2007994
AvgPrdctnVol	.0022669	.0015923	1.42	0.156	-.0008724	.0054062
AvgPrdctnVol2	.0016759	.003134	0.53	0.593	-.0045029	.0078547
Arrangements	.0068203	.0087546	0.78	0.437	-.0104398	.0240805
Arrangements2	-.0101381	.0162852	-0.62	0.534	-.042245	.0219689

Table 17: Testing the hypothesis that the factors have the same impact on new content and carryover spikes. Variable names that end with 2 represent the test for the equality of the impact of that factor on new content and carryover spikes. Robust standard errors. Product type dummy is not reported for brevity.

4. Conclusion

In this paper, we analyzed the field problem-solving process and new product introduction spikes. We found that the impact of different factors on the time between issue creation and project activation is different compared to their impact on the time between project activation and problem close in terms of statistical significance and magnitude. Interestingly, the importance of a problem has less influence on the time spent between project activation and problem close compared to other factors. Its influence on the time spent between issue creation and project activation is bigger. It also has a considerable influence on the cancellation probability. More important problems have a lower chance of being cancelled. We also found that the amount of late changes is significantly correlated with the new product introduction spike. Therefore, the amount of late changes might be used as an indicator that predicts a high new product introduction spike.

Appendix 1: Correlation coefficients of the variables included in the analysis.

	Team Members	Plants	Parts	Average Hours to Failures	Warranty Benefits	Sales Benefits	Score
Team Members	1						
Plants	0.04	1					
Parts	0.04	0.22	1				
Average Hours to Failures	-0.08	0.14	0.18	1			
Warranty Benefits	0.06	0.01	0.04	0.01	1		
Sales Benefits	0.04	0.07	0.06	-0.02	0.29	1	
Score	0.14	0.07	0.11	0.03	0.24	0.28	1
Warranty at Project Activation	0.02	0.22	0.31	0.20	0.06	0.03	0.13
Failures at Project Activation	0.008	0.42	0.42	0.15	0.04	0.12	0.12
Subsidiary Workload Between Project Creation and Activation	0.04	-0.05	-0.11	-0.21	0.009	-0.02	-0.06
Subsidiary Workload Between Project Activation and Close	0.02	-0.05	-0.09	-0.11	0.03	-0.01	-0.05
Engineer Experience Level at Activation	0.03	-0.06	-0.07	-0.13	-0.004	0.001	-0.02
Engineer Average Number of Problems Between Project Activation and Close	-0.03	-0.07	-0.03	-0.02	0.004	-0.01	-0.05

	Warranty at Project Activation	Failures at Project Activation	Subsidiary Workload Between Project Creation and Activation	Subsidiary Workload Between Project Activation and Close	Engineer Experience Level at Activation	Engineer Average Number of Problems Between Project Activation and Close
Failures at Project Activation	0.63	1				
Subsidiary Workload Between Project Creation and Activation	-0.04	-0.07	1			
Subsidiary Workload Between Project Activation and Close	-0.04	-0.06	0.77	1		
Engineer Experience Level at Activation	-0.01	-0.03	0.33	0.29	1	
Engineer Average Number of Problems Between Project Activation and Close	-0.03	-0.07	0.28	0.38	0.33	1

Appendix 2: Ordinary least squares results for the time from issue creation to project activation. The dependent variable is the natural logarithm of the time between issue creation and project activation.

```
Linear regression                                      Number of obs =    5026
                                                       F( 69,  4952) =       .
                                                       Prob > F      =       .
                                                       R-squared     =  0.3271
                                                       Root MSE      =  1.2314
------------------------------------------------------------------------------
             |               Robust
lncreateac~e |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
HardProblems |   .1910544   .0579018     3.30   0.001     .0775413    .3045676
MediumProbs  |   .2294337   .0446275     5.14   0.000      .141944    .3169235
 teammembers |   .0179227   .0053056     3.38   0.001     .0075212    .0283241
      plants |   .0159814   .0091271     1.75   0.080    -.0019118    .0338747
       parts |  -.0001884     .00121    -0.16   0.876    -.0025605    .0021837
avghrstoflrs |   .0001812   .0000721     2.51   0.012     .0000398    .0003225
   warrbnfts |  -1.29e-08   3.12e-08    -0.41   0.680    -7.41e-08    4.83e-08
  salesbnfts |   1.28e-08   1.56e-08     0.82   0.412    -1.78e-08    4.33e-08
       score |  -.0001507   .0000156    -9.63   0.000    -.0001813     -.00012
subsworkload |   .0537661   .0220251     2.44   0.015     .0105872    .0969451
```

Appendix 3: Gamma regression for the time between project activation and project close. The lead engineer fixed effect is used in addition to the factors in Table 6.

```
Gamma regression -- accelerated failure-time form

No. of subjects =         5028                     Number of obs    =      5028
No. of failures =         3479
Time at risk    =   1345585.15
                                                   LR chi2(730)     =   2714.82
Log likelihood  =   -4914.9896                     Prob > chi2      =    0.0000
------------------------------------------------------------------------------
          _t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
HardProblems |   .2024237   .0433486     4.67   0.000     .1174621    .2873854
MediumProbs  |   .2011322   .0307823     6.53   0.000     .1407999    .2614644
 teammembers |   .0605569   .0050127    12.08   0.000     .0507321    .0703816
      plants |   .0387698   .0081339     4.77   0.000     .0228277    .0547119
       parts |   .0044457   .0010639     4.18   0.000     .0023606    .0065309
avghrstoflrs |    .000295   .0000497     5.93   0.000     .0001975    .0003925
   warrbnfts |  -3.50e-08   2.09e-08    -1.68   0.094    -7.60e-08    5.94e-09
  salesbnfts |  -2.46e-09   1.02e-08    -0.24   0.810    -2.25e-08    1.76e-08
       score |  -.0000298   .0000101    -2.93   0.003    -.0000497   -9.89e-06
 warractivtn |   1.01e-08   1.82e-08     0.56   0.578    -2.55e-08    4.57e-08
 flrsactivtn |  -.0000522   .0000242    -2.16   0.031    -.0000997   -4.78e-06
subsworkload |   .1544906   .0232677     6.64   0.000     .1088868    .2000945
  engrexprnc |  -.0121595   .0021018    -5.79   0.000    -.0162788   -.0080401
engavgproj   |  -.0616101   .0095709    -6.44   0.000    -.0803687   -.0428516
-------------+----------------------------------------------------------------
     /ln_sig |  -.5972462   .0708913    -8.42   0.000    -.7361905   -.4583018
      /kappa |   1.465275   .1822046     8.04   0.000     1.108161     1.82239
-------------+----------------------------------------------------------------
       sigma |    .550325   .0390132                      .4789349    .6323566
```

Appendix 4: OLS results for the natural logarithm of the time between project activation and close for projects activated within the first year of our data set.

```
Linear regression                                      Number of obs =     1487
                                                       F( 48,    335) =        .
                                                       Prob > F      =        .
                                                       R-squared     =   0.2416
                                                       Root MSE      =   .83185
                          (Std. Err. adjusted for 336 clusters in engineerid)
```

lnactivate~e	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
HardProblems	.5635364	.0709938	7.94	0.000	.4238866	.7031863
MediumProbs	.3779318	.0545387	6.93	0.000	.2706503	.4852133
teammembers	.0502905	.0105462	4.77	0.000	.0295454	.0710356
plants	.0616985	.011445	5.39	0.000	.0391855	.0842116
parts	.0053335	.0017389	3.07	0.002	.0019129	.0087541
avghrstoflrs	.0002282	.0000892	2.56	0.011	.0000527	.0004038
pcmpnt_flrs	-.0944841	.0612558	-1.54	0.124	-.2149786	.0260105
warrbnfts	-4.29e-09	1.68e-08	-0.26	0.798	-3.73e-08	2.87e-08
salesbnfts	-9.90e-09	4.78e-08	-0.21	0.836	-1.04e-07	8.40e-08
score	-.0000354	.0000196	-1.81	0.072	-.0000739	3.14e-06
warractivtn	3.00e-08	2.15e-08	1.39	0.165	-1.24e-08	7.23e-08
flrsactivtn	-.0000692	.0000219	-3.16	0.002	-.0001123	-.0000261
subsworkload	.3878942	.0990099	3.92	0.000	.1931347	.5826537
engrexprnc	-.0236071	.0046718	-5.05	0.000	-.0327969	-.0144172
engavgproj	.0165715	.0075363	2.20	0.029	.0017471	.031396

Table 18: Regression results for projects activated in 2004 and closed before June 2007. ln(time between activation and close) is the dependent variable. Robust standard errors are used. Clustering was done at the lead engineer level.

Appendix 5: We regressed the new product introduction spike, new content spike, and the carryover spike on the factors used before, but excluded the late change index from the analysis to see the significance of part new content in the absence of late change index. When they are both in the model, part new content is not significant for the new product introduction spike and the new content spike. It is borderline significant for the carryover spike.

Linear regression

Number of obs	= 125
F(19, 104)	= .
Prob > F	= .
R-squared	= 0.2536
Root MSE	= .38203

NPISpike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
PartNewContnt	1.115525	.4096518	2.72	0.008	.3031698	1.92788
PilotsPerVol	.0033503	.0324166	0.10	0.918	-.060933	.0676336
AvgPrdctnVol	.0035684	.0016901	2.11	0.037	.0002168	.00692
Arrangements	-.0000959	.0090684	-0.01	0.992	-.0180789	.0178871

Table 19: Regression results for the new product introduction spike. The late change index is not included. Robust standard errors. Product type dummy is not reported for brevity.

Linear regression

Number of obs	= 125
F(19, 104)	= .
Prob > F	= .
R-squared	= 0.2928
Root MSE	= .40031

NCSpike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
PartNewContnt	.961577	.4309369	2.23	0.028	.1070129	1.816141
PilotsPerVol	-.0086826	.0272509	-0.32	0.751	-.0627222	.0453571
AvgPrdctnVol	.003013	.0016337	1.84	0.068	-.0002267	.0062528
Arrangements	.0018019	.0095538	0.19	0.851	-.0171437	.0207475

Table 20: Regression results for the new content spike. The late change index is not included. Robust standard errors. Product type dummy is not reported for brevity.

Linear regression

Number of obs	= 125
F(19, 104)	= .
Prob > F	= .
R-squared	= 0.2061
Root MSE	= .77401

CO Spike	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
PartNewContnt	2.222084	.8640844	2.57	0.012	.5085726	3.935596
PilotsPerVol	.0506708	.0440519	1.15	0.253	-.0366857	.1380274
AvgPrdctnVol	.0050461	.0028855	1.75	0.083	-.000676	.0107681
Arrangements	-.0107383	.013085	-0.82	0.414	-.0366863	.0152096

Table 21: Regression results for the carryover spike. The late change index is not included. Robust standard errors. Product type dummy is not reported for brevity.

References

Clark KB, Fujimoto T. 1991. Product Development Performance: Strategy, Organization, and Management in the World Auto Industry. Harvard Business School Press: Boston, MA, USA.

Cleves MA, Gould WW, Gutierrez RG. 2004. An Introduction to Survival Analysis Using Stata. Stata Press: College Station, TX, USA.

Griffin A. 1997. The effect of project and process characteristics on product development cycle time. Journal of Marketing Research, Special Issue on Innovation and New Products, 34(1) 24-35.

Repenning NP. 2001. Understanding fire fighting in new product development. The Journal of Product Innovation Management, 18 285-300.

Sentas P, Angelis L, Stamelos I. 2008. A statistical framework for analyzing the duration of software projects. Empirical Software Engineering, 13 (2) 147-184.

Thomke S. 2003. Experimentation Matters: Unlocking the Potential of New Technologies for Innovation. Harvard Business School Press: Boston, MA, USA.

Wheelwright SC, Clark KB. 1992. Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality. The Free Pres: New York,USA.

Essay 3: Problem Solving Dynamics of Multi-generation Products

Abstract
Although problem identification and resolution in project management and product development have been studied extensively in the system dynamics literature, problems are formulated at the aggregate level. However, heterogeneity is observed across problems, in terms of their importance and complexity, and capturing this heterogeneity is essential to answering important policy questions related to field problem-solving. In order to answer the policy questions using these factors, we built a simulation model in which we represent each problem, defect, and failure as a separate entity. The model was used to test the effectiveness of different prioritization policies, such as prioritizing carryover problems. Simulation results show that prioritizing carryover problems demonstrates considerable potential to improve product reliability. We also tested the impact of different information availability scenarios and problem investigation policies on the reliability of products. We find that these policies do not have much impact on product reliability.

1. Introduction

New product development and the quality of new products are central to corporate success. Unfortunately, in most cases, product development teams fail to identify and solve all product problems before the start of production. Therefore, problem solving activity continues after the start of production, and this is called the field problem solving process. Field problem solving both improves and has a stabilizing impact on the quality of products.

This paper focuses on the field problem solving process and makes two contributions to the existing literature. First, we show that appropriate field problem solving policies can improve the quality of products drastically. Most of the literature on product development focuses on problem solving at the upfront stages of product development (Repenning, 2001; Thomke, 2003). Problem solving in the early stages of product development is obviously very beneficial, but usually it does not solve all problems, and field problem solving is needed to complement the product development problem solving processes. Second, even though problem identification and resolution in project management and product development have been studied extensively in the system dynamics literature, problems are formulated at an aggregate level (Ford and Sterman, 1998; Repenning, 2001). However, heterogeneity is observed across problems, in terms of their numbers of failures, and capturing this heterogeneity is essential to answering important policy questions related to field problem solving.

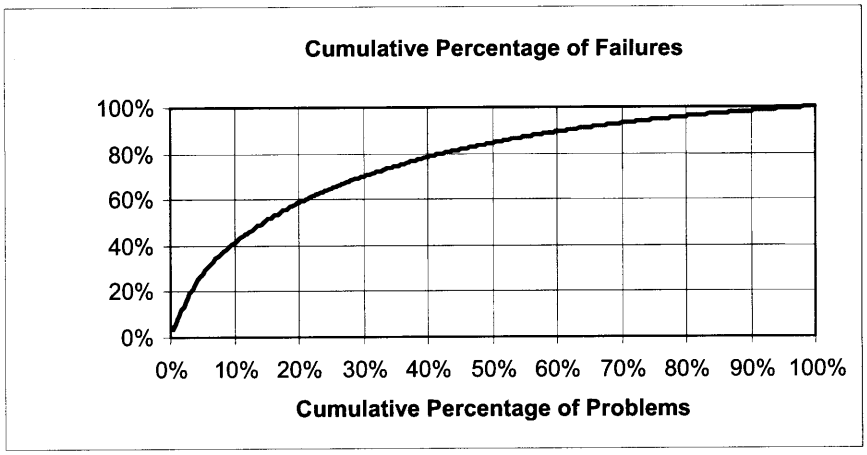


Figure 1: Cumulative percentage of failures vs. percentage of problems.

Data collected at a major motor vehicle manufacturer show that a relatively small fraction of problems lead to the majority of failures. In Figure 1, 20% of the problems account for approximately 60% of the failures. Therefore, firms must use resources efficiently to identify problems and choose the "right" problems to solve. To answer important questions regarding which problem to solve next or which criteria to use to identify problems, decision makers need to consider several factors related to problems. Decisions regarding which problem to solve next depend on various factors, such as the expected number of failures a problem is going to cause each month, the time during which it will remain in production, and whether or not it will be carried over to the next generation. Essay 1 shows that for multi-generation products, carryover problems remain in production for a much longer period than non-carryover problems.

In terms of identifying real problems among potential problems, an important issue is deciding when an engineer should investigate a problem. Investigating every problem after the first failure might result in the waste of resources for many "false positives." On the other hand, waiting too long to investigate an important problem might lead to the loss of valuable time. Therefore, setting an appropriate threshold for the number of failures for identifying problems is an important policy issue.

Another issue related to problem identification is determining the frequency with which to update expectations about the number of failures to which the problem will lead. Overly frequent updates may waste engineers' time without substantial benefits, while too infrequent updates lead to decisions made on less relevant information.

The factors mentioned above - such as the expected number of failures a problem may cause each month, the time during which it will remain in production, whether it will be carried over or not, the number of failures to date, and the operating hours until failure - vary widely across problems and also change over time. In order to answer the policy questions using these factors, we built a simulation model in which we represent each problem, defect, and failure as a separate entity. A problem is defined as a flaw in the manufacturing process or in the design of a group of parts that leads to defects on multiple vehicles. A defect is a part or group of parts in a single vehicle that will eventually result in a failure, but has not yet resulted in a failure. A defect results in a failure after the vehicle is used by the customer for some period of time. The model keeps track of all failures for each problem and uses this failure information to estimate the number of expected failures that will be generated by each problem, if it remains unsolved. In the following sections, we will describe the model and report simulation results. Simulation runs include analysis of different prioritization, estimation, and investigation policies.

2. Model

Our model simulates the field problem solving process on multi-generation products (Figure 2). Unsolved problems lead to defects, as vehicles are produced based on the Weibull distribution of operating hours until failure. Defects lead to failures after a time delay, following which failure information is reported to engineers. Engineers use this information to prioritize and solve problems. As the next generation of the product is introduced to the market, some problems become obsolete, some are carried over to the next generation, and some new problems are introduced. The fraction of problems carried over to the next generation is a function of the fraction of newly designed parts and the number of carryover problems solved before the introduction of the next generation.

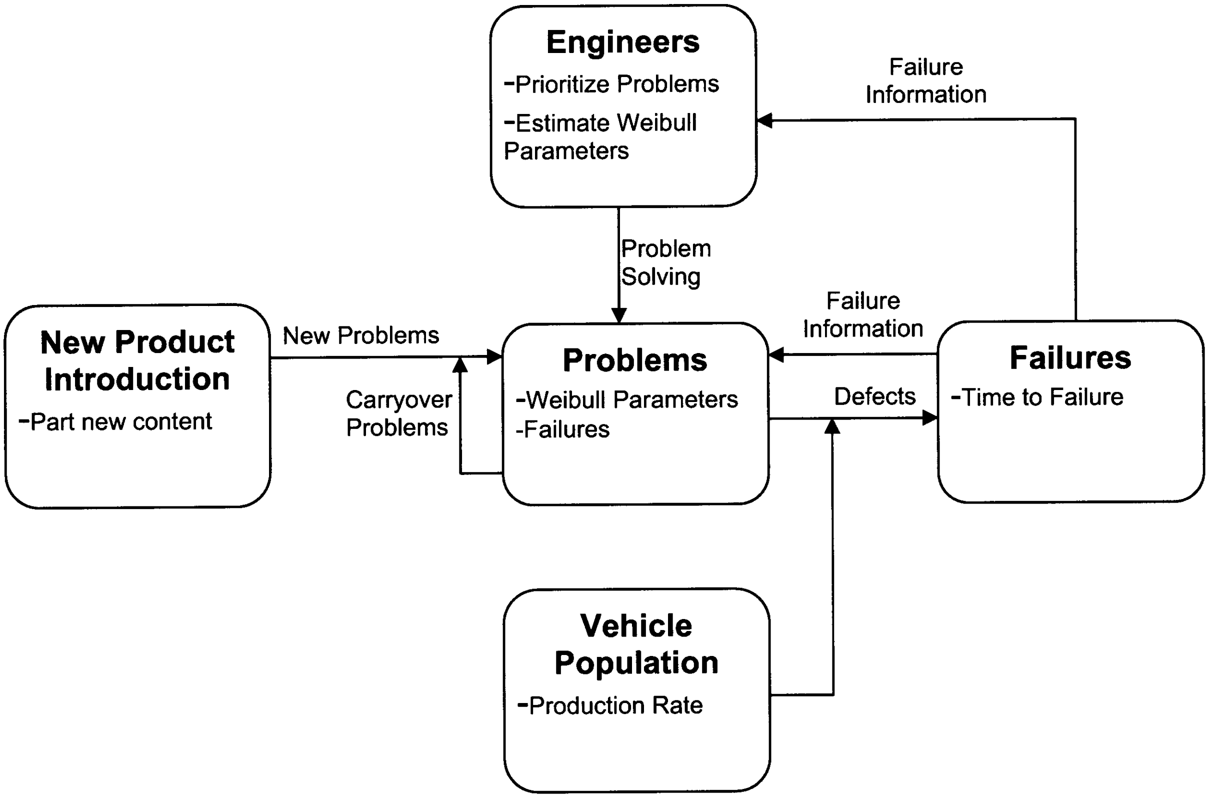


Figure 2: Overview of the simulation model.

Figure 3 shows the reliability dynamics of a typical product at our research site. After the introduction of the previous generation product, problem solving teams identify problems and improve product reliability by solving those problems. When the next generation product is introduced, reliability deteriorates initially, but then improves over time through the work of problem solving teams.

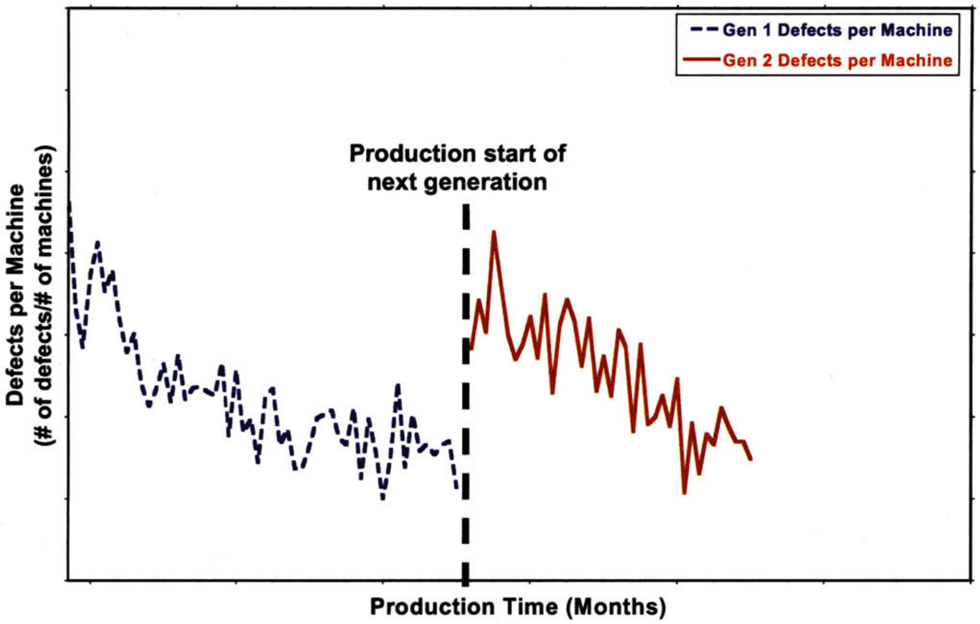


Figure 3: Defects per vehicle versus production date of products.

The problem solving process is shown in Figure 4. When a new problem is created, it has not yet failed. As failures accumulate, an issue is created. These issues are investigated, and if the engineers decide that the issue is worth being solved, the issue becomes a project. Newly created projects initially enter the queue of pending projects and wait for resources. Resources are assigned to problems chosen from the pending project list, and these projects are activated. If the engineers conclude that they found a fix to the problem and implemented it, the project is closed. When the product transitions to the next generation, non-carryover problems become obsolete and do not lead to further defects. Projects might be cancelled during investigation, while they are pending problems, or after problem activation.

The following subsections describe important formulations in the model.

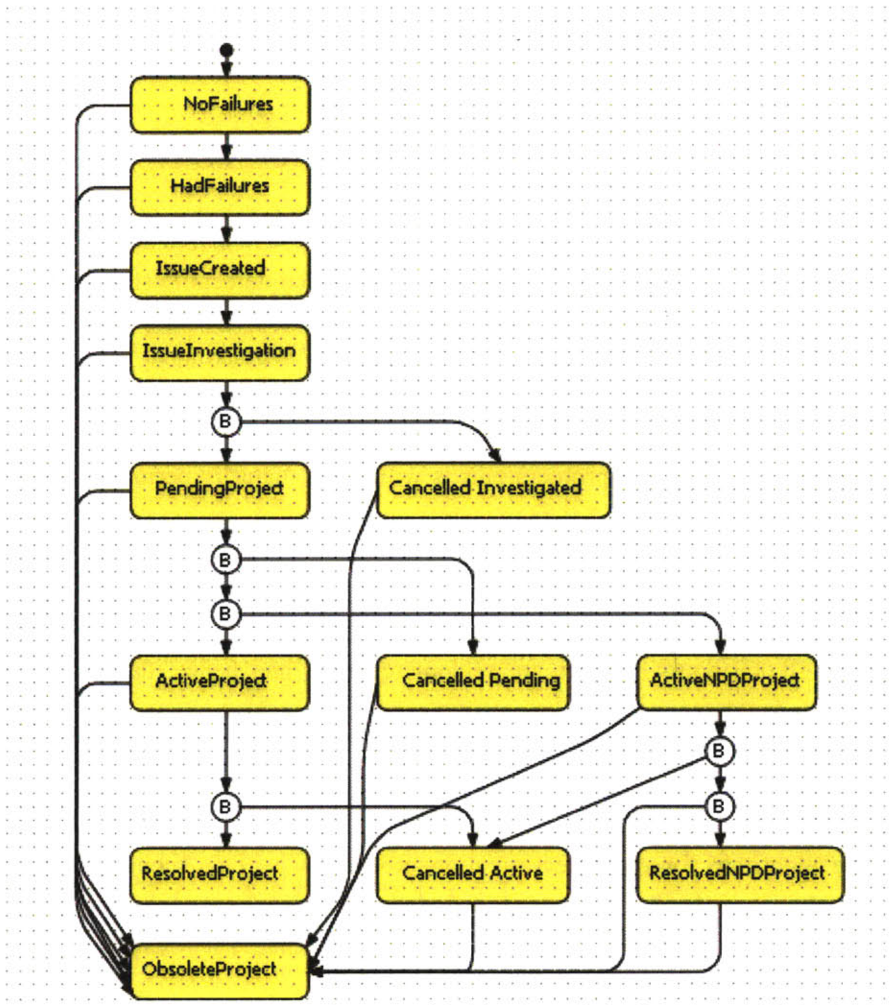


Figure 4: Statechart for problems

2.1 Defect Creation

A new problem starts in the "NoFailures" state and leads to defects, until it is resolved or becomes obsolete. Defect creation and the occurrence of failures are modeled by assuming that the operating hours until failure follow the Weibull distribution. The Weibull distribution is widely used in reliability engineering to model hours to failure due to its flexible shape. The probability density function of the Weibull distribution is:

$$f(x;k,\lambda)=\frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1}e^{-(x/\lambda)^k}$$

for $x > 0$ and $f(x; k, \lambda) = 0$ for $x \leq 0$, where $k > 0$ is the slope parameter and $\lambda > 0$ is the characteristic value.

We used the Weibull distribution of each problem to introduce defects to some vehicles according to the following algorithm: for each vehicle produced in a given month, we drew a random number for each problem from the problem's estimated Weibull distribution of operating hours to failure. If the operating hours to failure drawn for a vehicle-problem pair is within the maturity period of the product, then we created a defect on that vehicle associated with that problem. The first 10,000 hours of operation is called the maturity period of a vehicle. This is the period in which the majority of reported failures, more than 80%, occur. After 10,000 hours, failures are seldom reported, and hence, data are sparse. The defect causes a failure when the vehicle reaches the operating hours to failure. To increase the computational efficiency of the model, we do not keep track of vehicles in the model. Instead, the model tracks defects and failures. The algorithm for the defect creation process in a given month can be summarized as follows:[1]

```
For i=1 To "Vehicles produced this month"
  For j=1 To "Number of Problems"
    Operating hours to Failure(i,j)= Random number from Weibull(Slope(j),Characteristic
    Value(j))
    If
      Operating hours to Failure (i,j)<Maturity period of product
    Then
      Create a defect that will fail after "Operating hours to Failure(i,j)"
```

[1] This is the algorithm used for the first generation product of a problem. For subsequent generations, the "vehicles produced this month" variable in the first for loop is replaced with another value, which will be explained later.

Else

Do nothing

Each problem has different parameter values for the Weibull distribution characterizing that particular problem. The parameters of the Weibull distribution used in the model for each problem are estimated using data from our research partner. We used the maximum likelihood method for estimation. If all vehicles have failures associated with a problem, estimation is performed using failure data for all vehicles, including the operating hours to failure for each vehicle. However, in many vehicles, a failure is not observed. In those cases, information indicating that the vehicle has been in use to date and has not failed is useful. These cases are called suspensions, in reliability engineering terminology. The maximum likelihood estimation exploits both failure and suspension information to estimate the distribution of operating hours to failure.

The likelihood function for cases that include suspensions is:

$$L = \prod_{i=1}^{n} f(x_i;k,\lambda) \prod_{j=1}^{p} \left[1 - F(y_j;k,\lambda)\right]$$

where n is the number of failures, x_i is the operating hours to the i^{th} failure, p is the number of suspended data points, y_j is the operating hours to date of the j^{th} suspension, and $F(y_j;k,\lambda)$ is the cumulative distribution function.

When a problem leads to a defect, a defect agent is created. There are three time delays before a defect fails in the field and is reported: the sales delay, operating hours to failure, and the reporting delay. Calendar months to failure for a defect is calculated as follows:

$$\text{Sales Delay} + \frac{\text{Hours to Failure}}{\text{Operating Hours per Month of the Vehicle}} + \text{Reporting Delay}$$

The maximum likelihood analysis on data from our research partner showed that the Weibull distribution is the best fitting distribution for sales delay and operating hours per month of the vehicle (Appendix). Reporting of failures is done on a periodic basis by updating databases, so a failure that occurs one day before the update has a very short reporting delay, whereas a failure that occurs one day after the update must wait for the full reporting cycle. Therefore, we assumed that reporting failure is distributed uniformly, with a minimum value of 0 and a maximum value that equals the length of the reporting cycle. When a defect leads to a failure, a failure agent is created.

2.2 Issue Investigation

When the first failure occurs, a problem moves to the "HadFailures" state in the problem statechart (Figure 4). After the third failure, an "issue" is created and the problem moves to the "IssueCreated" state. Issues might be important problems, but the importance of an issue needs to be investigated to ensure that it is a problem worth solving. Depending on the investigation results, the issue either becomes a pending project and transitions to the PendingProject state, or it is cancelled because the engineer concludes that it is not a real or sufficiently important problem and moves it to the Cancelled Investigated state. Data analysis shows that the exponential distribution is the best-fitting distribution for investigation delays of issues (Appendix 1).

As mentioned before, pending issues wait in a queue for resources. When an issue becomes a pending problem, its Weibull parameters are estimated in the model.

2.3 Problem Prioritization and Project Activation

When an issue becomes a pending project, it is activated immediately if there is an available engineer to work on the problem, and the problem moves to the Active Project state. If all engineers are busy working on other problems, the problem is added to the queue of pending problems. An engineer picks a new problem from the pending problems queue and starts working on it, as soon as the engineer finishes working on a problem. Active projects are chosen from the pending problem list according to the prioritization policy used. Prioritization of pending problems is an important policy question, and we analyze different prioritization policies using the model. The following subsections explain the base prioritization policy and our proposed prioritization policies in more detail.

2.3.1 Base Prioritization Policy

In the base prioritization policy, problems are ranked according to the regression results in essay 2. In essay 2, we estimate the time between issue creation and project activation using survival analysis.

Survival analysis results show that such factors as problem complexity, score, subsidiary workload, number of team members, number of plants involved in the problem, and average operating hours before seeing failures significantly influence the time to activate problems. Even though it is a mix of several factors, essay 2 shows that the overall tendency is to choose problems that will be solved in shorter periods of time. Survival analysis results also yield the hazard function for problems. The hazard function is the probability that the project is activated at time t, given that it was not activated until time t. Therefore, at any given time, we can compute the hazard function of each problem, given the specification of the regression model in essay 2, and significant factors that influence the hazard. We use these values to rank problems in the base case scenario. When an engineer becomes available, she chooses the problem with the highest rank.

2.3.2 Using Carryover Information for Prioritization

Note that the base policy does not consider the carryover benefits of solving a problem. In policy 1, we take the carryover benefits of solving a problem into account to prioritize problems. Problems are ranked according to the expected failures that will be prevented by solving the problem, including the carryover benefits in Policy 1.

Expected failures that will be prevented by solving the problem is equal to the expected number of failures that will be created per month, if the problem is not solved, multiplied with the benefit period of solving the problem.

ExpFailures = ExpFailuresPerMonth * BenefitPeriod (1)

ExpFailuresPerMonth: Expected monthly number of failures that will be created if the problem is not solved.

BenefitPeriod: Period in which solving the problem will have an impact

The number of expected failures per month equals the defect rate multiplied by the expected fraction of vehicles that will fail unless the problem is solved. However, there cannot be more failures than vehicles, so the number of failures per month is capped at production rate. The defect rate will be explained later in the discussion of the carryover spike.

ExpFailuresPerMonth=Min(ProdRate,DefectRate*ExpFractionOfFailure)

The expected fraction of vehicles that will fail is equal to the probability of having a failure on a vehicle. This is equal to the probability of having an operating hours to failure value smaller than the maturity period. This probability is the Weibull cumulative distribution function value at the maturity period. Note that in this section, we use the actual values of the parameters of the Weibull distribution. We will explain the case in which estimated Weibull parameters are used in Section 2.3.4.

$$ExpFractionOfFailure = 1 - e^{-\left(\frac{MaturityPeriod}{ActualCV}\right)^{ActualSlope}}$$

The benefit period is the time during which defective products will be built if the problem is not solved. For carryover problems, this period covers the time until the end of production for the next generation, which equals the time until the end of this generation, plus the entire production cycle time of the next generation. However, problems are not solved instantaneously, so we subtract the median time to solve problems.

BenefitPeriod=Max(0,TimeBetweenTwoGenerations-NextTransitionTime-CurrentTime-MedianProjectCompletionTime)

During the product development project, engineers learn which problems will be carried over to the next generation only after the release of the bill of materials for the next generation product. Therefore, in the model, engineers can only use the carryover information for prioritization after the release of the bill of materials and before the introduction of the next generation product. Before the release of the bill of materials, they do not know which problems will be carried over, and the benefit period does not include TimeBetweenTwoGenerations for carryover problems. Obviously, the BenefitPeriod for non-carryover problems does not include the time between two generations in any situation.

NextTransitionTime: Introduction date of next generation product

CurrentTime: Current date

MedianProjectCompletionTime: Median time it takes to solve a problem

Note that the time to solve a problem is different for each problem. We assume that this follows the Weibull distribution, since Weibull was the best-fitting distribution according to our data analysis results (Appendix).

We will test the impact of implementing this policy with our simulation model. By using carryover information and considering the longer benefit period of carryover problems, this policy might prefer a carryover problem with fewer expected failures per month, rather than a non-carryover problem with more expected failures per month. However, it is not certain that this policy will lead to fewer defects, due to the dynamic and stochastic nature of the problem solving process. Suppose only two problems exist - Problem A and Problem B. Engineers can solve only two problems before the end of production for this generation, but only one problem can be solved at a given time. Problem A leads to twice the number of failures per month as Problem B does. However, Problem B is a carryover problem, and its benefit period is three times that of Problem A. Solving both problems will take the same amount of time. If engineers can be certain that another problem will not emerge, solving Problem A first, even though it is not a carryover problem, is the better choice, since it will prevent more problems due to its higher failure rate per month. Since there are sufficient resources to solve Problem B before the end of production for this generation as well, they will also be able to reap the long term benefits of solving Problem B, which is a carryover problem. However, unlike this example, real world problems cannot be predicted and may emerge unexpectedly. While engineers may start solving Problem A or Problem B, other problems that will dominate both Problem A and Problem B might emerge directly before the chosen problem is solved, and engineers will not be able to solve the problem that is not initially chosen from the set of Problems A and B. In that case, solving Problem B instead of Problem A is more beneficial, since this offers greater overall benefits compared to Problem A. As demonstrated by this example, the policy of using carryover information for prioritization is not necessarily beneficial in all cases. Therefore, we will use the simulation model to analyze whether this policy will lead to fewer defects.

2.3.3 Solving Carryover Problems for Only the Next Generation

In this policy, engineers are given the option to solve problems for the next generation only. If they prefer to do so, fixes for the problems are designed and implemented for only the next generation, not for the current generation. The advantage of this policy stems from the decreased effort in designing and implementing the fix. Since the fix does not need to take into account the design of the current generation, engineers will have a greater degree of freedom in designing the fix. A change in the current generation requires designing the fix so that it functions well with the interacting parts of the current generation, which is challenging, since the current generation is already in production. This might even lead to a redesign of the interacting parts of the current generation or changes in manufacturing processes.

Designing the fix for only the following generation will be easier, especially if the next generation's design is not complete, since engineers will have fewer constraints. Based on interviews with expert reliability engineers, we assumed that the policy of fixing carryover problems for only the next generation would lead to a 25% reduction in the time required to solve problems. Experts told us that approximately half of the time is spent on problem identification and root cause analysis, which would be the same in both cases. However, they estimated a 50% reduction in the actual problem solving activity, resulting in a 25% reduction in the time necessary to solve the problem. The obvious disadvantage of this policy is that it will not improve the reliability of the current generation and hence the benefit period will be shorter.

The benefit period when a problem is solved for only the next generation is:

BenefitPeriod=Max(0,TimeBetweenTwoGenerations+min(0,NextTransitionTime-CurrentTime-MedianTimeToSolveProblemsForOnlyNextGeneration))

If the MedianTimeToSolveProblemsForOnlyNextGeneration is longer than the time remaining until the next generation, the benefit period will be less than the TimeBetweenTwoGenerations. However, if the MedianTimeToSolveProblemsForOnlyNextGeneration is less than the time remaining until the next transition, the benefit period equals the time between the two generations.

If the option of solving problems for only the next generation is allowed, this option is compared to solving the problem for both the current generation and next generations in a longer period of time. If the ratio of "expected failures" to "expected time to solve" of the next generation solution only is greater than that of solving the problem for both generations:

$$\frac{ExpFailures(ForNextGenerationOnly)}{ExpTimeToSolve(ForNextGenerationOnly)} \geq \frac{ExpFailures(ForBothGenerations)}{ExpTimeToSolve(ForBothGenerations)}$$

then engineers will prefer solving this problem for only the next generation; the corresponding benefit to the time to solve ratio will be used to run the problem at hand against other problems. If this ratio is ranked highest among all pending problems, then the problem will be solved for only the next generation product. In that case, the problem will transition from the PendingProject state to the ActiveNPDProject state in Figure 4.

2.3.4 Estimation of Weibull Parameters

In this section, we describe the procedure used in the model to estimate the parameters of the Weibull distribution for hours to failure of each problem. The maximum likelihood and ranked regression are two popular methods used for estimating parameters of a Weibull distribution. Both of these methods are used by our research partner. The maximum likelihood is useful for problems with many suspensions, but it is not recommended for samples with a small number of failures, and it is computationally more time-consuming than ranked regression. Due to the relatively small number of failures of problems and the computational intensity of the maximum likelihood method, we used the ranked regression method in the model.

In this model, we estimate the Weibull parameters of all pending problems when they become a pending project, then update those estimates every simulated month. We also report simulation results, in which we compare monthly update results to the no update policy, as well as the case in which the true parameter values are used for analysis.

The next subsection describes the ranked regression procedure used in the model.

2.3.4.1 Ranked Regression for Estimating Weibull Distribution Parameters of Problems

This section explains how we implement the ranked regression estimation for Weibull parameters of problems in the simulation model. Let us assume that we have f failures out of N operating vehicles in total. The ranked regression method uses the "ranks" of failed vehicles' hours to failure within the entire vehicle population. We first rank all data points according to the total hours of operation - in other words, we use "hours until failure" for failed vehicles and "total operating hours to date" for vehicles that did not fail, namely suspensions.

This method requires that the simulation model keep track of the ranks of all produced vehicles to compute the ranks of suspensions. Creating a separate agent for each produced vehicle, regardless of whether the vehicle fails, considerably slows down the simulations. Therefore, we used an approximate method to run the model without creating agents for vehicles that did not fail. In this method, we approximate the ranks of failed vehicles by using the distribution of operating hours per month for all vehicles and the distribution of the sales delay. F() is the cumulative distribution function of operating hours per month. Suppose that a defect on a vehicle called FailedVehicle failed at FailureTime hours of operation at month FailureMonth. Our goal is to find the overall rank of FailedVehicle in terms of hours to failure out of all vehicles in the sample. We know the HoursToFailure on all failed vehicles, and we know the ranking of FailedVehicle among other vehicles with failures. We use the approximation for finding its ranking compared to vehicles without failures. Assume that production started in month 0 and that the production rate is constant and equal to ProductionRate. At month FailureMonth, non-failed vehicles produced at month 0, whose operating hours per month are less than FailureTime/(FailureMonth-Median Sales Delay), have a lower rank than FailedVehicle. Similarly, vehicles produced at month 1, whose operating hours per month are less than FailureTime/(FailureMonth-1-Median Sales Delay), have a lower rank than FailedVehicle, and so on. Since we know the distribution of operating hours per month, we find the overall rank of FailedVehicle, m_j, by running the following loop:

m_j =1+ Number of Failures with Hours To Failure less than FailedVehicle's Hour To Failure

For(i = 0;i < FailureMonth;i++)

{

$m_j = m_j$ + ProductionRate * (1-f/N)*F(FailureTime / (FailureMonth – i -MedianSalesDelay))

}

We multiply ProductionRate by (1-f/N) to obtain the number of vehicles without failures.

Since suspensions have accumulated operating hours but have not failed, there is a chance that they will fail in the future and will change the ranking of failed vehicles. Therefore, we need to adjust the rankings for failed vehicles, by taking into account the possibility of suspensions to fail. The following equations are used to adjust the rankings of failed vehicles (Mathpages, n.d.):

$r(0) = 0$

$$r(j) = r(j-1) + \frac{N+1-r(j-1)}{N+1-(m_j-1)}$$

$j = 1,2,3,\ldots,f$

where:

$r(j)$: adjusted rank of j^{th} failure

N: total number of vehicles (both failed and suspensions)

f: total number of failed vehicles

j: ranks of failed vehicles, in terms of hours to failure within failed vehicles

m_j: overall ranks of failed vehicles, compared to hours to failure for failed vehicles and operating hours to date for suspensions.

Once we adjust the ranks of failed vehicles, we run a linear regression to estimate the two parameters of the Weibull distribution, namely the slope parameter (k) and characteristic value (λ), using data from all failures. The following equations are used to estimate the Weibull parameters.

$$k = \frac{f\sum_{j=1}^{f} y_j v_j - \left(\sum_{j=1}^{f} y_j\right)\left(\sum_{j=1}^{f} v_j\right)}{f\sum_{j=1}^{f} y_j^2 - \left(\sum_{j=1}^{f} y_j\right)^2}$$

$$\lambda = \exp\left(\frac{\left(\sum_{j=1}^{f} v_j\right)\left(\sum_{j=1}^{f} y_j^2\right) - \left(\sum_{j=1}^{f} y_j\right)\left(\sum_{j=1}^{f} y_j v_j\right)}{-k\left(f\sum_{j=1}^{f} y_j^2 - \left(\sum_{j=1}^{f} y_j\right)^2\right)}\right)$$

where:

$y_j = \ln(x_j)$

x_j = operating hours to j^{th} failure

$$v_j = \ln\left(\ln\left(\frac{1}{1-\frac{r(j)-0.3}{N+0.4}}\right)\right)$$

2.4 Problem Solving

When a pending problem is assigned to an engineer, it moves from the "PendingProject" state in Figure 4 to the "ActiveProject" state, if it is to be solved for both the current and the next generation. If the problem is to be solved for only the next generation, it moves to the "ActiveNPDProject" state.

We use regression results from essay 2 to assign the project completion time of each problem. Essay 2 shows that factors related to problem complexity and importance influence the project completion time. However, as in most regressions, these factors do not explain the entire variation in project completion times. Hence, we use the values of significant factors for each problem in combination with the regression coefficients to determine the project completion time of each problem, but we also add randomness to the project completion time, based on our data analysis.

It takes less time to solve a problem in the "ActiveNPDProject" state than in the "ActiveProject" state. The ratio of time to solve the problem for both generations to solving it for the next generation only is as follows:

$$\text{ratio\_FixBothGens\_NPDFix} = \frac{\Pr ojectCompletionTimeBothGens}{\Pr ojectCompletionTimeNPDFix}$$

When a project is solved for both generations and moves to the "ResolvedProject" state, it immediately stops creating defects. On the other hand, if a problem moves to the "ActiveNPDProject" state in a product generation and the problem is solved before the end of that product generation, it moves to the "ResolvedNPDProject" state and continues to create defects until the end of that generation. This occurs because the fix is effective for the next generation product. At the end of that generation, the problem moves to the "ObsoleteProject" state and no longer leads to defects. However, if the engineer started working on the problem during the production of generation x and solved the problem after the production of generation x+1 commenced, the problem moves to the "ObsoleteProject" state, and the fix becomes effective immediately. In that case, the problem stops leading to defects, since the engineer has solved the problem for generation x+1.

2.5 New Generation Introduction

When a new generation is introduced, some problems are carried over to the new generation, some become obsolete, and some new problems are created. The number of problems carried over depends on the fraction of new parts in the next generation, and the number of carryover problems solved before the introduction of the next generation. As generation x is introduced, we label some problems to be carried over to generation x+1, unless they are solved before the introduction of generation x+1. If generation x+1 involved no new parts, all unsolved problems at the end of generation x would be carried over. If all parts of generation x+1 were new, all of its problems would be new. Therefore, we assumed that the probability of a problem to be carried over to the next generation is a function of the fraction of carryover parts.

P(problem(i) will be carried over to the next generation) = Fraction of Carryover Parts = 1 - Fraction of New Parts

A carryover problem that is not solved until the end of generation x is carried over to generation x+1, when generation x+1 is introduced.

2.5.1 Carryover Spike

Empirical results shown in Essay 2 demonstrate that carryover problems result in more failures in next generation products; this phenomenon is called the carryover spike. We modified the defect creation algorithm to capture this finding. The number of random variables drawn from the Weibull distribution for defect creation in Section 2.1 was changed to *defect rate,* instead of *vehicles produced per month*. The defect rate is equal to the production rate when a new problem is introduced. However, if that problem is carried over to next generation, the defect rate of the problem becomes:

defectrate=carry_over_spike*defectrate

Note that the defect creation algorithm caps the number of failures at the number of vehicles produced each month.

Data analysis shows that the carryover spike is primarily caused by manufacturing and assembly related issues, and these problems are solved by plant employees outside of the scope of field problem solving. On average, the carryover spike's impact vanishes in three years. Therefore, we reduce the impact of the carryover spike linearly over time and reduce its impact to zero in three years.

The carryover spike is a function of the fraction of new parts. If there are no new parts, the carryover spike would equal 1, corresponding to no change in the failure rate of carryover parts. As new content increases, however, the carryover spike also increases. The sensitivity of the carryover spike to new content is found in essay 2.

carry_over_spike=1+sensitivity_of_carryover_spike_to_newcontent*newcontent

2.5.2 New Problems

In addition to carryover problems, a new generation also encounters new problems. We assumed that the number of new problems introduced increases linearly, as the fraction of new parts increases. The fraction of carryover parts and new parts add up to 1, so as the fraction of new parts increases, the fraction of carryover parts decreases.

New Problems = Fraction of New Parts * Problems of a Completely New Product

When a new problem is introduced in the simulation model, a problem is randomly chosen from the actual data set of problems and its Weibull parameters are assigned to the new problem created in the simulation model.

2.5.3 Problems Becoming Obsolete

When the next generation product is introduced, non-carryover problems become obsolete. An obsolete problem stops creating defects immediately.

2.5.4 Using Previous Generation's Failure Data to Predict the Importance of Problems Carried over from the Previous Generation

As shown in essay 1, a new generation product contains carryover parts with problems that were not solved before the introduction of the new generation. In addition, failures may occur on carryover parts of a previous generation vehicle after the introduction of the new generation. Typically, these are the problems resulting in failures after extremely long time delays. In this section, we will describe a policy that aims to use failure information from the previous generation vehicles to estimate the future of the current generation. Since these problems cause failures after a substantial time delay, it will be too late if engineers wait until failures are observed in the new generation. This policy aims to solve these problems as soon as possible, even if they have not yet caused failures in the current generation.

Under this policy, when a next generation product is introduced, all problem agents that are carried over continue as open problems; defect and failure agents are kept in memory if this policy is implemented. Failures of both the current generation and the preceding generations are used to estimate the Weibull parameters of the problem. This method increases the accuracy of Weibull parameter estimates, especially during the early phases of the next generation product, when there are few next generation failures upon which to run the Weibull analysis. If the policy is not implemented, a new problem that has identical parameters to the previous problem, except the defect rate and carryover, is created. The defects and failures of the previous problem are not transferred to the new problem.

Data from our research site reveal that this policy increases the amount of available information immensely. Figure 5 shows failure data for identical parts in two successive generations. The blue curve shows the information available from the previous generation. The pink curve shows the information available from the new generation. The red curve is the combined failure information from the two generations. The combined information is much richer and provides much more abundant data, compared to second-generation information only.

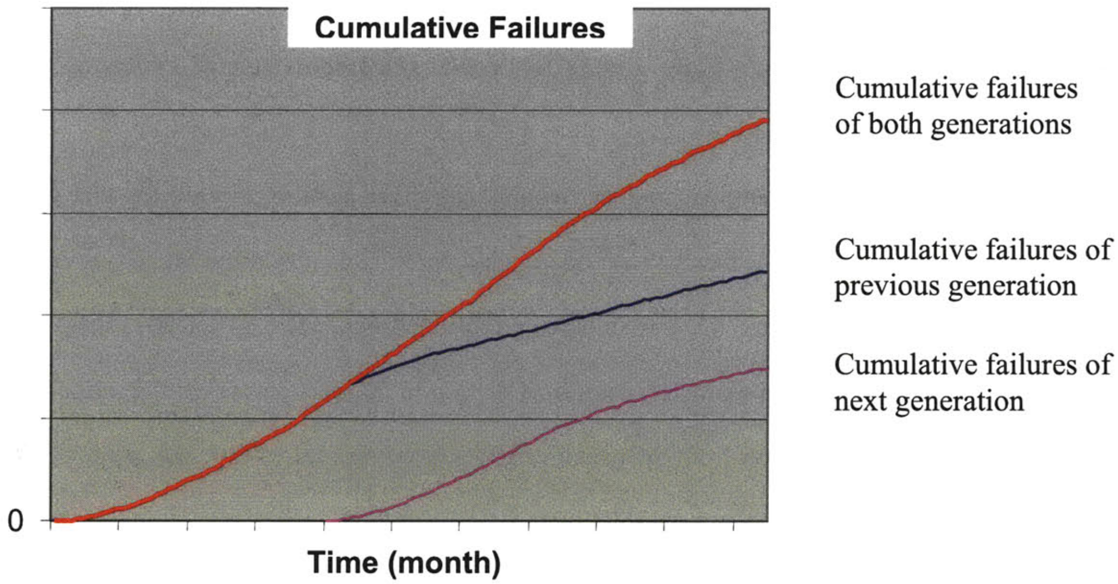


Figure 5: Cumulative failures of two successive generations of a product. The blue curve represents the previous generation, and the pink curve represents the next generation. The red curve represents the total failures of the two generations.

Not considering the previous generation's information regarding carryover parts corresponds to using only the failure data, indicated by the pink curve, to estimate the Weibull parameters of problems. On the other hand, using information from both the previous generation and the new generation corresponds to using the failure data, as shown by the red curve. Figure 5 shows that the difference in information availability is dramatic.

3. Simulation Results

We ran the simulation model using parameter values obtained from a representative product for 250 months. A new generation is introduced once every 50 months. Note that problems are not cancelled in our simulation runs, since cancellations do not influence our policy questions. We first present the results of different prioritization scenarios, in combination with the policy of using the previous generation's failure data to predict the importance of problems carried over from the previous generation. Then, we analyze the impact of having perfect information of Weibull parameters versus the boundedly rational case of estimating Weibull parameters, using failure data (Morecroft, 1983). Finally, we analyze different investigation policies, in which we test the impact of using different threshold levels for the number of failures to start investigating a problem.

3.1 Prioritization Policy Analysis

In the previous sections, we described different prioritization policies, such as using carryover information for prioritization and having the option to solve problems for the next generation only. We first analyze the case of not implementing either of these two policies. Then, we implement the policy of using carryover information for prioritization. If carryover information is not used for prioritization, it is not possible to have the option of solving the problem for only the next generation. Hence, we do not implement the scenario of solving the problem for only the next generation by itself. We implement this together with the policy of using carryover information for prioritization. Then, we compare these policies to the case of choosing the problem with the maximum number of problems to date. Note that this policy neither uses the Weibull estimation nor considers the future benefits of solving a problem. Next, we analyze the case in which problems are chosen randomly. We also analyze these policies in combination with the policy of using the previous generation's failure data to predict the importance of problems carried over from the previous generation. Except the case of choosing problems randomly or choosing the problem with the highest number of failures, Weibull parameters are estimated using ranked regression, and estimates are updated every month. The investigation of problems starts after the third failure.

In the base case, none of the policies are implemented, and the number of total defects in this scenario is the benchmark. Figures 6 and 7 show the numbers of defects and failures, respectively. After the introduction of a new generation, problem solving efforts start to reduce the number of defects. Then, a new generation is introduced, and reliability deteriorates, due to the introduction of new problems and the carryover spike. The reason that the initial number of defects is higher for the second generation, compared to the first generation, lies in the carryover spike. After the second generation, the system is in equilibrium, which is consistent with our data analysis.

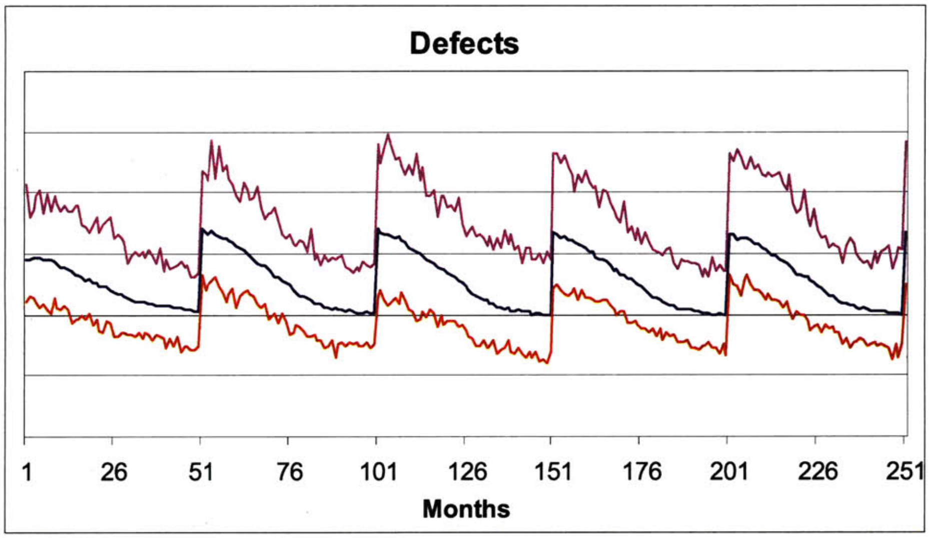


Figure 6: Number of defects created. None of the prioritization policies are implemented. Graph shows maximum, minimum, and average values of simulation runs.

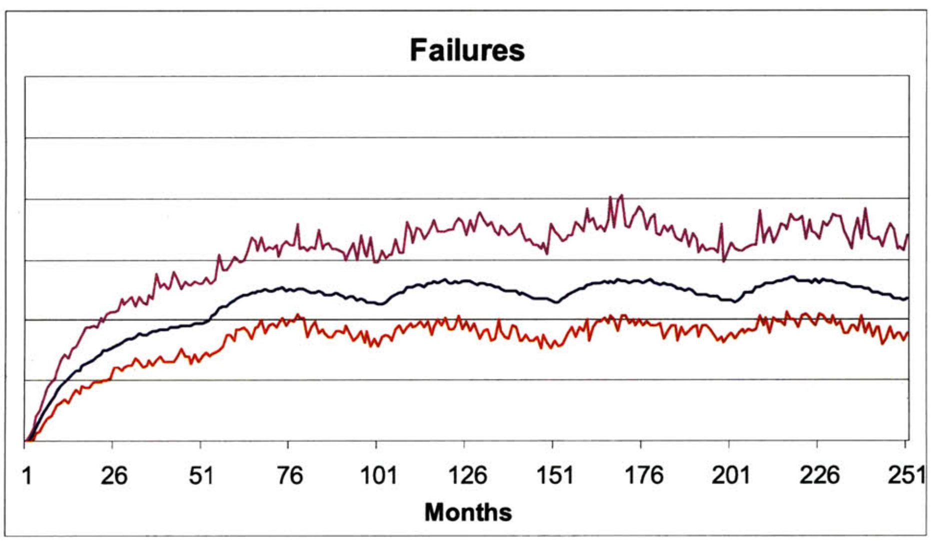


Figure 7: Number of failures. None of the prioritization policies are implemented. Graph shows maximum, minimum, and average values of simulation runs.

Simulation results of all scenarios are presented in Table 1. Using carryover information for prioritization reduces the number of defects by 12%. When combined with the option to solve problems for the next generation only, the total reduction in defects increases to 17%. Interestingly, using carryover information from the previous generation increases the number of defects by 5%. However, when this policy is combined with the policies of using carryover information for prioritization and having the option to solve problems for the next generation only, the combined benefit of the three policies is a 25% reduction in defects, showing that there are synergies. This is a drastic improvement in quality obtained without increasing resources.

Table 2 shows that the base case scenario reduces the number of defects by only 11% compared to choosing problems randomly. The policy of choosing the problem with the maximum number of failures leads to a 14% improvement, with respect to the base case. This is a smaller improvement when compared to the 25% reduction in the case of implementing our proposed policies.

Use Carryover Information For Prioritization	Have the Option to Solve for Next Generation Only	Use Carry Over Information From Previous Generation	Weibull Parameter Estimation	Total Defects in 250 Months
No	No	No	Ranked Regression Monthly Update	100%
Yes	No	No	Ranked Regression Monthly Update	88%
No	No	Yes	Ranked Regression Monthly Update	105%
Yes	Yes	No	Ranked Regression Monthly Update	83%
Yes	No	Yes	Ranked Regression Monthly Update	78%
Yes	Yes	Yes	Ranked Regression Monthly Update	75%

Table 1: Total defects created in 250 simulated months by different prioritization scenarios. All policies are significantly different than the base case scenario, with significance levels of less than 10^{-4}.

Use Carryover Information For Prioritization	Have the Option to Solve for Next Generation Only	Use Carry Over Information From Previous Generation	Other Prioritization Policy	Weibull Parameter Estimation	Total Defects in 250 Months
No	No	No	No	Ranked Regression Monthly Update	100%
No	No	No	Choose Problems Randomly	N/A	111%
No	No	No	Choose the Problem With Maximum Failures	N/A	86%
No	No	Yes	Choose the Problem With Maximum Failures	N/A	91%
Yes	Yes	Yes	No	Ranked Regression Monthly Update	75%

Table 2: Total defects created in 250 simulated months by different prioritization scenarios. All policies are significantly different than the base case, with significance levels of less than 10^{-17}.

These results show that combining the policy of using carryover information from the previous generation with the policies of using carryover information for prioritization, in addition to having the option to solve problems for the next generation only, leads to substantial benefits. The difference between this scenario and the base case is shown in Figure 8.

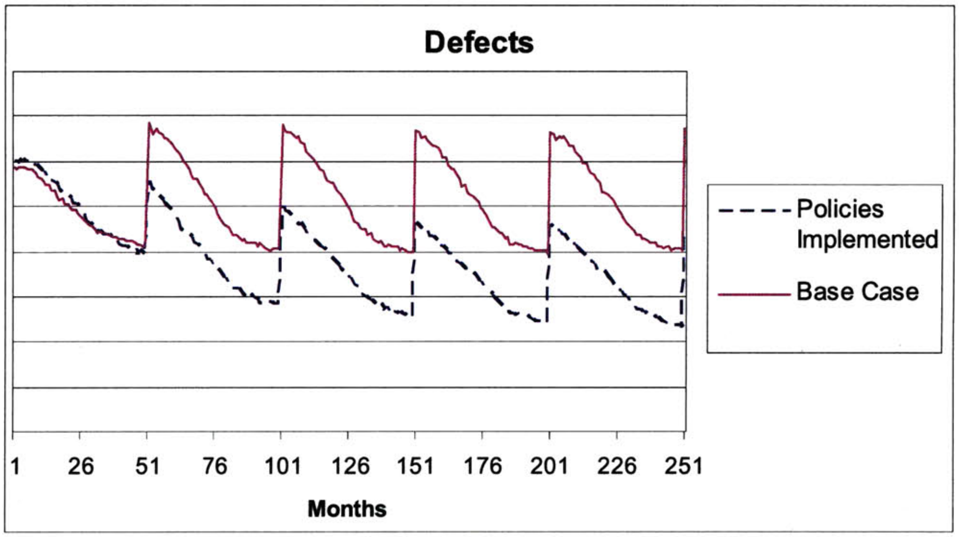


Figure 8: Number of defects under the base scenario and the case of implementing the policies

When the model is run for much longer periods (e.g. 3000 months), the base case scenario remains in a steady state, whereas the number of defects created when policies are implemented continues to decrease beyond 250 months and then reaches a steady state. Therefore, at steady state, the difference between the base case and the policies is greater, but since the simulation is started using real data and the current horizon of 250 months is already a very long time period, using the steady state results is not realistic.

3.2 Bounded Rationality and Information Availability

In this section, we analyze the impact of the availability of different levels of information on the number of defects. In the previous section, Weibull parameters of problems are estimated using failure data, and the parameter estimates are updated every month. To understand the impact of update frequency on the number of defects, we also report simulation results in which the estimates are performed once, when the problem becomes pending and never updated. We also analyze the other extreme case, in which engineers know the actual Weibull parameter values of problems as soon as the problems become pending. We simulate these three cases by implementing the three policies of using carryover information from the previous generation, using carryover information for prioritization, and having the option to solve problems for the next generation only.

Table 3 shows that updating the Weibull parameters every month leads to an 8% reduction in defects, compared to not updating them at all. Using actual Weibull parameters reduces the number of defects by another 5%. However, this is not a realistic case, since it is almost impossible to know the actual Weibull parameters. Therefore, updating Weibull parameters monthly obtains a significant part of the potential gains of possessing perfect information.

Use Carryover Information For Prioritization	Have the Option to Solve for Next Generation Only	Use Carry Over Information From Previous Generation	Weibull Parameter Estimation	Total Defects in 250 Months
Yes	Yes	Yes	Ranked Regression Estimate Once	100%
Yes	Yes	Yes	Ranked Regression Monthly Update	92%
Yes	Yes	Yes	Actual Weibull Parameters	87%

Table 3: Impact of information availability on the number of defects. Prioritization policies are implemented.

3.3 Impact of investigation policies on the number of defects

In the base scenario, problems are investigated after they accumulate three failures. In this section, we analyze the impact of different threshold levels on the number of defects. Having a low threshold speeds up the process of investigating important problems and puts them into the queue of pending problems faster. This, in turn, might lead to faster solutions for important problems and fewer defects caused by important problems. However, a low threshold also allows problems of less significance to become pending problems. Especially during the early phases of the lifecycle of a generation, when most problems have very few failures, a low threshold might increase the probability of solving a problem with limited importance.

To analyze the impact of this tradeoff on the number of defects, we ran the model with different threshold levels. In all simulations, prioritization policies are implemented, Weibull parameters are estimated with ranked regression, and the estimates are updated monthly. Since ranked regression requires at least 2 data points for regression, the minimum threshold level is 2. We ran simulations with threshold values between 2 and 10. Table 4 shows the simulation results.

Use Carryover Information For Prioritization	Have the Option to Solve for Next Generation Only	Use Carry Over Information From Previous Generation	Threshold for Investigation	Total Defects in 250 Months
Yes	Yes	Yes	2	100%
Yes	Yes	Yes	3	98%
Yes	Yes	Yes	4	98%
Yes	Yes	Yes	5	96%
Yes	Yes	Yes	6	98%
Yes	Yes	Yes	7	98%
Yes	Yes	Yes	8	99%
Yes	Yes	Yes	9	100%
Yes	Yes	Yes	10	101%

Table 4: Number of defects with different threshold levels for investigation. Weibull parameters are estimated with ranked regression, and estimates are updated monthly.

Overall, there is not much variation between different threshold levels. Investigating problems after 10 failures leads to the highest number of defects. A threshold of 5 failures leads to the lowest number of defects, reducing the number of defects by 2%, compared to the base case scenario of a threshold level of 3. However, these results are very close to each other, demonstrating that the threshold policy does not have much impact on the number of defects.

We also analyzed the impact of different threshold levels by using actual Weibull parameters for prioritization. This scenario eliminates the possibility of choosing less important problems, due to estimation errors, and favors lower thresholds. Since we do not use ranked regression in this case, the minimum threshold level is 1. A threshold level of 1 leads to a minimum number of defects (Table 5). The number of defects increases as the threshold increases, and the spread between the minimum and maximum numbers of defects is much bigger in this case.

Use Carryover Information For Prioritization	Have the Option to Solve for Next Generation Only	Use Carry Over Information From Previous Generation	Threshold for Investigation	Weibull Parameter Estimation	Total Defects in 250 Months
Yes	Yes	Yes	1	Actual Values	100%
Yes	Yes	Yes	2	Actual Values	103%
Yes	Yes	Yes	3	Actual Values	103%
Yes	Yes	Yes	4	Actual Values	104%
Yes	Yes	Yes	5	Actual Values	107%
Yes	Yes	Yes	6	Actual Values	107%
Yes	Yes	Yes	7	Actual Values	109%
Yes	Yes	Yes	8	Actual Values	109%
Yes	Yes	Yes	9	Actual Values	111%
Yes	Yes	Yes	10	Actual Values	112%

Table 5: Number of defects with different threshold levels for investigation. Actual Weibull parameters are used for prioritization.

3.4 No resource constraint

We simulated the model with no resource constraints. In this case, the investigation of problems starts after one failure, and the investigation delay is equal to 0. A problem becomes activated as soon as the first failure is observed. The distribution of the time to solve problems is the same as in other simulation runs. Under this scenario, the time between the creation of a problem and its solution is equal to the time delay before the first failure, plus the time to solve the problem. A comparison of the total number of defects created in 250 months reveals that this scenario reduces the number of defects by 66%, compared to the base case in which there are resource constraints and prioritization policies are not implemented. When compared to the resource constrained case in which policies are implemented, the reduction in the number of defects is 44%.

Simulation results are shown in Figure 9. In the first generation, the number of defects decreases substantially. However, it does not drop to 0, since some problems did not have 3 failures until the end of the production. In the following generations, new problems also increase the defect rate when a new generation is introduced, and they then are solved quickly.

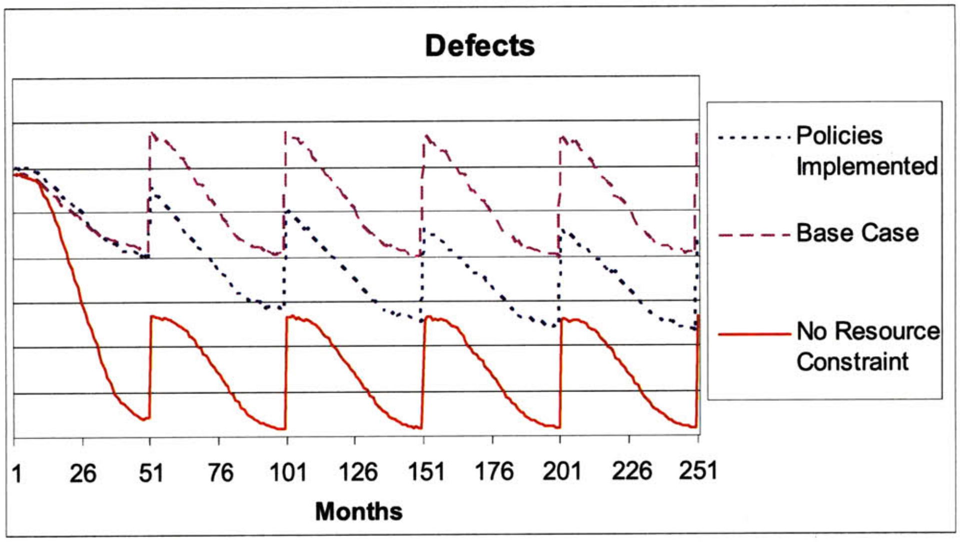


Figure 9: Simulation results with no resource constraints compared to resource constrained scenarios.

3.5 No carryover spike

We tested the impact of the carryover spike on a number of defects by setting the carryover spike to 0. Figure 10 shows the results of the scenarios with and without spikes. In both scenarios, all three policies are implemented. Eliminating the spike leads to a 6% reduction in the total number of defects. When policies are not implemented, eliminating the spike leads to an 8% reduction in defects.

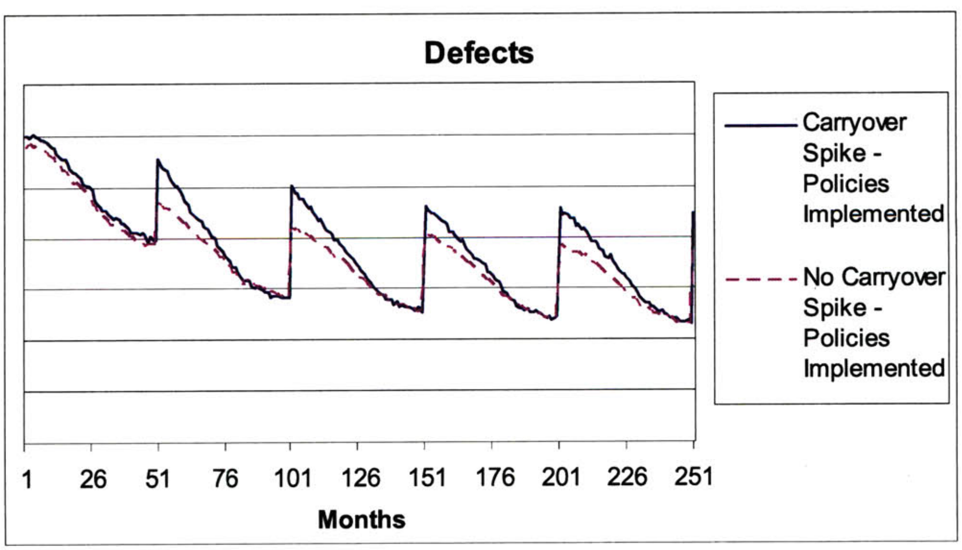


Figure 10: No carry-over spike scenario versus the carry-over spike scenario. All three policies are implemented in both cases.

Next, we analyze the impact of implementing the policies on the number of defects under the no spike case. Eliminating the spike leads to a minor reduction in the effectiveness of the three policies, but the policies still have considerable impact. Figure 11 shows the impact of implementing the three policies under the no carryover spike scenario. Policies reduce the total number of defects by 24%. Their impact was 25% with a carryover spike.

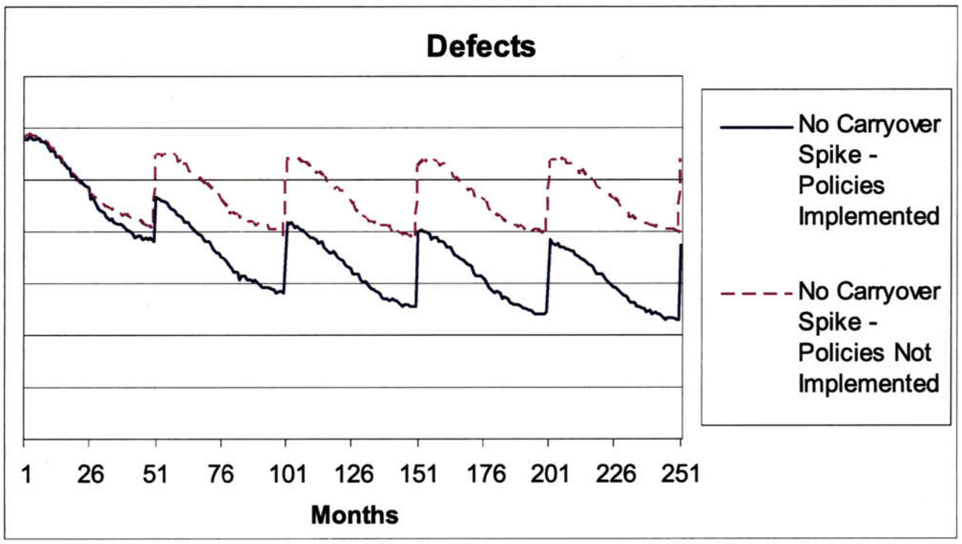


Figure 11: Impact of implementing the three policies under the no carryover spike scenario.

3.6 Fraction of New Parts

We simulated the model with different levels of fractions of new parts. In the base simulations, that fraction is 0.3. We also simulated the model with new parts fractions of 0.15 and 0.45. Figure 12 shows simulation runs in which all three policies were implemented. The total number of defects increases as the fraction of new parts increases.

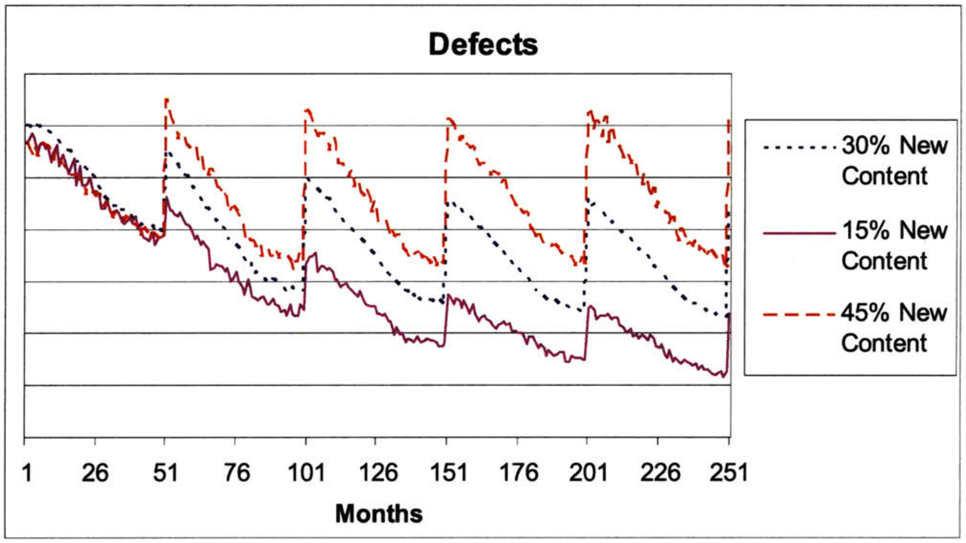


Figure 12: Simulation results with 15%, 30%, and 45% new parts. All three policies are implemented in all runs.

As the fraction of new parts increase, the effectiveness of the policies decreases slightly. Implementing the policies, versus not implementing them, leads to a 30% improvement in the 15% new parts case, a 25% improvement in the 30% new parts case, and a 24% improvement in the 45% new parts case. As the fraction of new parts decreases, the fraction of carryover parts increases and carryover policies become more effective.

3.7 Additional Resources

In this section we analyze the impact of additional resources on the cost and benefit of the field problem solving process, by increasing the amount of resources by 10%. The estimated warranty benefit is the difference between the number of defects multiplied the average warranty cost of a failure. However, the warranty cost does not represent the overall cost to the enterprise, since it does not capture the impact of unreliability on price and sales. We present the cost/benefit analysis for both metrics. On the other hand, the cost is estimated by multiplying the number of solved problems by the average cost of solving a problem. To preserve the confidentiality of our research partner, we do not provide those numbers in this paper.

The warranty benefits/cost ratio of increasing the number of engineers by 10% is less than 1 for both the base case and the case in which all three policies are implemented. Therefore, warranty benefits do not justify additional resources. However, the overall benefits/cost ratio is equal to 1.56, and this metric justifies additional resources.

4. Conclusion

We presented a disaggregated model that simulates failure creation, investigation, and prioritization processes at the level of an individual problem. The model was used to test the effectiveness of different prioritization policies regarding carryover problems. Simulation results show that implementing those policies leads to a 25% reduction in the number of defects over 250 months. Other policies, such as solving the problem with the maximum number of failures, also reduce the number of defects, compared to the base case, but they are not as effective as our policies.

We also tested the impact of different information availability scenarios and problem investigation policies on the reliability of products. Our results show that updating the Weibull estimates of problems monthly obtains a significant fraction of benefits yielded by perfect knowledge of Weibull parameters. Finally, simulation results reveal that the number of failures to start investigating a problem does not greatly impact product reliability.

These policies use several information cues regarding problems, such as complexity, importance, number of failures to date, hours until failure of those failures, whether the problem is going to be carried over, whether it is carried over, and so on. Unlike aggregated models in system dynamics literature, our disaggregated model enables us to answer policy questions that use all these information cues.

Appendix: Best-fitting distributions for sales delay, operating hours per month of the vehicle, and investigation delay.

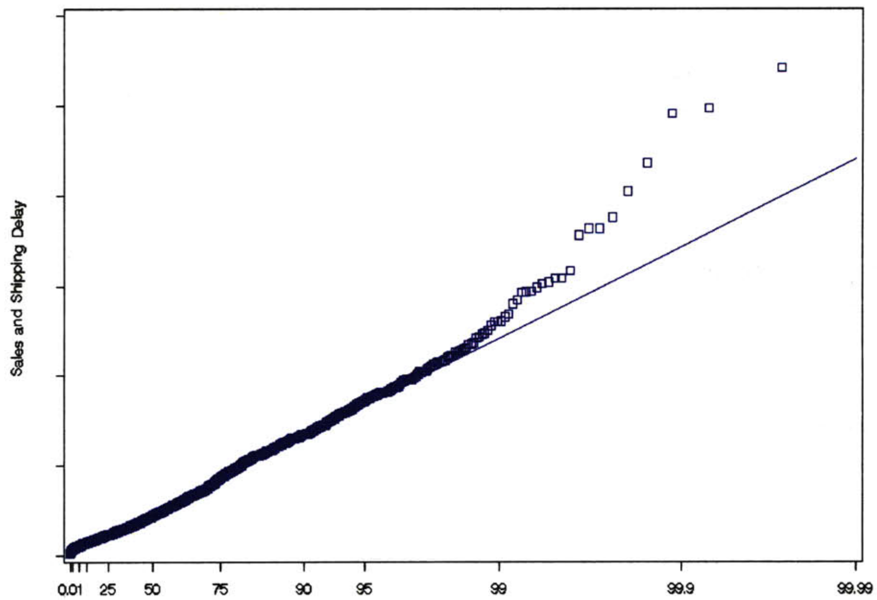


Figure 13: Weibull probability plot for sales and shipping delay

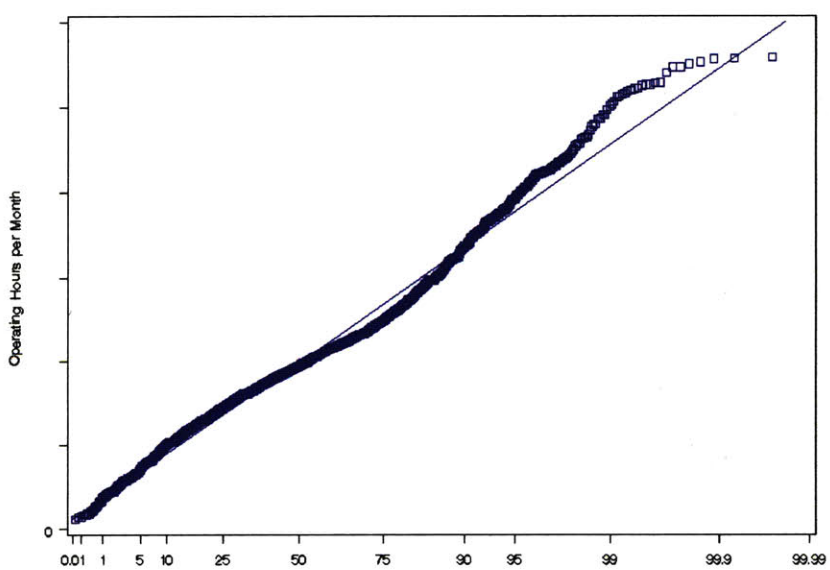


Figure 14: Weibull probability plot for operating hours per month for vehicles.

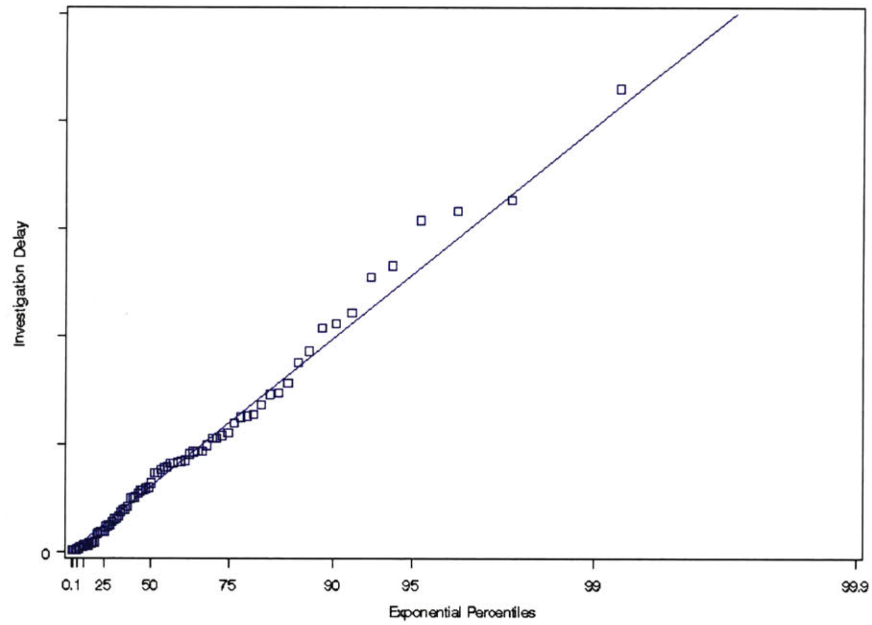


Figure 15: Exponential probability plot for investigation delay.

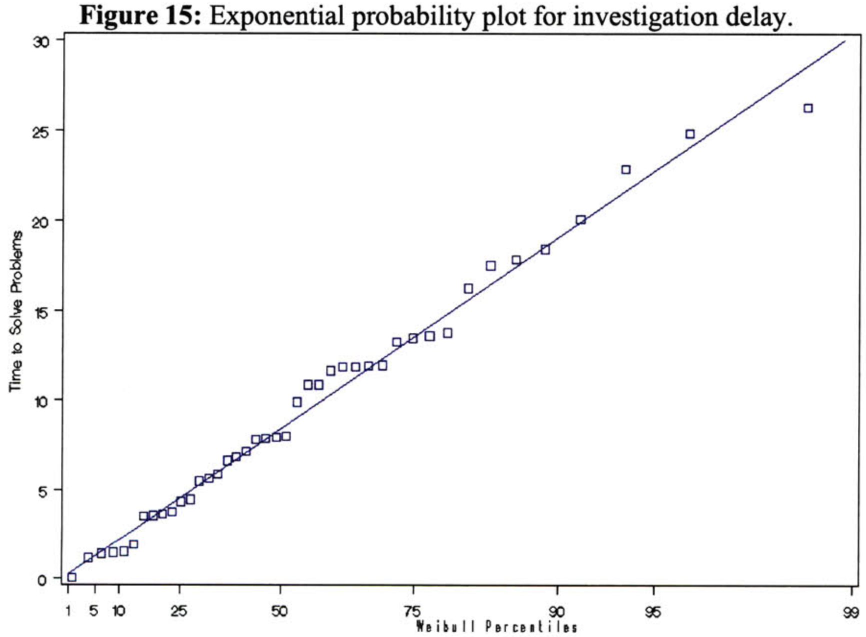


Figure 16: Weibull probability plot for time to solve problems.

References

Ford DN, Sterman JD. 1998. Dynamic modeling of product development processes. System Dynamics Review, 14(4) 309-340.

Mathpages. n.d. Retrieved August 10, 2008, from http://www.mathpages.com/home/kmath122/kmath122.htm .

Repenning NP. 2001. Understanding fire fighting in new product development. The Journal of Product Innovation Management, 18 285-300.

Morecroft, J. 1983. System dynamics: Portraying bounded rationality. Omega, 11(2) 131-142.

Thomke S. 2003. Experimentation Matters: Unlocking the Potential of New Technologies for Innovation. Harvard Business School Press: Boston, MA, USA.