2-1-2013

Warranty prediction during product development: Developing an event generation engine

Sivakumar Sundar

ROCHESTER INSTITUTE OF TECHNOLOGY

Warranty Prediction during Product Development: Developing an Event Generation Engine

A Thesis

Submitted In Partial Fulfillment of the Requirements for the Degree Of

Master of Science in Industrial and Systems Engineering

Department Of Industrial and Systems Engineering

Kate Gleason College of Engineering

Sivakumar Sundar

February, 2013

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NY

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Sivakumar Sundar
Has Been Examined and Approved By The
Thesis Committee as Satisfactory For The
Thesis Requirements For The
Master of Science Degree

Approved by:

Dr. Marcos Esterman, Thesis Advisor

Dr. Carol J. Romanowski, Committee Member

Dr. Brian K. Thorn, Committee Member

Dedication

This work is dedicated to my parents for having provided unconditional support over all these years. I love you!

Acknowledgements

First I would like to thank Dr. Marcos Esterman for having guided me throughout this work. His constant encouragement over the last two years has been invaluable and he has also helped my professional career immensely. The best part about him is that I always feel that I need to work more after our meetings and this speaks about his ability to motivate his students. Thank you so much for your patience, I enjoyed every moment I worked with you.

I would also like to thank Dr. Carol Romanowski for having helped me understand some of the statistical concepts and providing valuable input over the last couple of years. I also like to acknowledge Dr. Brian Thorn for all his help.

I would also like to thank the Department of Industrial and Systems Engineering for providing me with all the resources and a great atmosphere for research.

Finally, I want to give a shout out to my roommates Sidhartth, Hari and Karthikeyan for making my stay in Rochester memorable. I really enjoyed every moment I spent with you guys!

Abstract

In 2010, high tech industries including computer makers, peripherals manufacturers, and medical equipment manufacturers spent a total of $8 billion on warranty. Reducing warranty costs improves the manufacturer's profit and helps to reduce the overall cost of the product. An often cited principle is that approximately 80% of the eventual product cost is 'locked in' during the very early stages of product development, however, traditional methods of warranty analysis are not well suited to predict the warranty costs during these early stages. Thus, product development personnel need better tools to make good predictions about the warranty costs so that they can make better decisions to reduce warranty earlier in product development.

In order to address this gap, previous research defined a warranty prediction framework, which at its core was a warranty event generation engine that integrated the disparate data sources available early in the product development process. The objective of this work was to create this event generation model, which would give the probability of occurrence for a warranty event, given the length of time of service for the system. The model developed in this work used different data sources namely, field data, product development data and engineering judgment data from our industrial partner. The datasets were then combined using a two-stage numerical Bayes method to predict the probability of occurrence of an event. Various test cases were created, by using the different datasets as priors and likelihoods. The results were then compared to actual field data set to understand how well the model performed. It was found that the model performed well and was able to produce a bounded solution. The future research agenda is to create a tool for product development professionals that will help them predict warranty costs.

Contents

1 Introduction

1.1 Motivation

A product warranty is a contractual obligation between the buyer and the manufacturer, which assures the buyer that the product will perform as represented (W. Blischke & Murthy, 1995). So if the product fails to perform satisfactorily during its warranty period, the manufacturer incurs warranty costs. Warranty costs for the manufacturer includes the cost of the components for repair, shipping charges and labor costs. To put this point in perspective, let's look at an example.

For approximately 80% of the products produced in the world, an average of 3% of the product cost (measured in terms of net sales) is due to the warranty coverage that is provided by the manufacturer (W. Blischke & Murthy, 1995) . In 2010, high tech industries including computer makers, peripherals manufacturers, and medical equipment manufacturers spent a total of $8 billion on warranty ("Eighth Annual Warranty Report, Totals & Averages, 1 April 2011,"). So keeping the warranty costs down helps to improve a manufacturer's profit or reduce the product's cost.

From a buyer's perspective warranty provides protection by assuring the buyer that in case the product is defective, it will either be repaired or replaced at a reduced or at no cost to the consumer. The buyer may also infer the reliability of the product from its warranty terms (W. Blischke & Murthy, 1993).

From a producer's perspective, warranty has two roles. One, it is protectional. Warranty terms specify the use condition under which the warranty will be honored. So in the case of product misuse the warranty coverage may be minimal or nonexistent. The second role is promotional. That is, the warranty is used as an advertising tool that conveys the producer's willingness to stand behind their product, and thus it has become an important criterion for buying a product. (W. Blischke & Murthy, 1993)

According to Wilson (Wilson, 1993), getting the product definition right early in product development is important as it gives designers the greatest leverage in saving costs through design decisions; thus a strategy is needed to understand what the warranty costs of a product will be early in the product development process. The challenge is that at that point in product

development, there is generally not enough available data, which makes it extremely difficult to understand what the ultimate warranty cost will be when the product is launched.

1.2 Background

Esterman et al. (Esterman, Gerst, Stiebitz, & Ishil, 2005) identified some important aspects of warranty. They found that traditional methods for assessing warranty were inadequate. The problem was that either the design engineer was not aware of the impact of warranty events after identification of warranty events or they were not able to identify the warranty events in the first place.

In a Stanford research roundtable organized around this topic, some of the needs to be addressed in design for reliability and warranty emerged (Esterman et al., 2005). One of the issues in complex systems is the identification of failure events. In these systems, a class of failures that is most concern is "unknown - unknown" which are unanticipated failures from mechanisms that are not understood well. One of the opportunities expressed was need for effectively identifying these failures.

Another theme was the need for models which lead to improved reliability predictions. So an opportunity here is to develop event rate prediction models that can efficiently integrate historical field experience, product development testing data and other quantitative and qualitative data.

The aim of this work is to effectively combine these different streams of data using inference methods to create an event generation engine for predicting warranty events for a product.

1.3 Problem Statement

The overarching goal of this research is to develop a model that can be used to by product development personnel to accurately predict warranty events. The model will also be used as a tool for management to understand warranty costs. Once the importance of the model is established this will enable streamlined data collection at various stages in the products life cycle so that the model performance can be improved by reduction data integrity issues.

The specific research questions this thesis aims to answer are

1. Can we integrate different data streams to predict the occurrence of a warranty event?
2. Can the data used be of continuous form?
3. What is the best approach to integrate the data streams?
4. Can the model be transferred to different platforms without major modifications?
5. Can the model be applied to actual product data and produce good results?

2 Literature review

Warranty and warranty prediction are complex topics and have different components associated with them. This section will orient the reader toward understanding the basic concepts of reliability and warranty analysis. Warranty and reliability are closely connected and many of the models and concepts developed for reliability can be applied to warranty as well. This section is divided into three parts. The first section will review traditional reliability analysis; what it means, how it is performed using different data sets, and especially how it is used during product development. The second section provides an overview of warranty in general; how warranty analysis is performed and how warranty analyses are different from traditional reliability analysis. The third section summarizes the few integrated approaches in reliability and warranty prediction that this work has uncovered.

2.1 Reliability analysis

Reliability's importance cannot be overstated and today customers expect a product to work well at no extra cost. It is also evident that reliability is a process that can be characterized, controlled and improved. So there are many different ways that these steps can be performed and some of them are discussed below.

2.1.1 Reliability growth models

Early in product development the reliability performance is far from the desired requirement and a formal growth program can be implemented to find the causes of poor reliability and define mitigation strategies to improve the reliability of the system. This process is then modeled to understand what the reliability will be at a future point in time based on the estimated rate of reliability improvement.

This section examines some of the work in the area of reliability growth models that are mainly dependent on product development test data. This approach is one of the most widely used methods to improve product reliability in industries (Ireson, Coombs, & Moss, 1996). The earliest models of reliability growth models were proposed by Duane (Duane, 1964). In this work it is assumed that the reliability is zero and the rate of failure is infinite at the start of the project, and the failure rate declines to zero when development time tends to infinity (Duane,

1964). Thus the reliability can be increased by testing and making the necessary changes continuously.

But this assumption is unrealistic and thus Murthy and Hussain (Hussain & Murthy, 2003) came up with a different model in which the failure distribution is not exponential, and the failure rate at the end of product development effort is a random variable, suggesting that more time spent on product development does not always mean improved reliability. The motivation for this model was that there is a need to find a tradeoff between product development cost and warranty costs associated with the product. Therefore such growth models give us an idea of the warranty costs associated with the product early in product development (Hussain & Murthy, 2003).

One of the drawbacks of reliability growth models is that these models mostly focus on reliability improvements during testing of the products, but there are very few models that focus on reliability during design. Improving reliability by testing and then making corrective actions is not as efficient as improving the reliability during the design process.

Recently there have been many works in the area of reliability prediction using field data from warranty claims (Huaiqing & Meeker, 2002; Ion & Sander, 2005; Yang & Cekecek, 2004). Since many of the products in the markets are evolutionary in nature, the field data are an important indicator of how the next iteration of the product might perform.

2.1.2 Reliability modeling using warranty Data

Majeske et al. (Majeske, Lynch-Caris, & Herrin, 1997) recommended a hazard function analysis to evaluate the impact of process and product changes on the time to first warranty claim. They achieved this by classifying the product and process changes into three categories. The three categories are

1. Adding customer features.
2. Quality and reliability improvements and
3. Cost reduction opportunities

The entire range of field data, collected from an automobile manufacturer, was classified into one of the three categories. The data consisted of field issues of a radio chassis in an automobile

for a period of three years. An important component in this research was characterizing the lifetime of the product in question so that warranty events occurring only inside this specified lifetime were considered.

Another consideration in this work was that a product is not sold immediately after it is manufactured. Thus there is a time lag between production and sale. So it is necessary to understand the mean time difference between production and sale in order to correctly classify the product in one of the categories mentioned above. A hazard plot analysis on the field data from this work showed that the increase in complexity of the product increased the warranty cost. When a new design for a radio chassis was created, it increased the warranty cost (Majeske et al., 1997). Also this work reinforced the fact that current product support and new product design engineers should constantly communicate with each other in order to avoid the design changes that negatively affect the product warranty cost in the long run (Majeske et al., 1997).

Another work showing the importance of field data is by Huaiqing and Meeker (2002). This work is about the early detection of reliability issues using warranty databases (Huaiqing & Meeker, 2002). The authors used automobile warranty data as their test data. The warranty database that was used is actually a combination of the production data which includes identification number, date of manufacture, date of sale etc. and also field warranty data like date of repair, problem code, cost of action etc.

The early detection is modeled using a non-parametric approach and thus it is flexible as it can be used for different problem warranty issues (codes which specify problems like breakage, non-functioning, etc.) without much modification. The model uses the warranty data to report the occurrences of the problem. Once a certain number of occurrences has been reported for a particular failure event, a flag is raised which indicates an issue to be investigated by the appropriate product engineer. This "red flag" threshold is determined based on different parameters like false alarm probabilities, units sold as a fraction of units produced, etc. (Huaiqing & Meeker, 2002).

This model did a good job of detecting the reliability issues that might occur in the field. But the model was not able to capture the effects of seasonality. Also serious issues that happen in the field are generally due to sudden changes in design or manufacturing processes. This model does

not efficiently detect these issues in its current form. Another drawback is that this is not directly used by the product development engineers. So if a problem is discovered by the model, the design engineers should be made aware of it and thus make the necessary changes in the next iteration of the product.

A common theme in both of these works is that, there is not an obvious linking of the information that has been generated back to the product development engineers on a continuous basis. Yang and Cekecek (Yang & Cekecek, 2004) proposed an approach that addresses the shortcoming identified above. It is important to understand which parts have the most number of failures, and reliability improvement must be geared towards those parts so that we can reduce warranty costs on that part. Vulnerability of a design depends on the structure of design. Design can be defined as a mapping of functional requirements in functional space and design parameters in component space. The expected cost of vulnerability is proportional to the component replacement cost and cost of losing the relevant functional requirements (Yang & Cekecek, 2004). The warranty data that are available can be used to identify which components and subsystems fail the most in a product, and when coupled with dissatisfaction scores, expected warranty cost can be estimated.

Here the authors define improvement cost as the cost associated with improving the reliability of the product. Therefore the total cost of the product is the sum of vulnerability cost and improvement cost. This is then converted into an optimization problem and the goal is to reduce the total cost. Finally they conclude that design improvements should be targeted at those components where the delta between the improvement cost and vulnerability cost is the highest (Yang & Cekecek, 2004). Their framework thus assigns the responsibility of improving a product's reliability to the product development engineers who can now understand the highest drivers of warranty costs based on field information.

Even though this work suggests a way to send feedback to the product development team, it only uses a single data source to improve reliability. Furthermore, their analysis is a one-time analysis, but typically in product development there is a continuous stream of data from various sources that are available and these should be used to the maximum possible extent and updated frequently.

2.1.3 Reliability modeling using manufacturing data

One other source of data that can be used for reliability modeling is the manufacturing data captured by MRP systems.

Mannar et al. (Mannar & Ceglarek, 2005) proposed a fault region localization methodology that linked warranty failures to manufacturing measurements. This work helped to identify the relationship between warranty failures, design parameters and process variables.

Figure 1 illustrates a product lifecycle. During the design phase, the functional requirements and tolerances for each of the requirement are specified. During manufacturing, various test equipment capture the important parameters and during its operation in the field, various field performance measures are captured.

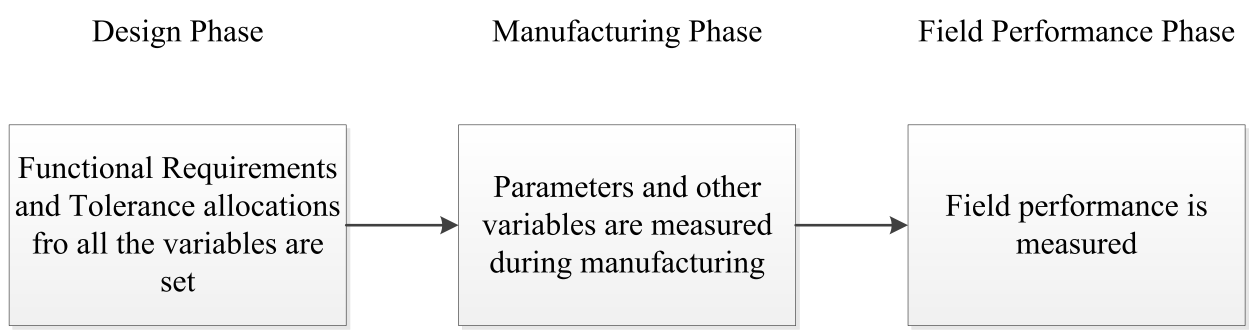

Figure 1: Product life cycle information for a multi-station manufacturing system. Adapted from (Mannar & Ceglarek, 2005)

After a specific failure scenario is established, the authors identified the manufacturing measurements that were related to specific failures and then used a generalized rough sets methodology to identify the fault region. They accomplished this in two steps.

In the first step they used a supervised classification methodology to come up with a subset of manufacturing parameters and then this is used to identify the warranty fault region and the boundary region. The next step is revaluating the tolerances so as to reduce field failures. The tolerance from this method provides an interpretation of the results in the form of warranty fault

region, normal region and boundary region graphically. Thus the design engineers using this information can set the tolerances of the particular variable accordingly in the next iteration.

This work is important in two ways. First it identifies an opportunity to link the manufacturing data to failures in the field. Second is the ability to combine two disparate datasets to identify opportunities for improvement. But this work does not take inputs from the test data that are available during product development.

2.1.4 Reliability modeling during product development

Bayesian methods are commonly used in reliability modeling as they offer the flexibility of updating the model as more information becomes available (Mazzuchi & Soyer, 1993; Ming, Zhang, Tao, & Chen, 2010; Zhou, Jin, Dong, & Zhou, 2006). Some of these works that model reliability using Bayesian methods are discussed below.

Mazzuchi et al. (Mazzuchi & Soyer, 1993) proposed a complete Bayesian model for the Barlow-Scheuer reliability growth model by unifying all the existing approaches in reliability growth. They developed a methodology to predict the reliability of the product during the early stages of testing. The failure modes in the reliability model are of two types namely

- Non fixable failures (failure which can be corrected only after significant technological advancement) and
- Fixable failures (which can be solved by making small modifications)

The authors developed a methodology which incorporated an initial test during product development and assigned the failure found during testing to one of two failures types mentioned. Then after every subsequent test, design changes are made to improve the reliability. In this model the test data from the previous iteration of tests are used as the prior distribution and after subsequent testing, this probability is updated. The important differentiating factor from other approaches is that this gives future reliability estimates of the product instead of just current reliability estimates (Mazzuchi & Soyer, 1993). Using their approach it is possible to determine the amount of testing required to complete the product development program and also assess the performance of the product in the field. Here the focus of the authors is the perceived reliability growth pattern as a whole and not just for a single stage of the improved design.

Zhou et al. (Zhou et al., 2006) used a different variant of the Bayes theorem. For their analysis they used Bayesian networks. Generally, traditional reliability analysis has considered that failure modes are binary in nature. But often this is not the case. Most products have various failure modes, or components that have only one failure mode but with different degree of failures. Therefore it is important to have a method for obtaining reliability parameters for generic multi-state systems.

In multi state systems, Bayesian networks have a well defined theory of probabilistic reasoning and a way to handle probability events in reliability analysis. The important feature of Bayesian networks is that instead of just giving a probability of a failure mode, it also gives the severity of a single component in the system or subset of components based on the occurrence of the failure mode (Zhou et al., 2006).

A Bayesian network is generally constructed based on a reliability block diagram, where each component has different states. Once this is done, the leaf nodes and the intermediate nodes for the system are identified and, based on the probability distribution of failures of corresponding components in the system, the conditional probability of each state of the system can be calculated.

As an example, the Bayesian network of a system is shown in Figure 2. By knowing the probability of the leaf nodes (last row), the probability of the failure for the subsystem or for the entire system can be calculated (Zhou et al., 2006).

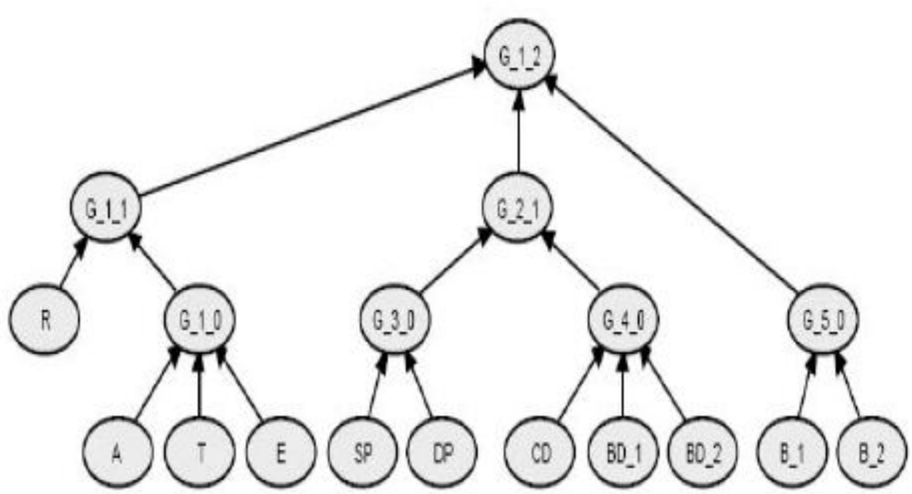

Figure 2 : Bayesian network of radar system (Zhou et al., 2006)

Thus Bayesian networks could be used when complex dependencies of the components exist in the system. This work demonstrated the potential of using Bayesian networks for reliability analysis. These analyses are quantitative in nature and qualitative methods need to be developed in order for this approach to be more effective.

The applications of Bayesian statistics to reliability analyses demonstrate that they provide reliability estimates in terms of probability instead of point estimates. This is useful because it prevents people from becoming fixated on a single number when, in fact, reliability cannot be exactly quantified.

In this section we have seen how traditional reliability analyses are performed; how reliability growth models use product development test data to estimate the reliability of the product; how field data and manufacturing data can be used to perform reliability analysis; how Bayesian approaches are used in reliability modeling. Again it is emphasized that warranty is different from reliability and thus we need methods where warranty analysis can be performed. The next section provides a background on warranty and warranty modeling.

2.2 Warranty analysis

2.2.1 Introduction to warranty

As mentioned earlier, warranty is a contractual obligation between the buyer and the manufacturer, who assures the buyer that the product will perform as represented (W. R. Blischke & Murthy, 1992)

Currently there are many different types of warranty policies that are available for different classes of product. Blischke and Murthy (1992) give a good overview of them.

The most common type warranty policy seen in many of the products is a "simple non-renewing single item not involving product development" warranty policy. Though equipment like medical devices and printers fall into the category of renewing a "single item not involving product development" warranty, where service contracts are extended on a yearly basis. Other types of warranty policy are shown in Figure 3.

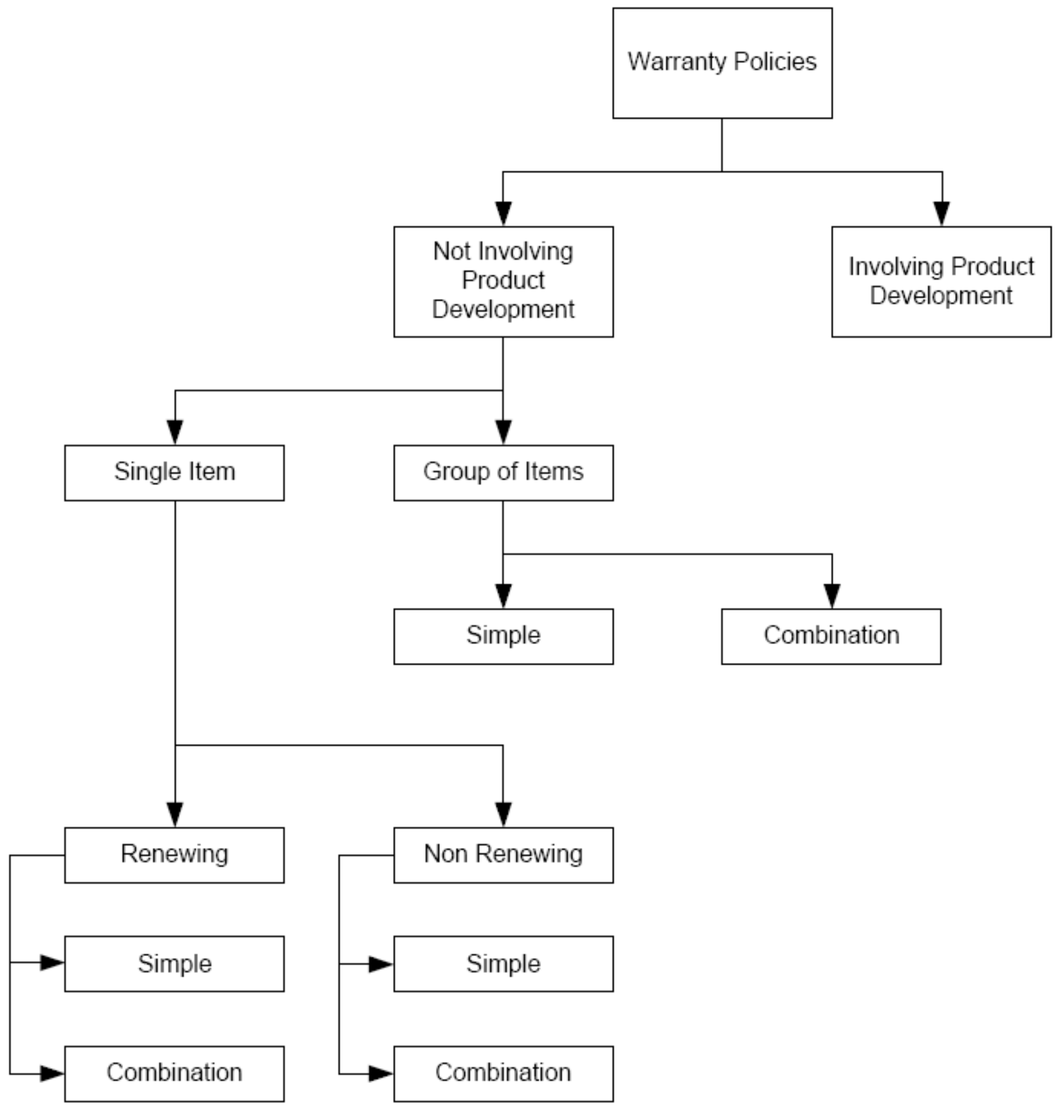

Figure 3 : Types of warranty policies

Very complex products like aircrafts and military equipment may have a warranty policy which involves product development, as these are not one shot systems and usually remain in use for very long periods of time.

2.2.2 Warranty analysis approaches

The most frequent parameters that are of interest in warranty analysis are the total expected cost of warranty and the warranty cost per unit time over the warranty period and over the life cycle of the product (Chukova, Arnold, & Wang, 2004). These approaches are different from

reliability analyses as these model the warranty events and warranty costs using the field or other product development data rather than predicting the reliability or failure event for the product.

One way to analyze and predict warranty is to model failure and repair processes. There are different kinds of repairs that can be performed that include:

a) Improved Repair – A repair that takes the product to better state than when it was purchased
b) Complete Repair – A repair that brings the product to as good as new condition
c) Imperfect repair – A repair that is a noticeable improvement to the product.
d) Minimal Repair – A repair that brings the product to a functioning state without improving the performance
e) Worse Repair – A repair that causes worsening of the product
f) Worst repair – A repair that accidentally destroys the product

The type of repair depends on the warranty terms, cost, safety requirements etc. Chukova et al. (2004) proposed a way to model these imperfect repairs. Every product has a lifetime distribution once it is in the field. Once a failure has occurred and it is repaired, this lifetime distribution changes for the product and can be modeled using characteristics like failure rate, mean time to failure and the cumulative distribution function. The repair could also be any of the ones mentioned above and thus have an influence on the lifetime distribution.

Chukova (2004) introduced an indexing parameter τ which specifies the degree of repair. For example, improved repairs have $\tau > 1$; for complete repair $\tau = 1$; for minimal repair $\tau = 0$. The value of τ can depend upon various factors like the time remaining until the expiration of the warranty period; the time elapsed since purchase; the length of time since the previous repair. This parameter τ is then introduced to the conventional life distribution characteristics and is evaluated numerically.

Thus once these functions are determined the lifetime distribution for a product undergoing different kinds of repairs during its lifetime can be determined using simulations of warranty data. This method is very effective in understanding the effects of repair but during product development it is very difficult to accurately predict the repair type that will be used for a particular failure event.

There has also been an interest estimating warranty cost from Failure Modes Effects Analysis (FMEA). Vintr et al. (Vintr & Vintr, 2005) proposed an approach to estimate warranty costs using FMEA. They calculated the total cost of warranty claims would be

$$C_c = C_R + C_T + C_D$$

Where C_c is the total warranty claim cost
C_R is cost of replacement components
C_T is the transportation costs and
C_D is the administrative cost

It is not possible to predict the warranty cost of every single sold product and thus an expected cost per item is calculated. So the FMEA listed out all the potential failure modes along with their occurrence and severity. Vintr et al. (2005) suggested adding columns like number of repairs expected, type of repair and other cost involved in the particular failure mode. Once that is done the number of failures can be estimated and thus the total expected cost of a warranty claim can be calculated using the expression

$$E[C(W)] = E[N(W)](E[C_M] + E[C_S] + E[C_A])$$

Where $E[C(W)]$ is the expected warranty cost.
$E[N(W)]$ is the expected number of failures
$E[C_M]$ is the man hour cost
$E[C_S]$ is the material and spare parts cost
$E[C_A]$ is the additional administrative costs

A sample warranty cost analysis form is show in Figure 4. This methodology is particularly useful because FMEAs are generally carried out during the early design stages (when applied properly) and with a little more effort this analysis can be easily extended to estimate warranty costs. The issue, though, is establishing a database which accurately tracks the spare cost, travel costs, etc. Also while using this information is useful, it does not account for warranty events that are found later in the product development—thus minimizing its usefulness.

Table 2 - Warranty costs analysis							
Failure code	Failure rate (F/10^{-6} hr)	Repair type	Number of failure (10^{-3})	Labor consumption (Man-hours/F)	Spare parts and material ($/F)	Additional costs ($/F)	Overall costs per warranty ($)
i	λ_i	-	$\overline{N}_i(W)$	$\overline{M}_i$	$\overline{S}_i$	$\overline{A}_i$	$\overline{C}_i(W)$
1	0.12	reconnection (MR)	0.701	0.5	-	15	0.0210
2	0.05	solenoid replacement (PR)	0.292	1.1	18.5	15	0.0194
3	1.20	solenoid replacement (PR)	7.01	1.1	18.5	15	0.4660
4	1.50	spring replacement (PR)	8.76	0.8	1.20	15	0.3520
5	0.12	reconnection (MR)	0.701	0.5	-	15	0.0210
6	0.45	relay replacement (PR)	2.63	1.5	11.2	15	0.1870
7	0.15	relay replacement (PR)	0.876	1.5	11.2	15	0.0624
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Σ	**18.3**	**-**	**107**	**35.7**	**309**	**525**	**5.82**

Figure 4 : Example of warranty cost analysis using FMEA

2.2.3 Difficulties in using warranty data

Although there are many advantages of field data for predicting reliability and warranty, field data is generally considered as “dirty” data. The service personnel don’t always have the time to log in customer problems accurately, which leads not only to data inaccuracies and missing data, but difficulty in identifying and modeling specific and actionable events. However, there is still valuable information that can be gleaned from these data. It has been observed that field data holds important information regarding the reliability of products which is influenced by the use patterns of the different customers. It is extremely important to use these data to analyze reliability and warranty improvements.

These works also reinforce the fact that warranty field data alone is not sufficient and that other streams of data are needed to effectively understand reliability issues.

2.3 Integrated approaches for warranty and reliability analysis

2.3.1 Challenges in warranty analysis during product development

From the previous studies it has been seen that it is important to have an integrated approach to warranty analysis during product development. This section summarizes some of the previous work in this regard.

There are many challenges when trying to perform warranty analysis during product development. During product development the interest is mainly to improve reliability and not necessarily to characterize it. Also early in product development, the design is not fixed and is constantly changing, and thus reliability testing is not practical during these early stages of product development.

During product development there is very little direct evidence that could predict reliability, thus the present methods do not accurately predict the reliability in these early stages. Another problem is that during internal testing the product is not run as a customer would run the product, and thus some problems are never identified that result in warranty dollars once the product is released. Conversely, it is also possible for problems that would not manifest themselves to receive undue attention. Also things that might be important to the product development team might not generate significant warranty costs.

In a nutshell, lack of resources, treating reliability and warranty analyses as a one-time analysis, and failure to leverage all the information available at a given point in time to conduct the analyses are all major challenges.

2.3.2 Review of approaches

Some work has been done looking at warranty from an organizational perspective. Murthy & Djamaludin (Murthy & Djamaludin, 2002) showed how different parts of the organization called "modules" interact with each other in a warranty management system. Optimizing reliability from a particular module's perspective will not achieve the right results for the organization, but rather how these work in tandem. Figure 5 shows the interaction between the various parts of the organization from a warranty management perspective.

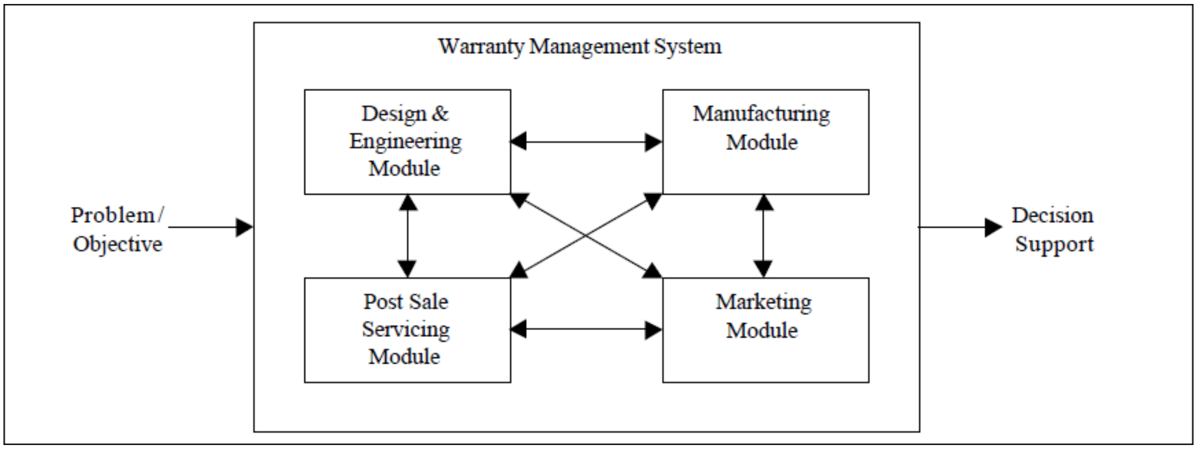

Figure 5 : Warranty management system

Figure 6 shows how different modules have different components which affect total warranty costs. So this further shows how important it is to have an overall organizational perspective to reliability. Estimating the cost also depends on the quality, quantity and timeliness of the data received from the different modules. So different data sources and the effectiveness in combining them will produce the best results in predicting or analyzing warranty.

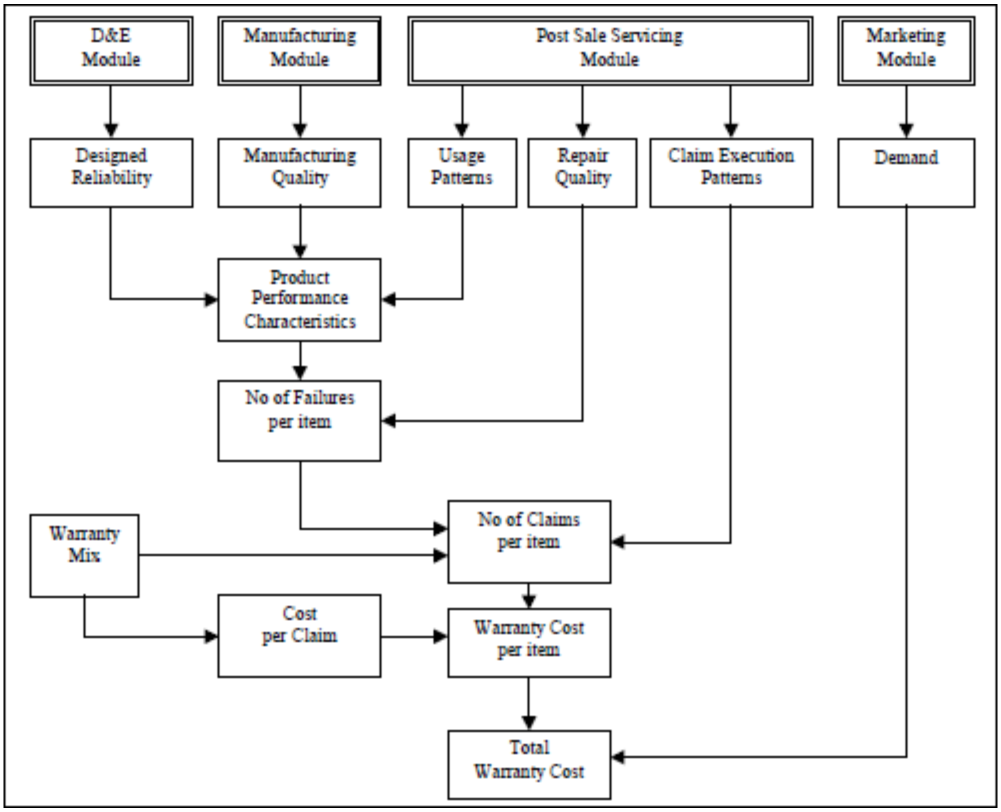

Figure 6 : Integrated model for total warranty costs

There have been some models that have proposed to integrate different data sources. De et al. (De, Das, & Sureka, 2010) proposed a method to find the root cause for a warranty failure using various sources of information that are available within an organization.

The first step converts FMEA into an Ontology-Relationship Diagram (ORD). ORD is a knowledge capture model which is based on the cause and effect principle. An ORD is drawn for every part in a product. Since this can be extremely cumbersome, we can either use this on the most failure prone part or those parts which trigger failure. Once this is done, the ORD is converted into a probabilistic network using Bayesian networks where the probability of failure is based on the warranty information and is calculated for each of the failure.

The framework also included the use of Corrective action reports (CAR) where the manufacturing department can enter important information regarding the problems that they

faced either due to supplier issues or in house manufacturing issues. Once these reports are created, these served as additional data to find the root cause if a problem arises in the field.

Another component of this methodology is the processing of the warranty claims database. The claims database is searched for specific strings and this is compared to the ORD developed for the product. Thus if a new claim is reported, using text processing algorithms, the root cause can be found out by searching the ORD.

This approach will help in faster detection-to-correction time for an organization and will act as an early warning for the organization about potential warranty issues, leading to reduced warranty costs due to field failures. Another advantage of this approach is the wide availability of electronic information. This helps to identify problems earlier so that product engineers can react faster to the issues.

Yadav et al. (Yadav, Singh, Goel, & Itabashi-Campbell, 2003) advanced this approach by describing a way to assess and predict reliability during the product development process which utilizes qualitative (fuzzy) information, prior knowledge, and quantitative data. The important difference from the previous approach and this work is the inclusion of engineering judgments as a source of data. By integrating all existing data like test data, field data, etc., better accuracy and realistic estimates of product reliability can be achieved.

In order to effectively track and manage reliability improvement during the development phase, continuous reliability estimation is necessary as a product moves from one design phase to another. As shown in Figure 7, the authors incorporated the fundamental Bayes theorem with fuzzy logic reasoning to enhance the capability of the Bayesian model to accept fuzzy information along with other information. This is due to the subjective and qualitative nature of engineering judgment, as well as other factors that do not provide hard numerical data.

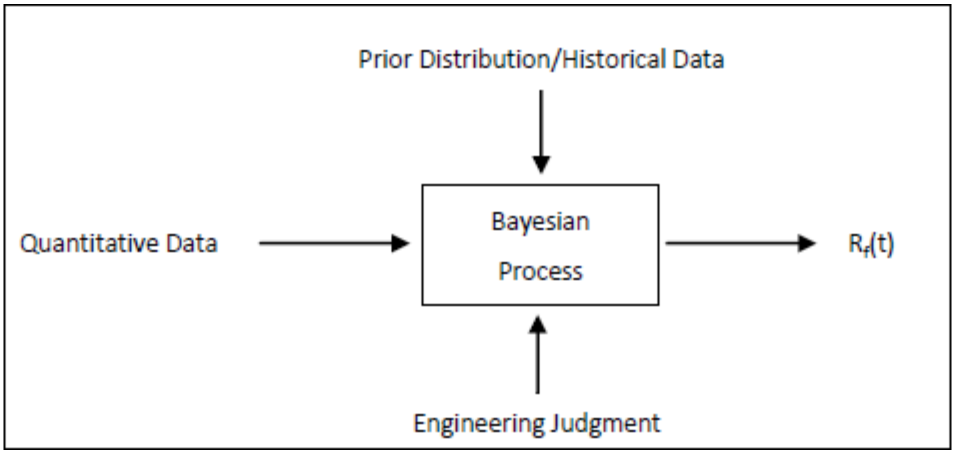

Figure 7 : Input output model. Adapted from (Yadav et al., 2003)

They suggest that by calculating the reliability estimate at each design phase, it will be possible to increase product reliability over time. This approach will result in a revised reliability estimate at the end of each stage which incorporates the engineering judgment for design changes, corrective actions, and other qualitative information. Ideally, this estimate will show a positive change in reliability improvement at the end of each stage.

2.4 Summary

Different approaches to improve reliability and predict warranty during product development have been presented. These studies show that an integrated approach to warranty prediction during product development is important in order to have more accurate predictions. This work seeks to define an approach that will effective integrate field data, product development data and engineering judgment data in predicting the warranty. The main goal is to generate the time to failure for a particular warranty event.

3 Methodology

This aim of this work is to develop an event generation model that can be used to predict the frequency of a particular warranty event. This model will allow warranty performance predictions during the product development phase by accomplishing the following objectives:

- The model provides insights to the development team for actions that they can take to mitigate warranty costs.
- It facilitates decision-making by increasing the product developers' and managers' confidence that their actions are leading to improved warranty performance in the field.
- It provides the management team an accurate projection of warranty costs so that the enterprise can appropriately plan for the financial impact of these costs. These impacts include product pricing; extended warranty support pricing, service inventory requirements, warranty accruals, etc.

The research is a more detailed development of the concepts described in Esterman et al. (Esterman et al., 2005). That work described that in order to predict the warranty performance for a product various steps need to be performed.

The first step is the identification of the relevant warranty event. The process of identifying this event will be described in greater detail below. Once the warranty event is identified, its probability of occurrence given the length of time in service of the product or system is characterized. From this probability, the expected cost for each warranty event can be calculated. These steps are shown in Figure 8. This research is focused primarily on developing the Warranty Scenario identification and Warranty Event generation processes that were proposed, but not detailed, in previous research.

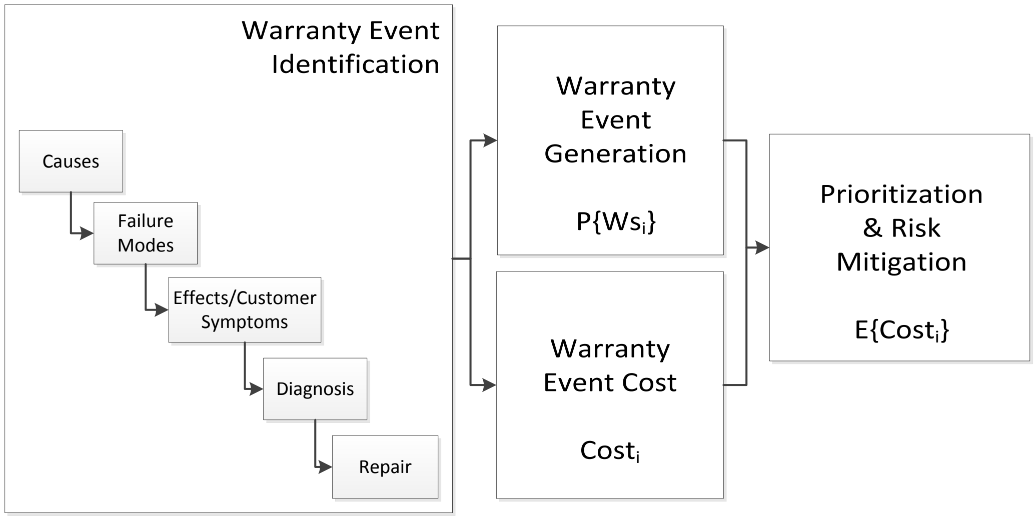

Figure 8 : Framework for predicting warranty performance during product development. Adapted from Esterman et al. (Esterman et al., 2005)

The focus of this work has three parts:

1. Identification of warranty scenarios, and ultimately warranty events
2. Characterization of the different available datasets needed for predicting warranty event rates in the field
3. The integration of these different datasets into a single event frequency prediction model.

These three sections will be discussed in detail in the following sections.

3.1 Identification of warranty scenarios

A warranty scenario is a family of warranty events that share similar roots causes for a particular symptom or a related set of symptoms and also share a similar set of diagnosis and repair steps associated with the symptom(s) as shown in Figure 9. The figure shows a complete warranty scenario.

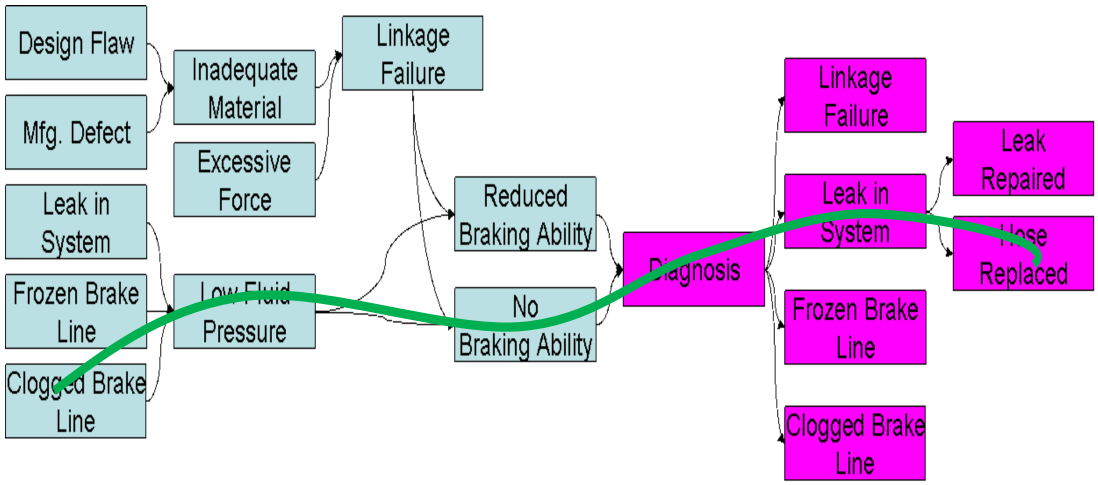

Figure 9 : Advanced FMEA

A warranty event is one unique path in a warranty scenario. Ideally, it should include an initiating root cause that leads to a series of observable effects (at least, in principle) until the observable effects leads to a customer symptom that initiates a warranty call. This then sets off the diagnosis and repair steps that complete the entire warranty event.

For example, in the figure above, the green line represents a single warranty event. Here the root cause is a clogged brake line which leads to low fluid pressure. From a customer's perspective, this problem manifests itself in the form of reduced braking ability which leads the customer to take the car in for service. The problem is diagnosed by service personnel, who discover that there is a leak and subsequently replace the hose.

It is essential to identify the warranty scenarios properly. The process of identifying warranty scenarios requires engineering judgment. In order to create a warranty scenario, however, it is apparent that a set of related failure mechanisms and resolution procedures needs to be developed simultaneously. This process generally involves a series of steps in which the causes, failure modes, effects/customer symptoms, diagnosis, and repair are identified. For the purposes of this work, a product service manual was used to develop the warranty scenario as this document captured all the aforementioned information and captured the most likely steps that the service personnel would used to address the issue as well as the data that would be captured in a service record.

3.2 Data characterization

A major part of this work was to identify and to characterize the different data sources that would be used to model a warranty event. Characterization of datasets entails converting the different data sources into continuous distributions so that they could effectively be integrated into the Bayesian model.

An industrial partner was identified so that real data could be modeled and so that the issues that were identified and addressed during the modeling effort would increase the likelihood that this work would be relevant to the practicing product developer. The data used in the model were field data, engineering judgment data and product development data. Each dataset has unique challenges when converting them into continuous distribution, which will be discussed below.

3.2.1 Field data

Field data are generally considered to be “dirty data”, but they are also the richest dataset in the sense that they represent the failure to meet the expectations of the customer. In addition, they are relatively less difficult to convert into a continuous distribution than converting qualitative data.

The real challenge lies in mining the field data to ensure that the actual warranty event that is being reviewed matches the warranty event that is being modeled. Given the poor descriptions of the system state, the diagnosis and the repair actions that are logged in a service record, this process will also entail some human judgment. Important fields that are used in order to make this assessment include the parts replaced, the initial recording of the customer symptoms, and the field service personnel’s descriptions of the actions taken.

Once the service record has been categorized as a particular warranty event, it is important to determine length of time in service of the system/product when the event occurred. The time need not necessarily be calendar time. It could also be cycle counts, miles, etc. The data are then converted into a failure distribution using the rank regression or the maximum likelihood method.

3.2.2 A brief introduction to maximum likelihood estimation (MLE)

MLE is a standard approach to parameter estimation. It has optimal properties like sufficiency, consistency, efficiency and parameterization invariance. In a probability distribution, the model's parameter is unique; if this parameter changes, different probability distributions are generated (Myung, 2003). So MLE uses the value of model parameter that maximizes the likelihood function. A likelihood function for a set of parameters, given the outcomes, is equal to the probability of the outcome gives the parameter values. The likelihood function is a generally based on the *pdf* of a given distribution.

$$f(x;\ \theta_{1,}\theta_{2,} \dots \dots .\ \theta_{k,}) \dots \dots \dots (1)$$

Where *x* represents the time to failure and $\theta_{1,}\theta_{2,} \dots \dots .\ \theta_{k,}$ are the parameters that need to be estimated. For the entire data the likelihood function is the product of the pdf functions with one element for each data point.

$$L = \prod_{i=1}^{n} f(x_i;\ \theta_{1,}\theta_{2,} \dots \dots .\ \theta_{k,}) \dots \dots \dots (2)$$

Here n is the number of data points that are available. The logarithm of this function is taken and the highest value for this function is estimated.

$$\vartheta = \ln L = \sum_{i-1}^{n} \ln f(x_i;\ \theta_{1,}\theta_{2,} \dots \dots .\ \theta_{k,}) \dots \dots \dots (3)$$

By taking the partial derivative of the log linear equation for each parameter and setting it to zero this highest value for this function can be estimated.

$$\frac{\partial \vartheta}{\partial \theta_j} = 0, j = 1,2,3, \dots k \dots \dots \dots (4)$$

3.2.3 Product development data

Product development data are generated primarily through system-level tests. These datasets are obtained from qualification, functional or reliability growth testing. These datasets are very useful because they give the best idea of how the system will perform once it is in the field. But there are generally two issues in using these datasets. One is that these tests are generally accelerated in time or stress levels, the second is that because of the testing regimen, the failures

exhibited may not be seen in the field since they are not representative of actual customer use patterns.

An accelerated test may not be representative of the length of time in service that would be observed in the field. Thus a conversion factor or correlation relationship may be needed. The other issues with these data are that sometimes the mere fact that they are being conducted in an accelerated manner means that the failures generated do not result from usage patterns that are consistent with actual usage in the field and thus they may never manifest themselves in the field.

3.2.4 Engineering judgment data

Engineering judgment data are another useful data source in product development. Engineers in product development have much collective experience in terms of failure modes, potential design weaknesses and the reliability of different components. This information is particularly useful early in product development, especially in the concept development and selection stages, as there is little direct evidence about the reliability of the product. The engineering judgment is often recorded in design documents like FMEA or can be obtained through interviews and other elicitation methods if they are necessary. The problem with these data is that they are qualitative and they need to be adjusted for the bias of the expert.

Formulation of engineering judgment is done by treating the expert opinion as information about the unknown parameter of interest (Dezfuli, Kelly, Smith, Vedros, & Galyean, 2009). In the example that will be discussed below, the engineering judgment data were coded using a log normal distribution. The log normal distribution was chosen because it can accommodate a variety of data sets and it can encode information about a parameter that varies with several orders of magnitude. The flexibility of using lognormal distribution is shown in Figure 10 where different parameters can greatly vary the shape of the distribution. The mu parameter in the lognormal distribution shows the central tendency of the dataset. The tau parameter in the lognormal distribution is used to assess the judgment. A small value means that the confidence in the estimates is low and vice versa. A bias factor can also be incorporated (not shown in figure), with bias less than one if it is believed that the expert underestimates the reliability, or greater than one if it's believed that the reliability is over estimated (Dezfuli et al., 2009).

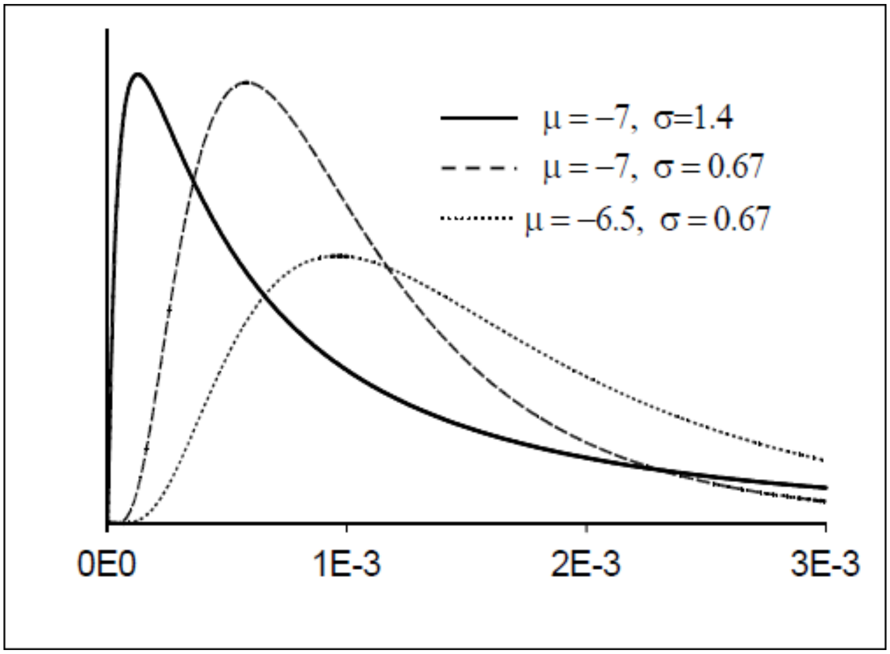

Figure 10 : Lognormal distribution variation with parameters.

3.3 Introduction to Bayes theorem

Once the warranty distributions for the warranty scenario are characterized, the integrated probability model can be developed. The integrated model is developed using the Bayes theorem. The Bayes theorem in its simplest terms is

$$\textbf{Posterior} \propto \textbf{Likelihood x Prior} \ldots\ldots\ldots (5)$$

Mathematically it can be expressed as

$$\mathbf{p(\theta|y) \propto f(y|\theta)\ p(\theta)} \ldots\ldots\ldots (6)$$

Where:

$p(\theta|y)$ is the posterior density

$p(\theta)$ is the prior density

$f(y|\theta)$ is the likelihood

Here y denotes the random variable that constitutes the data and θ denotes the parameter that indexes the family of densities.

The prior is the value of the parameter of interest that is available before analyzing the data. In this work the parameter of interest is the elapsed time to the warranty scenario. This can be specified using a probability density function and is called the prior density. Priors can either be diffuse or informative (Hamada, 2008). A diffuse or non-informative prior reflects the fact that little is known about this parameter. For example, a uniform distribution is a diffuse prior. When the information is contracted in a particular region of the parameter space then it is called an informative prior. In a majority of cases there is some amount of information available that can be used as an informative prior distribution. The information can be in the form of industry wide data, handbooks, experience with similar products and expert judgment.

Likelihood represents the data that have been collected during an experiment. These data could be different forms like a pass/fail data, or times to failure data, etc. and represent our current knowledge about how the product is performing. Then the prior distribution is updated using the likelihood and the posterior distribution is obtained. It is called the posterior as it reflects the probability beliefs after analyzing the data.

Prior distributions that have the same functional form as that of the posterior distribution are called as conjugate prior distributions. Conjugate prior distributions are easy to compute and thus are widely used. But it is important that the prior distribution should not be chosen for the sake of computational convenience but chosen for the adequacy of representation of the information. There are times when it is not possible to have conjugate prior distributions or the posterior distribution cannot be obtained in a closed form. In those cases numerical methods such as Markov Chain Monte Carlo (MCMC) are used to approximate the posterior distribution.

For example (Hamada, 2008) if we have data points $Y_1, Y_2, \ldots Y_n$ which are random samples from a normal distribution with parameters $N(\mu,\sigma^2)$,the non informative prior for that is

$$\pi(\mu,\sigma^2) \propto \frac{1}{\sigma^2}$$

Where μ and σ^2 are the parameters of the normal distribution.

The likelihood of the normal distribution for a given set of data points will be

$$\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i-\mu)^2}{2\sigma^2}\right)$$

The joint posterior is obtained by multiplying the prior and the likelihood

$$\pi(\mu,\sigma^2|y) \propto (\frac{1}{\sigma^2})^{\frac{n}{2}+1}\, x \exp\left\{\sum_{i=1}^{n} -\frac{(y_i-\mu)^2}{2\sigma^2}\right\}$$

Here the posterior distribution contains two parameters μ *and* σ^2.Since there are two parameters, inference on one of the parameters is difficult. So the other "nuisance" parameter needs to integrated out of the posterior distribution (Hamada, 2008). Though it is possible to integrate out this parameter in this case analytically, this can get very complicated when there are more parameters or if the posterior distribution is not of closed form. So in order to avoid this, MCMC methods are used.

3.4 Brief introduction to MCMC

Markov Chain Monte Carlo (MCMC) algorithms are computational methods which produce samples from posterior distributions. They can also be used to implement high dimensional posterior distributions.

Monte Carlo methods are computational algorithms which use random sampling to produce numerical results. They are useful in situations where the problem is too complicated to be solved analytically, and one of the most common applications of these methods is in integration of complex integrals.

A sequence of random elements of some set is called a Markov chain if the conditional distribution of X_{n+1} is dependent on the previous point only. Also a Markov chain has stationary transition probabilities if the conditional distribution does not depend on n. The marginal distribution of X_1 is called the initial distribution and the conditional distribution of X_{n+1} given X_n is called the transition probability distribution (Gilks, Richardson, & Spiegelhalter, 1995).

Another characteristic of Markov chains is infinite state space. So as the time tends to infinity in the simulation a Markov chain converges to its stationary distribution.

So essentially in MCMC, a Markov chain is created having specific properties of the underlying distribution. Once these chains are created, a random walk happens inside the space using Monte Carlo methods. After we start this process of random walk and after a certain period of time, these algorithms produce independent samples of the posterior distribution indicative of the underlying distribution.

There are different implementations of MCMC like Metropolis & Hastings algorithm, Gibbs sampling, slice sampling etc. For the purposes of this work, we use the Gibbs sampling methodology implemented by the BUGS language. The software package WinBUGS is an implementation of the language.

3.5 Integration of datasets

One possible example of a data integration strategy is shown in Figure 11. In this example, the product development data and engineering judgment data of a current generation product is combined with field data of the previous generation product. The current generation product represents the product that is under development and the field data, N, represents the actual field experience of the current generation.

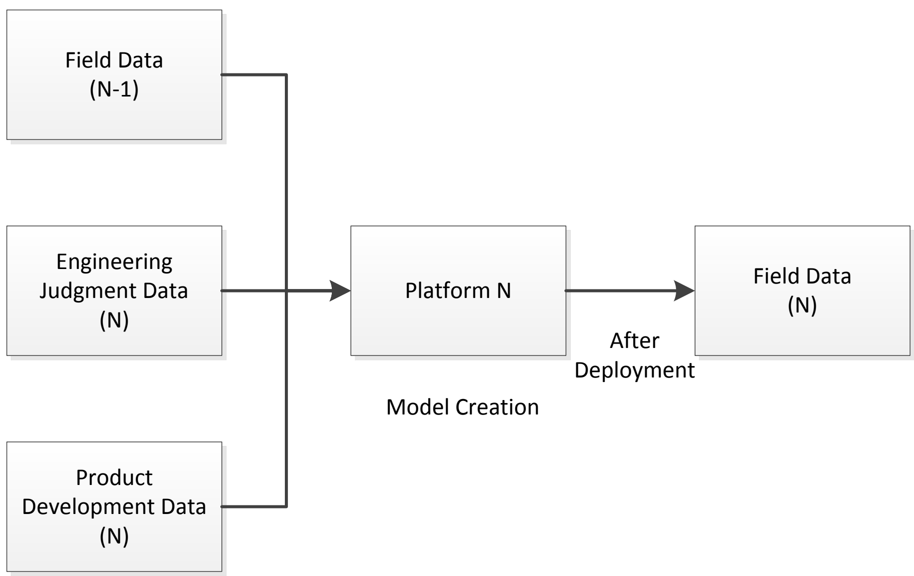

Figure 11 : Data integration approach

One question that was of interest in this work was the appropriate strategy to use to integrate the three data sources on the left, to generate a prediction of the actual field experience. In particular, which data sets should be integrated in which order? Furthermore, which data sets should be modeled as the priors and which data sets should be modeled as the likelihoods? Chronologically the datasets are available at different points in time. Expert or engineering judgment data is available as early as concept design and product development data is obtainable once the system integration is started and hardware is available to start generating testing data. Manufacturing data is available once the production processes are ramped up to product volumes. Field data starts arriving as soon as the product is launched. But the data stabilizes only

after a certain point in time. Initially there might be minor issues which cause service calls but they are not issues that are typically seen in the long term.

To address these questions, the model shown in Figure 12 was used to incorporate the different datasets. In order to accommodate more than two datasets, a multi stage Bayesian model was developed and this would use all of the available information in a sequential order. So the first two datasets are used as prior and likelihood which would give a posterior distribution. This posterior distribution can then be used either as prior or likelihood along with another type of data. After the model was developed, the posterior distribution of the last stage is analyzed. This posterior distribution will give the times to failure of the warranty scenario. This distribution can then be compared to the actual field data to check the model performance.

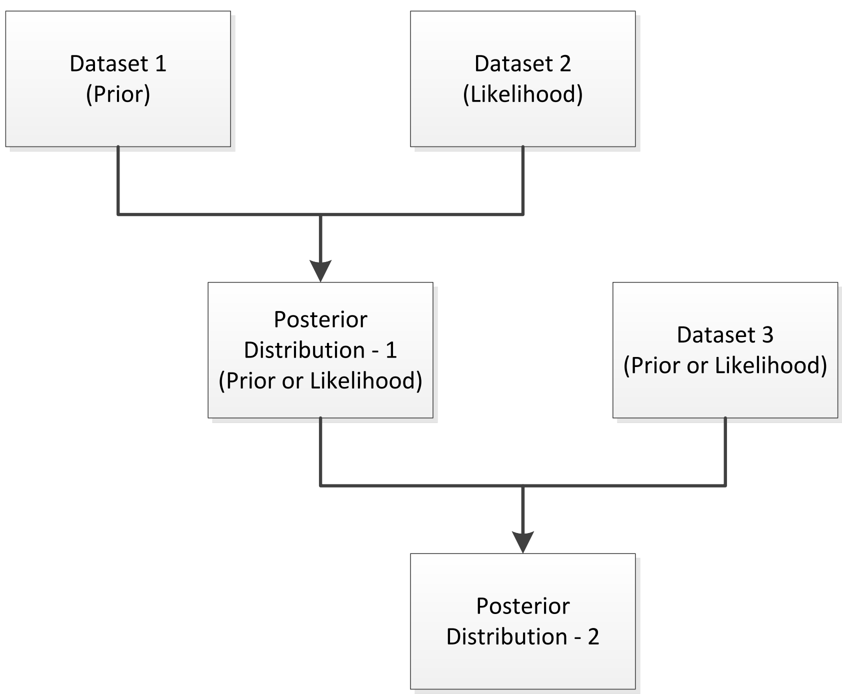

Figure 12 : Two stage Bayesian model

At the outset of the model development, it was hypothesized that the following integration order would yield the best results. The engineering judgment data is the first available data, and once the product testing starts the data that is collected will be used as likelihood and the engineering judgment would be the prior. As the field data stabilized it would subsequently used as

likelihood in the second stage. However, since the literature warns that the selection of the prior and the selection of the likelihoods is not always a straightforward proposition, it was decided to run all combinations of orders as well as switching the orders of priors and likelihoods.

An overview of the steps that were performed is listed below.

1. The likelihood distribution was created using the parameters of the dataset. This simulated the current knowledge of the product.
2. Using the data points derived from the likelihood distribution as the scale parameter and using the shape parameter of the prior, the samples for the posterior distribution were generated.
3. The initial 1000 points are removed for convergence because the distributions under consideration are relatively simple and thus 1000 iterations are sufficient for convergence (Dezfuli et al., 2009). The other points obtained are then converted into an approximate distribution, and their distribution parameters are calculated. This becomes an input for the next level.
4. In the next level, depending on the specific case, the derived posterior is either used as a prior or likelihood. The third data set complements this derived posterior as the prior or likelihood.
5. The resulting samples of the posterior distribution are then used to form a time to failure distribution plot.
6. These plots can be used to understand the life distribution of the model and compared to the original field data.

This will be further discussed in the results section.

4 Results

4.1 Identification of warranty scenarios

The warranty scenarios were created from the product service manual. Choosing a particular warranty event for this work contingent upon specific factors such as:

The event should have adequate representation in the field data

- Since the field data that were obtained did not have any free form fields which would give further explanation of what the problem was we had to rely on the parts that were replaced during the service event to understand what the possible warranty event was. Since different problems could have the same part replaced, particular care was given to choose a warranty event which had specific components replaced that were unique for the product.
- It was also important that the particular warranty event was adequately represented in the product development test data set and the engineering judgment data set as well so that the warranty distribution could be determined.

One such generalized warranty event that we chose for this work is shown in Figure 13. The condition code and other information are described more robustly on the product, but for illustration purposes, this has been simplified. If the machine flashes the conditions codes, field service is notified and the service personnel is dispatched to resolve the issue. Once the field tech is at the site, he looks for the symptom, which is in this particular case "is the channel from reservoir is free of material?" If this was not the case then Parts ABC and XYZ are replaced. This part combination replacement is unique for a particular scenario and does not occur anywhere else in the product.

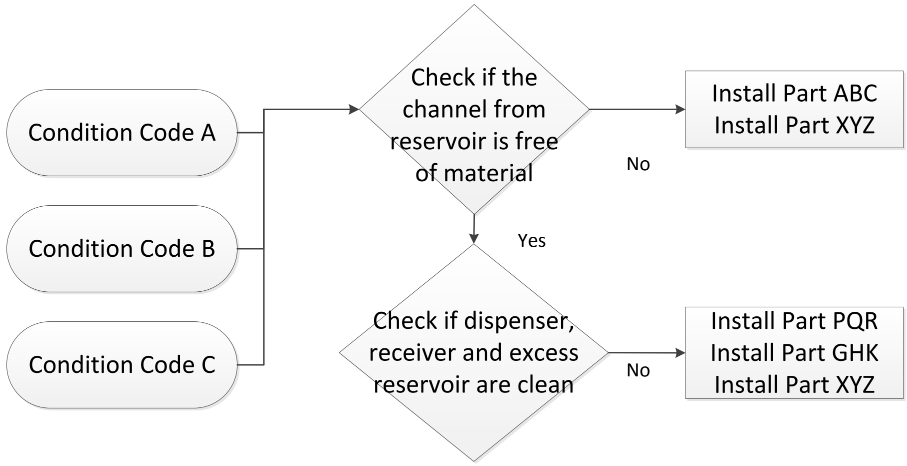

Figure 13 : Warranty scenario created from the service manual

4.2 Data characterization

The data sources that were used in this work came from two different generations of products as shown in Figure 14. The field data from the first generation was used along with the product development and engineering judgment data from the second generation. The data was collected from our industrial partner and was used throughout our work. The unit of measurement was calendar time, i.e. days to warranty event.

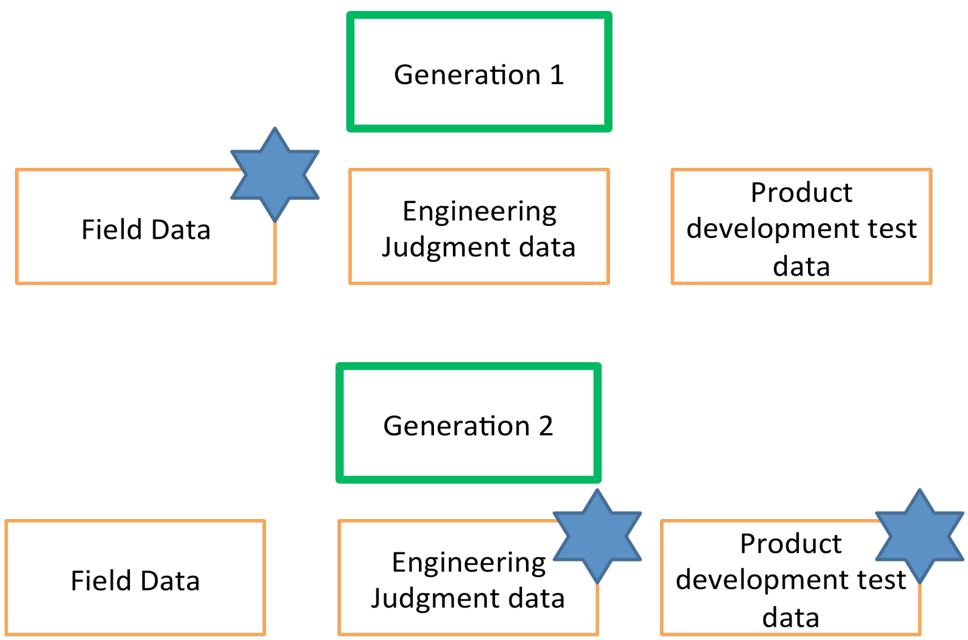

Figure 14 : Datasets used for the model

4.2.1 Field data

The field data had specific information pertaining to the service call such as the date, parts replaced during the service call, machine serial number etc., but it did not have any free form data field where the service personnel could enter any comments about the service call. So the information from field data could be converted into a warranty event only by counting the parts that were replaced together.

This specific warranty event consisted of replacing two components at the same time. So all such instances were counted and summarized as shown in Table 1. For example the first instance in the table shows that the particular machine was installed on 9/21/2009 and the event (i.e. replacement of the two components) happened two days after install. The corresponding cycle count is also shown in the table.

Table 1 : Field data example

Serial No	Date of Service Event	Cycle count	Install Date	Time to warranty scenario (in days)
0001	9/23/09	1837	9/21/09	2
0002	4/22/11	100	4/19/11	3
0003	3/31/10	308	3/22/10	9
0004	3/9/10	21864	2/26/10	11

There were also instances where the warranty event happened twice during the course of machine's lifetime. In that case, for the first occurrence, the difference between install date and event date is taken. For subsequent occurrences the difference between first occurrence date and second occurrence date is counted as time to warranty event.

There were also some unique problems that occurred when the data analysis was performed. It is quite possible that, sometime before this event had happened, either of the two components could have been replaced previously. While this scenario did not happen very frequently, there were some instances of this happening. These cases were discarded as they could pose problems when counting the time to warranty event.

The field data also had some other inconsistencies like a service event date was prior to install date and also some of the machines did not have an install date. Again these instances were discarded for the sake of simplicity.

The failure data in the table only shows the instances of this event happening. So that means the rest of the population did not see this event in the field. This is important point to consider as this describes the case of suspensions. The inclusion of suspension data gives us the true knowledge of the times to warranty event for this particular event. There were thousands of machines which did not exhibit this event. Since field data was available only until Jan 31, 2012, this was considered as the cutoff date. So for all the machines that did not have this event, the difference between the install date and this cutoff date was calculated as suspended time. This data was combined with the failure time to create the life distribution data for the field data.

The next step was to fit the data into an appropriate distribution. Weibull++ was used to perform this step. Weibull++ has the capability to fit various distributions to the data and specify which distribution is the best fit for the data. This is performed by using the distribution wizard tool as shown in Figure 15. After this analysis, it was found out that log normal distribution was the best fit for the given data and the parameter for the distribution was evaluated.

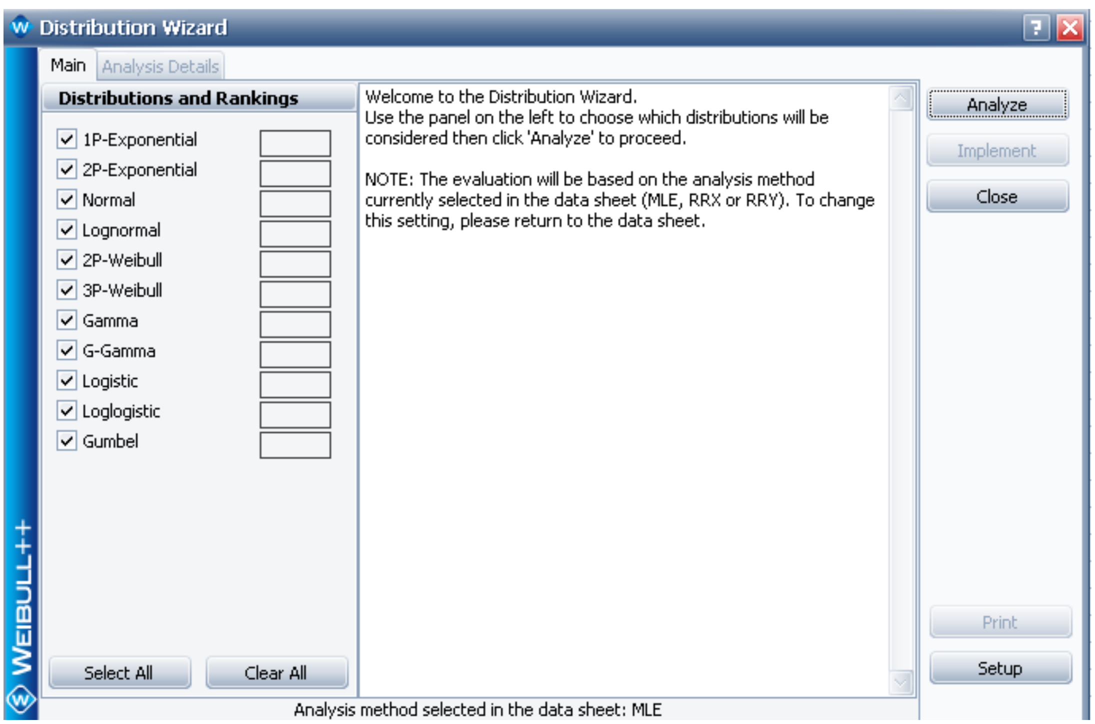

Figure 15 : Distribution wizard

Then the times to warranty scenario was used to calculate the failure distribution using the maximum likelihood method. The field data followed a lognormal distribution with the following parameters; Log-mean: 4.69 years and Log-SD: 2.01 as shown in Figure 16.

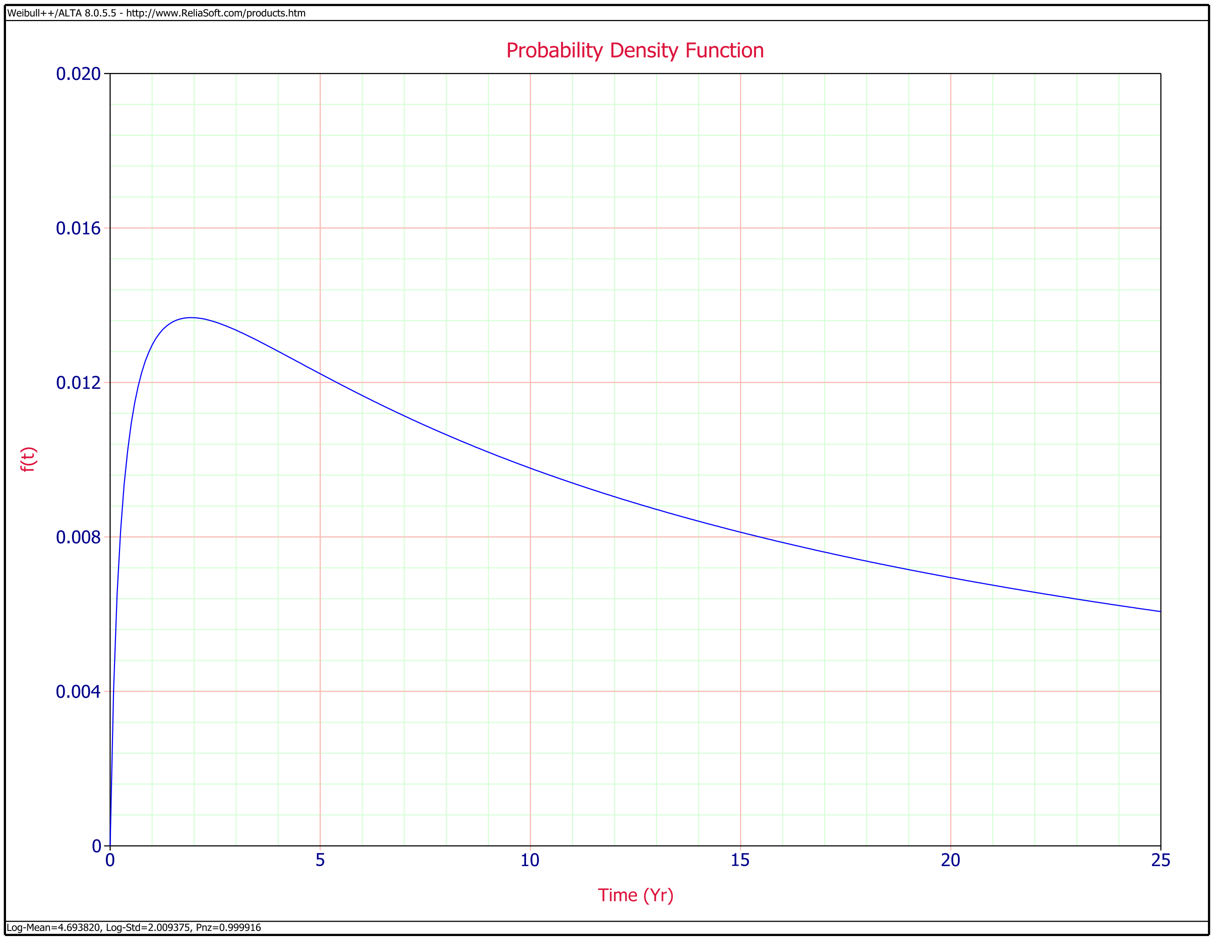

Figure 16 : Life distribution of field data

4.2.2 Product development data

The product development test data were obtained from the reliability growth testing data that was available for the product. These data include all the failure modes that happened in time along with other information like the cycle count, the date, and time that this failure mode occurred, the condition code that was observed, and the action that was taken to resolve the failure. In order to relate the test accelerated usage to the field usage, a simple linear model was created using the field data of the first generation product as shown in Figure 17. The model is a plot of normalized cycle count and calendar time. This was then used to create a linear equation, which would give the calendar time for a particular cycle count. This method had a very good fit and the R^2 was found to be 0.92. Now the calendar time for the warranty event in the product development data was computed using the model.

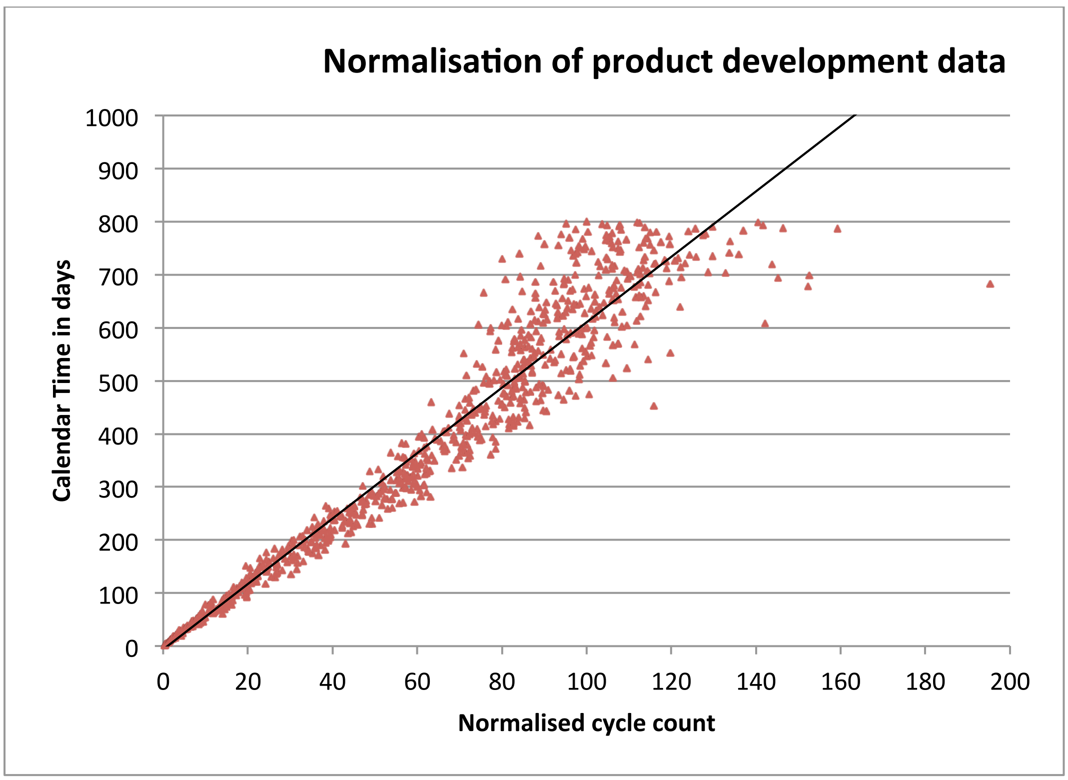

Figure 17 : Normalization of product development data

Then failure distribution using the maximum likelihood method using Weibull++ was found and this data followed a two parameter Weibull distribution with the following parameters; Beta (shape) = 1.749 and Scale = 21.469 years and is shown in Figure 18

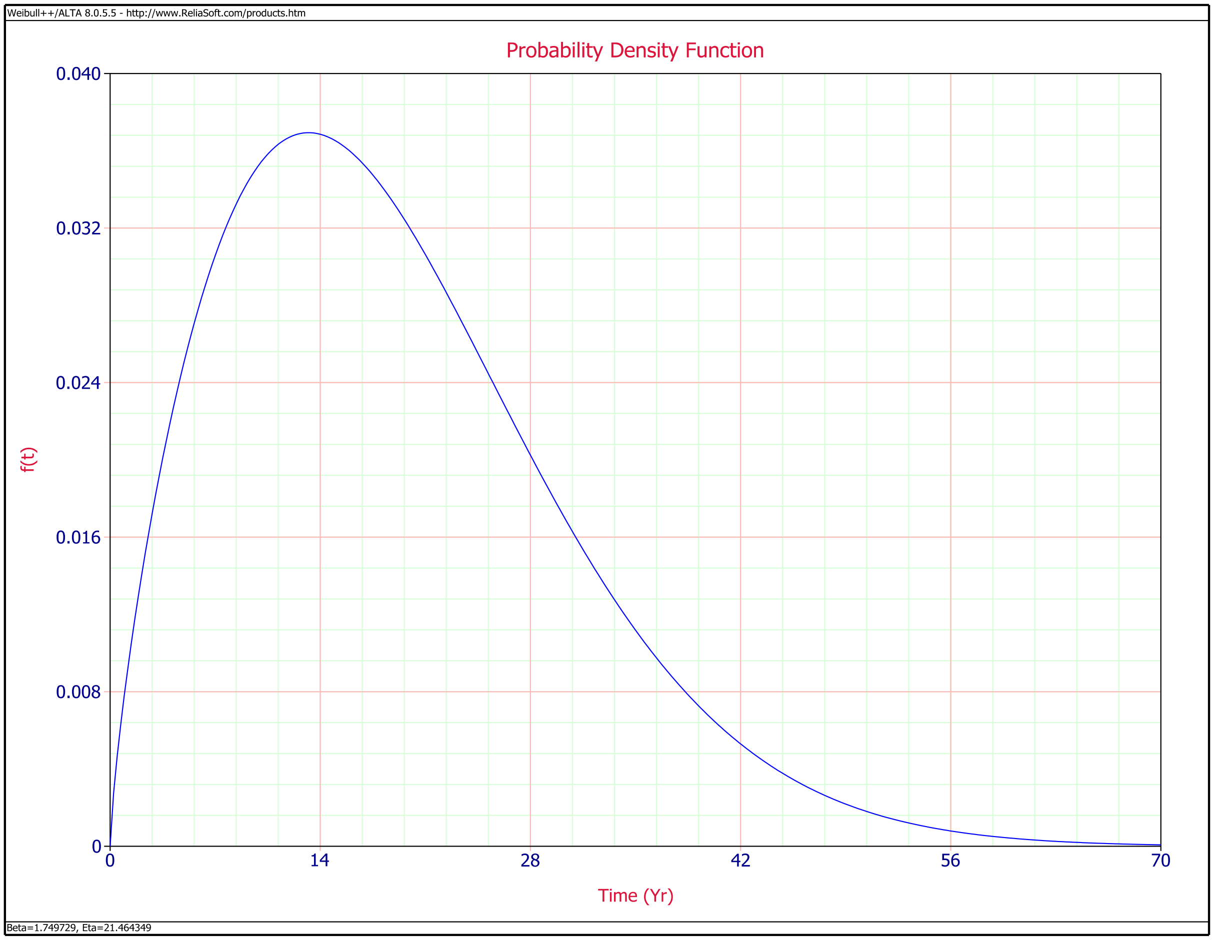

Figure 18 : Life distribution of product development data

4.2.3 Engineering judgment data

The engineering judgment data was obtained from the FMEA that was available for the product. The FMEAs had an occurrence rating for the particular event. This occurrence rating was then converted into calendar time using the occurrence rating scale that was used in the FMEA. For this particular event, the probability of occurrence for this event was determined as 1 in 10000 days. So this was converted to a time basis on a log scale and was used as the mean for the log

normal distribution. The error factor or tau was assigned 5 because of the uncertainty in the estimate. The factor was then used to create the SD for the lognormal distribution.

Thus the engineering judgment data had a lognormal distribution with log mean of 1.43 and log SD of 0.978 and life distribution is shown in Figure 19

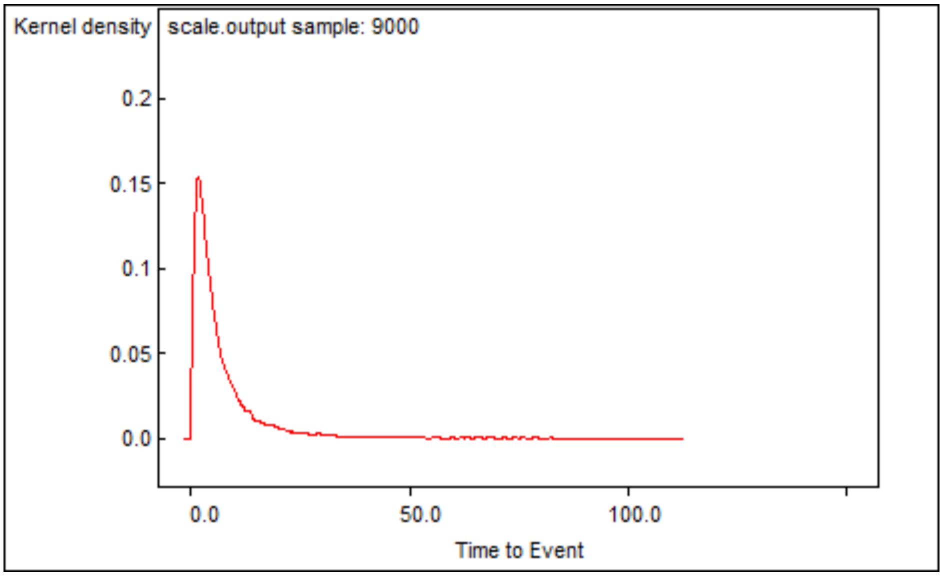

Figure 19 : Life distribution of engineering judgment data

4.3 Integration of datasets

Since there are three different data sources, the issue which needs to be addressed is the integration order and which distributions are to be used as priors and which are to be used as likelihoods. Since there is no compelling evidence as to which distributions work better as priors or likelihoods, it was decided to conduct a full combinatorial experiment with all three datasets. We used a two stage Bayesian model for this experiment, as described above in Figure 12. In the first stage two datasets were used as priors and likelihoods and the resulting posterior was used as likelihood, if the third dataset was used as prior and vice versa.

This approach resulted in a total of 12 different cases to examine in order to understand which combinations performed better. The twelve cases are shown in Table 2

Table 2 : All experimental combinations

Case No	Initial Prior	Likelihood	Resulting Posterior→ New Prior (For Next Stage)	New Likelihood
1	Product Development Data	Field Data	Function of Prior and Likelihood combined using MCMC	Engineering Judgment
2	Field Data	Product Development Data	Function of Prior and Likelihood combined using MCMC	Engineering Judgment
3	Product Development Data	Engineering Judgment	Function of Prior and Likelihood combined using MCMC	Field Data
4	Engineering Judgment	Product Development Data	Function of Prior and Likelihood combined using MCMC	Field Data
5	Field Data	Engineering Judgment	Function of Prior and Likelihood combined using MCMC	Product Development Data
6	Engineering Judgment	Field Data	Function of Prior and Likelihood combined using MCMC	Product Development Data
1x	Product Development Data	Field Data	Engineering Judgment	Function of Prior and Likelihood combined using MCMC
2x	Field Data	Product Development Data	Engineering Judgment	Function of Prior and Likelihood combined using MCMC
3x	Product Development Data	Engineering Judgment	Field Data	Function of Prior and Likelihood combined using MCMC
4x	Engineering Judgment	Product Development Data	Field Data	Function of Prior and Likelihood combined using MCMC
5x	Field Data	Engineering Judgment	Product Development Data	Function of Prior and Likelihood combined using MCMC
6x	Engineering Judgment	Field Data	Product Development Data	Function of Prior and Likelihood combined using MCMC

To create the two stage Bayesian model, we used the MCMC algorithms to sample the posterior distribution. This was primarily done so that there is flexibility in choosing the distribution of the

different datasets. It meant that we were not constrained to use a conjugate prior distribution and we were able to choose the distribution which represented the data best. For creating the MCMC we used the WinBUGS computational platform. WinBUGS is a popular MCMC software that uses the BUGS language, and an implementation of the Gibbs sampling methodology. The BUGS code for all the twelve cases are presented in the Appendix.

The complete BUGS code for Case 1 is shown below, and reader will be walked through all the steps that were performed in arriving at the final results

Case 1, Stage 1

```
model {
scale.posterior ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.output)),shape.pd)

scale.output ~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- mu.field
tau.analysis <-1/pow(field.sd,2)
}
data
list(mu.field=4.69,field.sd=2.009,shape.pd=1.73)  # pd prior, # field likelihood
```

4.3.1 Steps to create the model in WinBUGS

1. The WinBUGS code is entered in to an editor. Then from the tool bar Model ➔ Specification is chosen. The dialog box as shown in Figure 20 appears. Here the model is checked for errors and then the data is loaded. The model is then compiled and initial values for MCMC are either specified or generated automatically. There is also a provision to choose the number of chains that can be used in the model. For this research only one chain was used. Also the initial values were automatically generated.

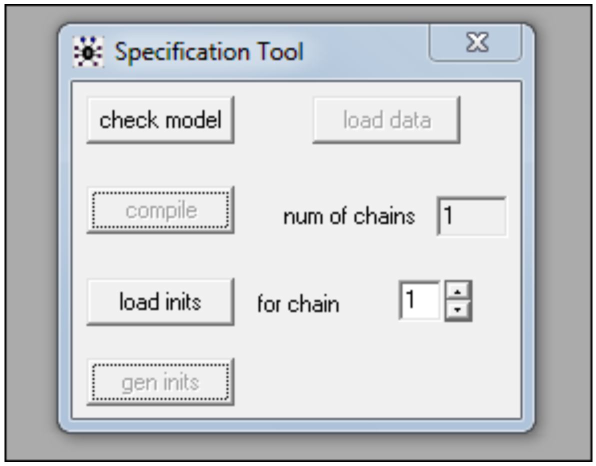

Figure 20 : Specification tool dialog

2. Then from the toolbar Inference ➔ Samples is chosen and a dialog box as shown in Figure 21 appears. In the node section, the parameter that needs to be monitored was entered. From the sample code shown above, "scale.output" and "scale.posterior" is monitored. Here it is also possible to specify the summarized results of the posterior we wish to see. It is also necessary to specify which points we need to monitor for the posterior distribution. The initial points from the simulation are discarded for convergence and only points starting from 1001 are included for consideration. Throughout this research all the simulations were run for 10000 points, discarding the first 1000 for convergence.

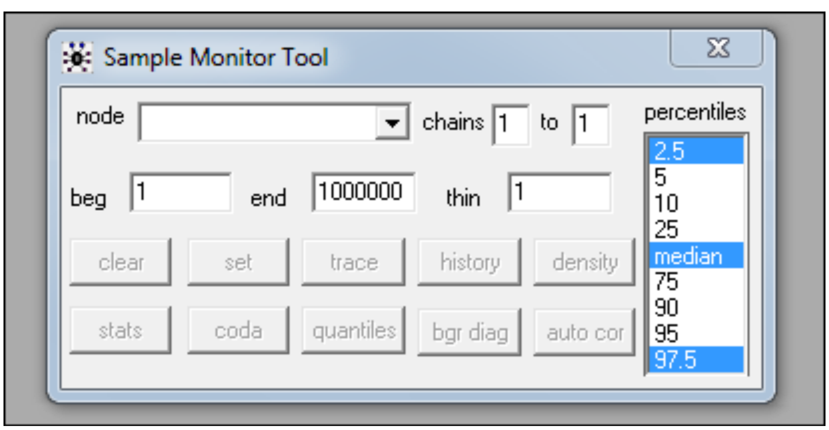

Figure 21 : Samples monitor dialog

3. Once the parameters that need to be monitored are specified, the next step is to simulate the model and create a posterior distribution samples. This is accomplished by selecting Model ➔ Update from the toolbar. The dialog box as shown in appears. Here it is possible to specify the update frequency for the simulation. Higher values will result in slower the simulation time.

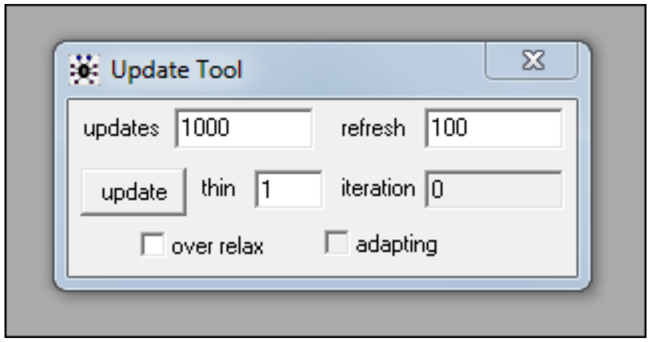

Figure 22 : Update dialog

4. Once the simulation is completed, it is possible to view the various statistics of the simulation. In Figure 23, we can see the mean, standard deviation and some percentile values of the output.

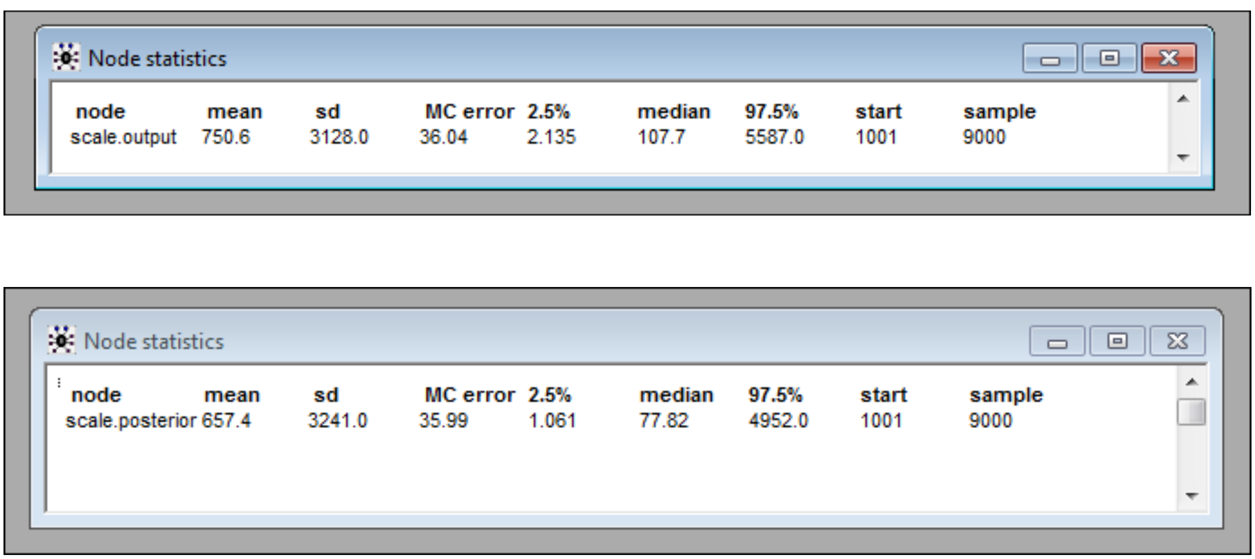

Figure 23 : Node statistics

5. The simulation plot which plots the kernel density vs. values plot shown in Figure 24 can be viewed by clicking on the "Density" button shown in Figure 21. These plots can give a basic idea of the distribution family the resulting posterior belongs to.

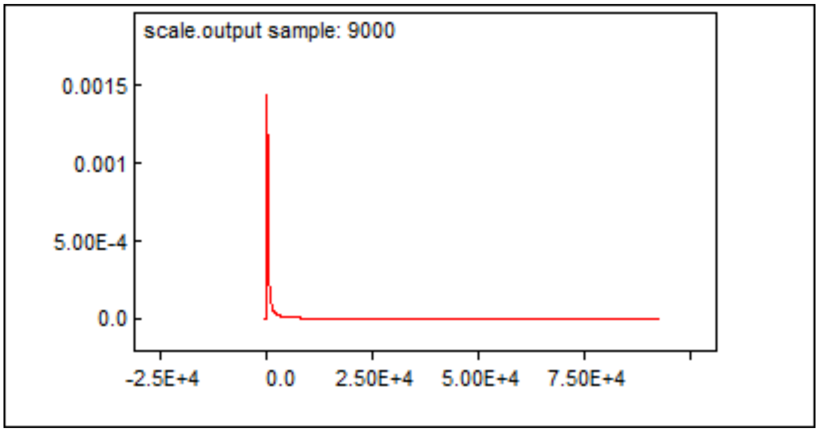

Figure 24 : Output

6. The values that were generated at the end of simulation can be extracted by looking at the Convergence Diagnostics and Output Analysis (CODA) values. The most important output from this stage one is the value of "scale.posterior". This is the resulting posterior distribution from Stage 1. The CODA values are converted into a lognormal distribution by estimating the log mean and log standard deviation of the values. This can be accomplished in MS Excel or can be determined from the output shown in Figure 25Figure **23**. This distribution output serves as the input to the second stage.

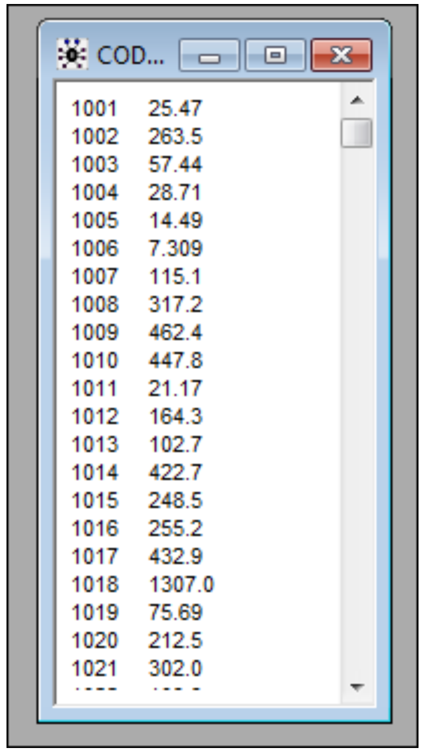

Figure 25: CODA Values

The code used for the second stage is shown below.

```
# Case 1 Stage 2
# Results of Stage 1 lognormal Distribution: Mean 4.34 SD 2.143
# Stage 2 uses stage 1 results as prior and engg judgment as likelihood

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.output3 <- log(scale.output2)
scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- mu.enggjd
tau.analysis2<-1/pow(sd.enggjd,2)
}
data
list(mu.enggjd=1.73,sd.enggjd = 0.97,stage1.sd = 2.143)
```

7. As described in the previous steps, this code is entered into the editor and all the steps previously mentioned are performed. The important output of this stage is the kernel density plot and the CODA values of *scale.posterior2.* This is the final output of the first case. The resulting posterior distribution's kernel density plot is shown in Figure 26.

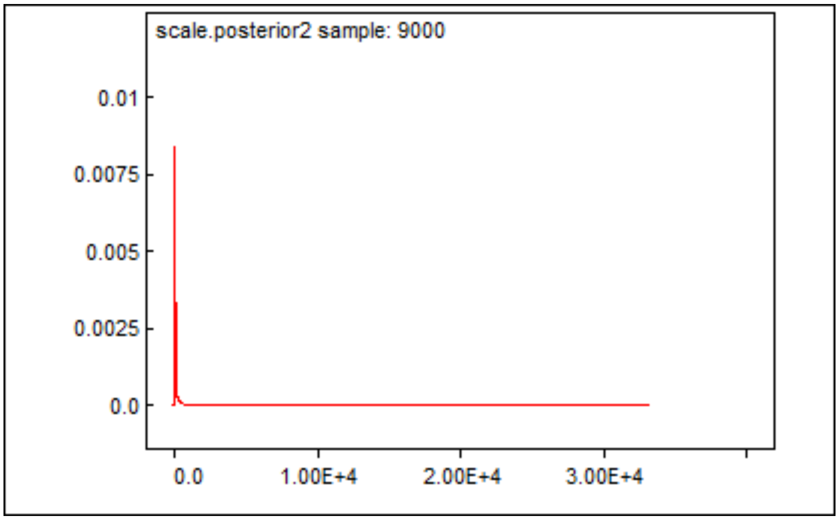

Figure 26 : scale.posterior2 kernel density plot

8. The CODA values are then fed into an Excel file and the values at various percentile ranks are determined as shown in Figure 27.

Percentile Ranks	2nd Generation Field	2nd Generation 95% lower Confidence bounds	2nd Generation 95% Upper Confidence bounds	Case 1	Case 1x
2.5	3.54	0.41	3.91	0.06	0.80
5	11.16	0.80	28.21	0.12	1.68
7.5	23.20	1.22	99.53	0.19	2.61
10	40.84	1.69	263.65	0.28	3.92
12.5	62.35	2.16	546.40	0.40	5.19
15	89.07	2.65	1009.55	0.51	6.70
17.5	125.83	3.24	1830.60	0.63	8.44
20	174.42	3.91	3211.60	0.80	10.52
22.5	234.11	4.63	5332.40	0.98	12.78
25	317.58	5.52	9015.50	1.17	15.63

Figure 27 : Snapshot of the spreadsheet file

All these aforementioned steps were performed for all the twelve cases and the results were tabulated. The results were then converted into a log scale for easy representation.

There were some instances where the output generated an error. It was because a single out of bound value was generated by the simulation. This was easily addressed by removing the outlier and recalculating the percentile values.

Once the results for all the twelve cases were computed, the next step was to compare the results to the actual field data to understand how well the model performed. For this, the field data was converted into a continuous form using Weibull ++ as previously described. The field data followed a lognormal distribution with the following parameters; Log-mean: 8.1 years and Log-SD: 5.69. Then the 95% upper and lower bound parameters for the distribution was also calculated. These three life distributions were simulated in WinBUGS and their CODA values were used to create a life distribution and captured in the table shown in Figure 27. The values were then plotted on a log scale as shown in Figure 28.

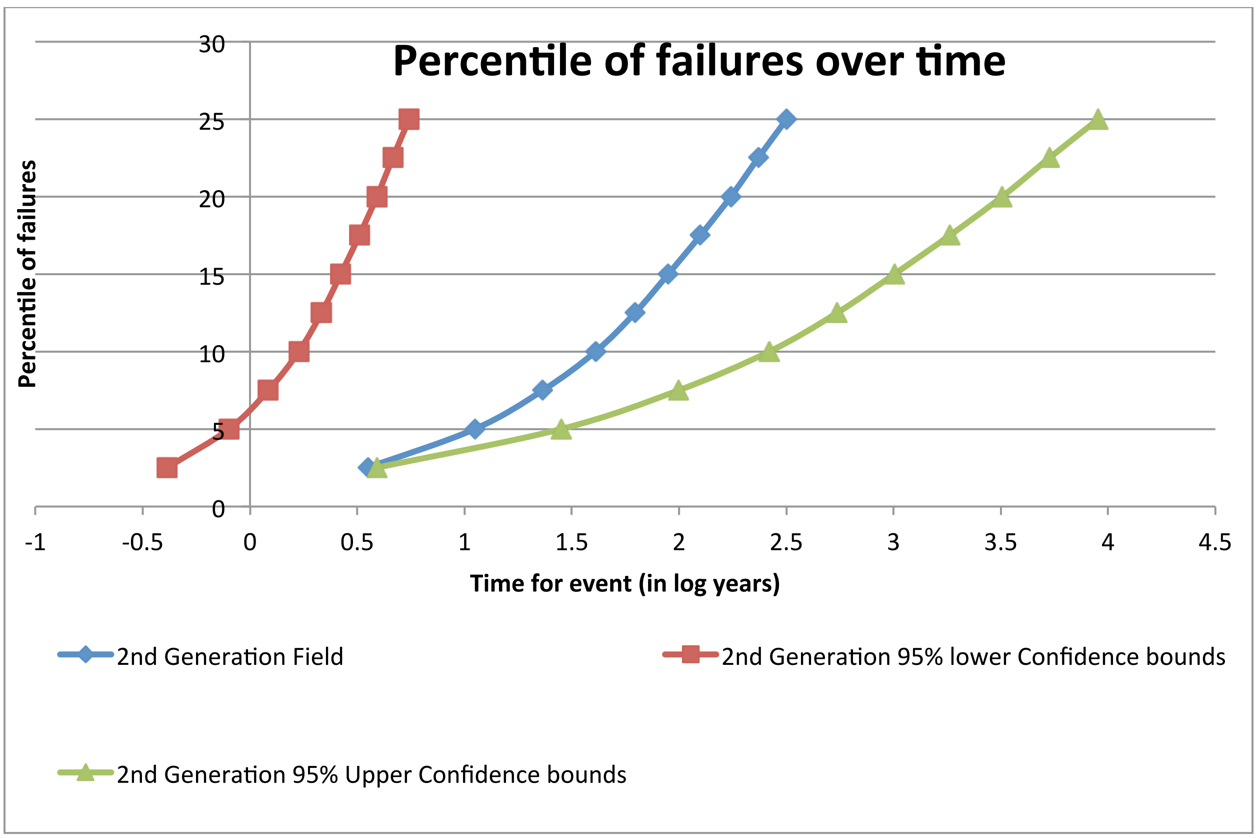

Figure 28 : Field data life distribution with confidence levels

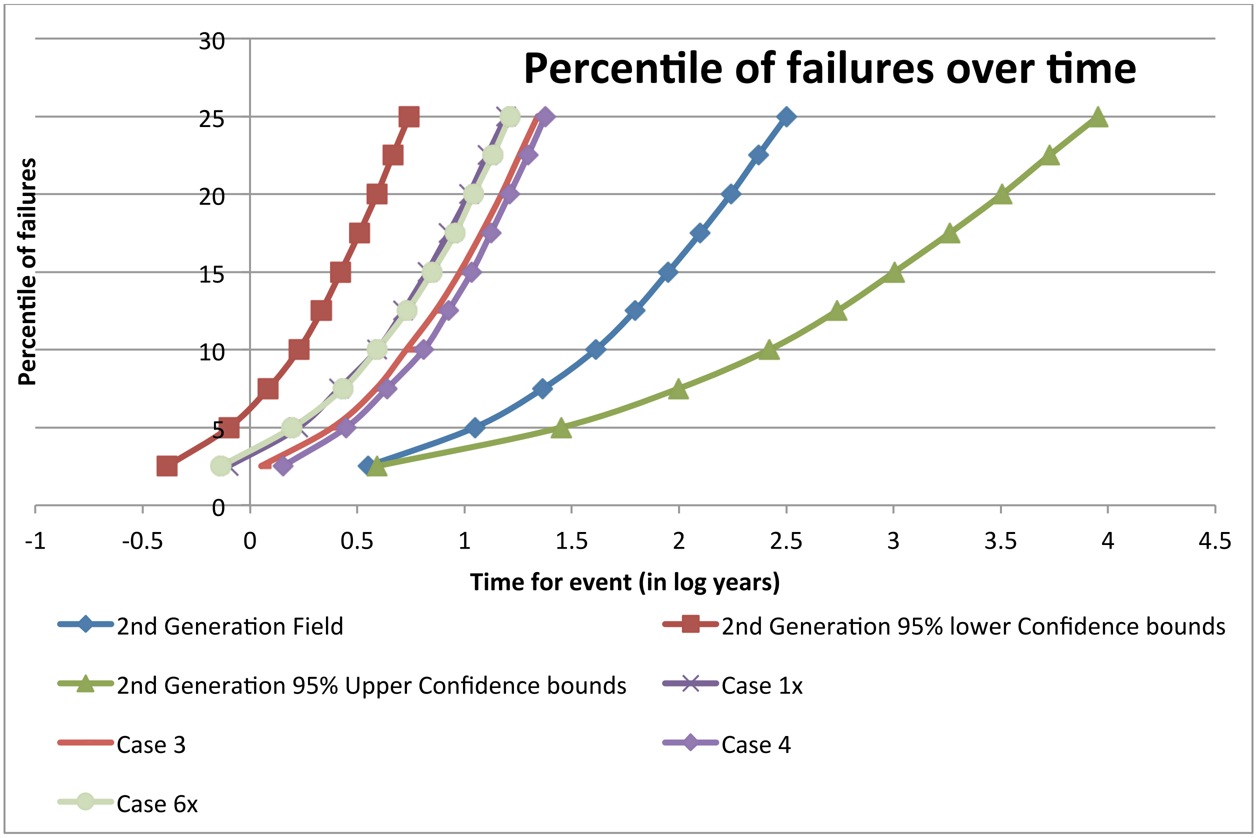

Figure 29 shows the experimental results of the field data and 95% upper and lower confidence bounds of the field data along with the four cases which fell between the lower confidence bound and the actual data. The four cases were 3, 4, 1x and 6x.

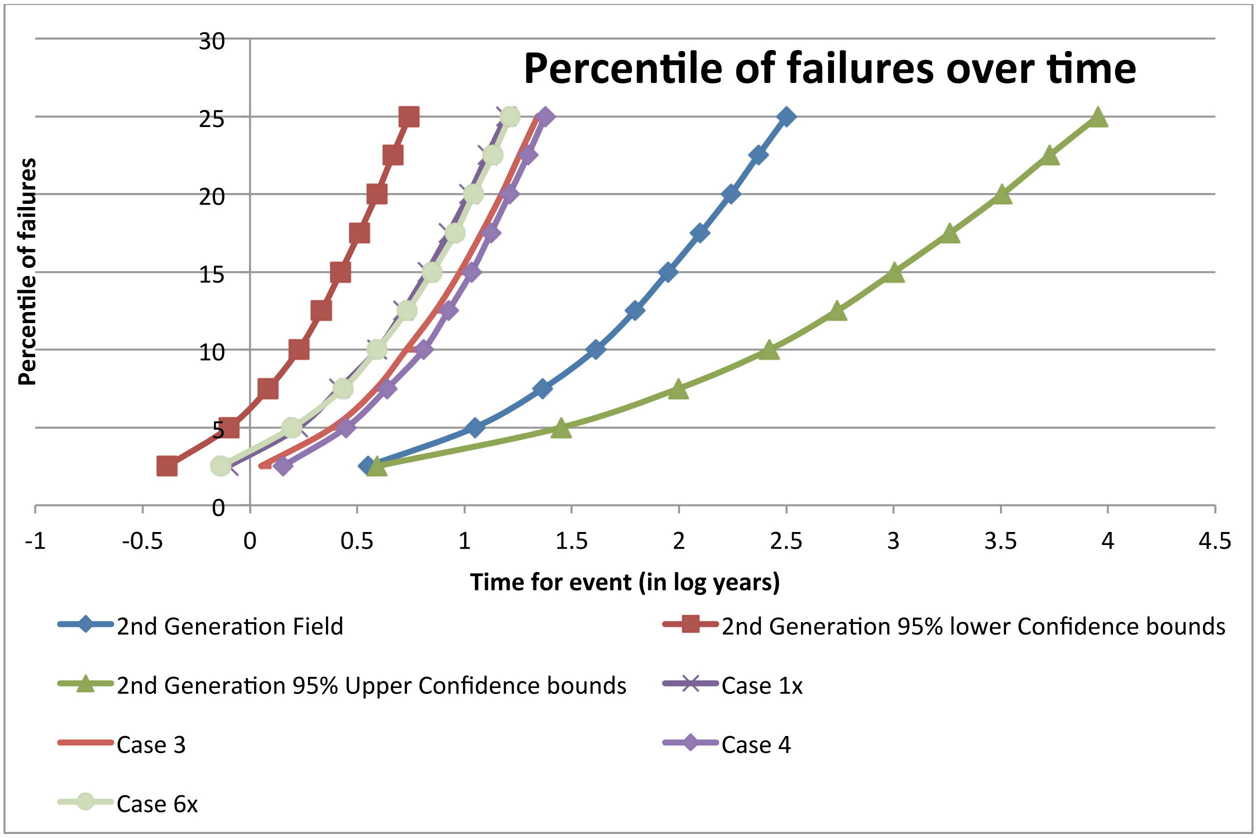

Figure 29 : Summary of results with four cases

The cases which performed well, i.e. provided estimate close to the actual field performance were those cases which had field data as their likelihoods in either stage one or stage two. We know that likelihood is the present knowledge about the system in consideration. So the posterior distribution generally provided more weight to the likelihood and uses the prior to guide the posterior. Our hypothesis is that since this a derivative product development project, it is not surprising these cases did well in the model from the stand-point that there is reason to believe that the follow-on product would follow similar patterns. However, what would not necessarily be expected is that this pattern would be adequately modified by the data generated in product development and the FMEA so as to fall within the 95% confidence intervals of the actual field experience.

Cases 1, 2 5x and 3x had engineering judgment data in their likelihood and is placed leftmost in Figure 30. This is possibly due to the qualitative nature of the data and difficulty in converting them into a continuous distribution. It is also possible that the design engineers over estimated

the effectiveness of the fix in the second generation, but since there was not enough data to suggest there was any bias, inclusion of bias factor into the distribution was ignored.

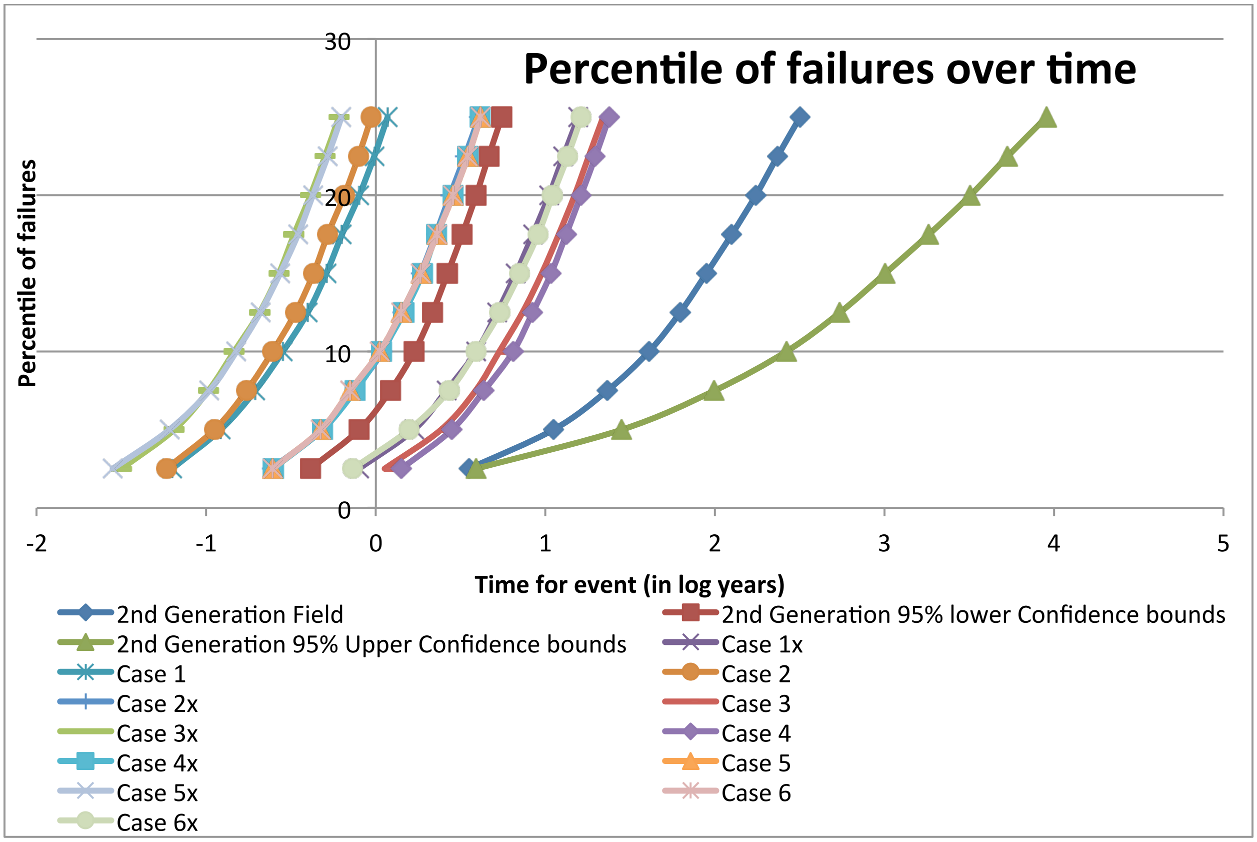

Figure 30 : Summary of results with all cases

The other four cases 5, 6, 4x and 2x had product development data in their likelihood clustered together just to the left of the lower confidence bound. Looking more closely at these results, it is observed that there is a large magnitude in difference between the upper confidence bound and the actual field data. The actual plot without conversion to log scale is shown in Figure 31.

Here it could be seen that there is a huge difference in the scales. It is hypothesized that this is due to the lack of data points in the field data. The distribution that was obtained for field data had very few failure events and large amount of suspension and this could be one reason. The other reason is based on previous experience; it generally takes some time, generally one year or more for the field data to stabilize. The second generation product was in the field only for seven

months when the data was collected. So it is possible that the confidence bound will get narrower once enough data is collected.

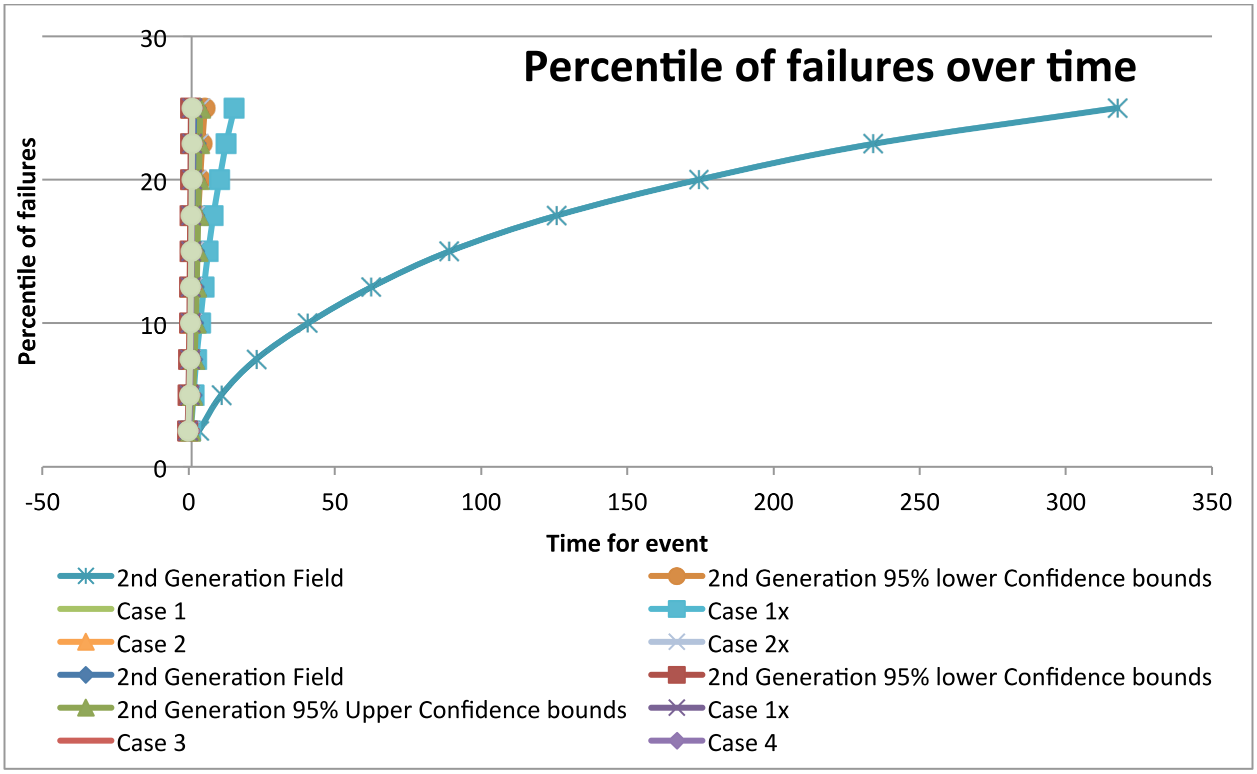

Figure 31 : Summary of results in normal time

One other point of interest was how well the model performed with respect to the first generation field data. The results are shown in Figure 32. From the figure it can be seen that first generation field data actually perform better than the model's prediction. Thus, the model predicted that the warranty event would become worse in the second generation.

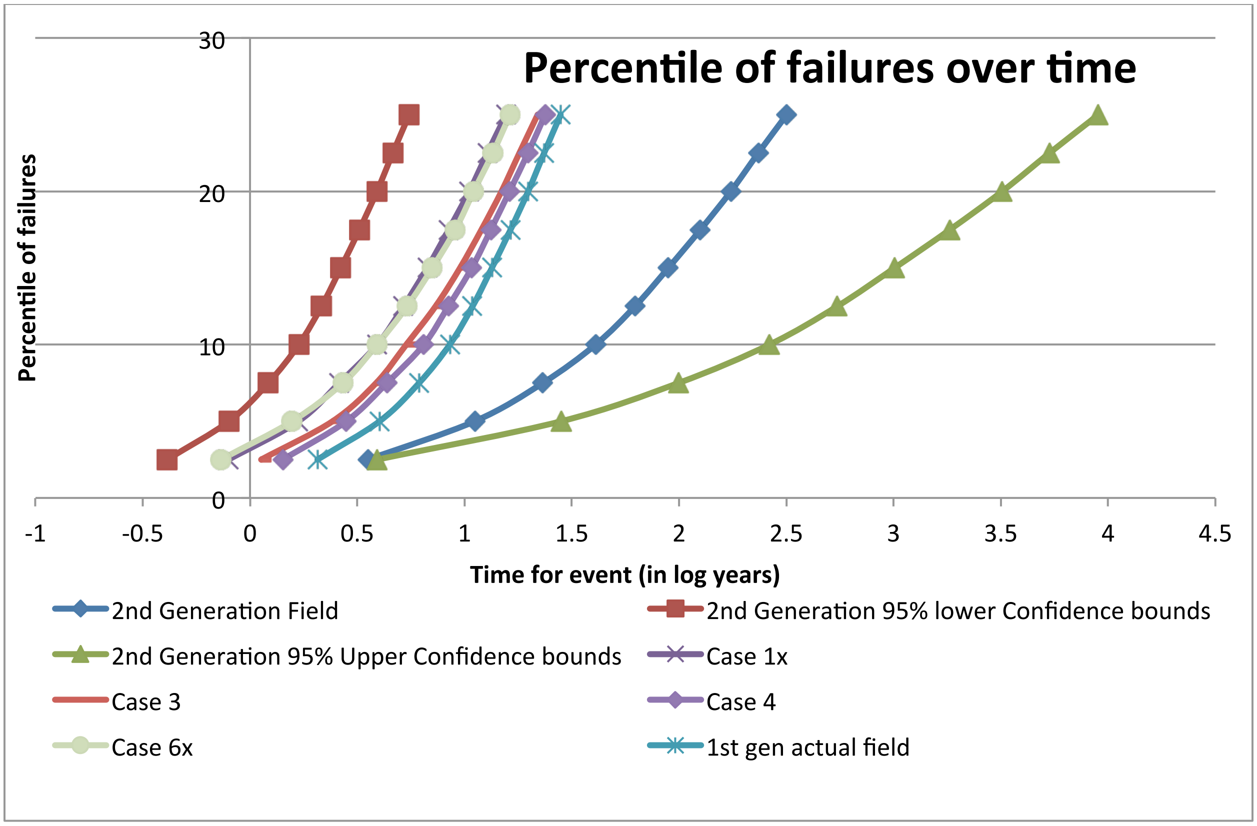

Figure 32 : Results compared to first generation field data

We contacted our industrial partner to understand the significance of the results. They said that after the data for this research was collected, the failure rates for the event actually rose and it was indeed higher than that of the first generation, which meant that the event got worse as the model predicted. So this lends even more credibility to the results that were generated from the model.

5 Challenges

There were unique challenges to this approach of predicting the time to warranty events.

Data collection: Field data was readily available but other miscellaneous information regarding the service call was absent. This presented issues when warranty scenarios and events had to be created. Part replacement data was the only way to create the scenarios and this is especially difficult to do when multiple failure modes have the same parts replaced. So we had to be careful to select an appropriate scenario to negate this issue. This resulted in lot of potential warranty scenarios not being used for this work.

Another issue was that the product development data did not have that many instances of the failure mode that was being considered. However, the results from this work seem to suggest that that limitation may be possible to deal with.

Finally, while the engineering judgment data that was used in this work was captured in the FMEAs, strictly speaking, this analysis would have been influenced by the previous field experience of the product.

6 Conclusions and future work

At the start of this work, the aim was to answer a set of research questions. This section will describe how well the questions were answered and what the future focus should be on.

1. Can we integrate different data streams to predict the occurrence of a warranty event?

This work showed that it is possible to integrate different data streams successfully to predict the occurrence of warranty event. The data was combined using Bayesian statistics. Also MCMC algorithms were used to generalize the application of the Bayes theorem in integrating the data sources.

2. Can the data used be of continuous form?

One of the main focuses of this work was to use continuous distributions of the different datasets to predict warranty. The datasets were successfully converted into a continuous form. So a framework to convert the data exists. But to use continuous form of the data, good datasets are required.

3. What is the best approach to integrate the data streams?

During the course of this research, the different datasets were combined in different combinations and it was found that, using field data as likelihood provided the most accurate prediction. But this was a second generation product; if future generations of products were used for analysis, this idea of using field data as likelihood could change. The approach looks promising but further validation is required before this can be used in an organization

4. Can the model be transferred to different platforms without major modifications?

For this research data for another product platform was not available. But since the methodology is generalized, it is hoped that using this model for other product platforms would give satisfactory results.

5. Can the model be applied to actual product data and produce good results?

This whole work was performed using data from our industrial partner and thus it is possible to apply this model to actual datasets. But it should be remembered that there is a still a lot of work that needs to be done to make the model more accurate.

In the future, the focus must be to replicate the analysis on a variety of warranty scenarios for a single product to further validate this model. Also this model was demonstrated on a second-generation product. The model could be totally different for a well-established product, something like a fifth generation product. In such a scenario, the engineering judgment is generally confounded with the field data of the previous generations. So the focus might be more on the field data and less on the others.

This analysis right now represents the data at a particular point during development. But during actual product development, there is a constant input of data, field data possibly changes every month and so would the distribution, tests are constantly performed on the product. So all this information could be used in the proposed model and needs to be validated.

In addition, for more novel development projects, the weight of different datasets may have to be altered. Lastly, other data sources like manufacturing data could be used in this model. However, given that the framework has been established and shown promise, it is conceptually easy to include other data forms.

7 References

Blischke, W., & Murthy, D. N. P. (1993). *Warranty Cost Analysis*: CRC Press.

Blischke, W., & Murthy, D. N. P. (1995). *Product Warranty Handbook*: CRC Press.

Blischke, W. R., & Murthy, D. N. P. (1992). Product warranty management, A Taxonomy for Warranty Policies. *European journal of operational research, 62*(2), 127-148.

Chukova, S., Arnold, R., & Wang, D. Q. (2004). Warranty analysis: An approach to modeling imperfect repairs. *International Journal of Production Economics, 89*(1), 57-68.

De, S., Das, A., & Sureka, A. (2010). Product failure root cause analysis during warranty analysis for integrated product design and quality improvement for early results in downturn economy. *International Journal of Product Development, 12*(Copyright 2011, The Institution of Engineering and Technology), 235-253.

Dezfuli, H., Kelly, D., Smith, C., Vedros, K., & Galyean, W. (2009). Bayesian inference for NASA probabilistic risk and reliability analysis. *NASA, Washington, DC*.

Duane, J. T. (1964). Learning Curve Approach to Reliability Monitoring. *Aerospace, IEEE Transactions on, 2*(2), 563-566.

Eighth Annual Warranty Report, Totals & Averages, 1 April 2011. from http://www.warrantyweek.com/archive/ww20110401.html

Esterman, M., Gerst, P., Stiebitz, P. H., & Ishil, K. (2005). *A framework for warranty prediction during product development.* Paper presented at the Proceedings of the ASME Design Engineering Division 2005, November 5, 2005 - November 11, 2005, Orlando, FL, United states.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1995). *Markov Chain Monte Carlo in practice: interdisciplinary statistics* (Vol. 2): Chapman & Hall/CRC.

Hamada, M. S., Wilson, A., Reese, C.S., Martz, H. (2008). *Bayesian Reliability*: Springer.

Huaiqing, W., & Meeker, W. Q. (2002). Early detection of reliability problems using information from warranty databases. *Technometrics, 44*(Copyright 2003, IEE), 120-133.

Hussain, A. Z. M. O., & Murthy, D. N. P. (2003). Warranty and optimal reliability improvement through product development. *Mathematical and Computer Modelling, 38*(Copyright 2004, IEE), 1211-1217.

Ion, R. A., & Sander, P. C. Y. J. (2005). Early reliability prediction in consumer electronics using Weibull distribution functions T2 - Reliability and Maintainability Symposium, 2005. Proceedings. Annual. *Reliability and Maintainability Symposium, 2005. Proceedings. Annual*.

Ireson, W. G., Coombs, C. F., & Moss, R. Y. (1996). *Handbook of reliability engineering and management*: McGraw Hill.

Majeske, K. D., Lynch-Caris, T., & Herrin, G. (1997). Evaluating product and process design changes with warranty data. *International Journal of Production Economics, 50*(Compendex), 79-89.

Mannar, K., & Ceglarek, D. (2005). *Fault region localization (FRL): Collaborative product and process improvement based on field performance.* Paper presented at the Intelligent Systems in Design and Manufacturing VI, October 24, 2005 - October 26, 2005, Boston, MA, United states.

Mazzuchi, T. A., & Soyer, R. (1993). A Bayes method for assessing product-reliability during development testing. *Reliability, IEEE Transactions on, 42*(3), 503-510.

Ming, Z., Zhang, Y., Tao, J., & Chen, X. (2010). Bayesian reliability assessment and prediction during product development. *Jixie Gongcheng Xuebao/Journal of Mechanical Engineering, 46*(Compendex), 150-156.

Murthy, D. N. P., & Djamaludin, I. (2002). New product warranty: A literature review. *International Journal of Production Economics, 79*(3), 231-260.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology, 47*(1), 90-100.

Vintr, Z., & Vintr, M. (2005). *FMEA used in assessing warranty costs.* Paper presented at the Reliability and Maintainability Symposium, 2005. Proceedings. Annual.

Wilson, E. (1993). Maximizing Designers' Impact on Market Success through Product Definition. *Design Management Journal (Former Series), 4*(4), 62-68.

Yadav, O. P., Singh, N., Goel, P. S., & Itabashi-Campbell, R. (2003). A Framework for Reliability Prediction During Product Development Process Incorporating Engineering Judgments. *Quality Engineering, 15*(4), 649 - 662.

Yang, K., & Cekecek, E. (2004). Design Vulnerability Analysis and Design Improvement by Using Warranty Data. *Quality and Reliability Engineering International, 20*(Compendex), 121-133.

Zhou, Z., Jin, G., Dong, D., & Zhou, J. (2006). *Reliability analysis of multistate systems based on Bayesian networks.* Paper presented at the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer-Based Systems, ECBS 2006, March 27, 2006 - March 30, 2006, Potsdam, Germany.

8 Appendix: Complete WinBUGS code for all test cases

Complete WinBUGS Code for all the cases

There are 3 data sources. Product Development test data (PD data), PD data follows a 2 parameter Weibull distribution with the following parameters. Beta (shape) = 1.749 and Scale = 21.469 years
Field Data, The field data follows a lognormal distribution with the following parameters. Log-mean: 4.69 years and Log-sd: 2.009
Engineering Judgment data, it follows a lognormal distribution with the following parameters, Log mean: 1.43 years and an error factor of 5

For stage 1, Case 1 and 1x has the same code, only in stage 2, code for Case 1 and 1x differs

 # Case 1
 # Case 1x

```
model {
scale.posterior ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.output)),shape.pd)

 scale.output ~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- mu.field
tau.analysis <-1/pow(field.sd,2)
}

data
list(mu.field=4.69,field.sd=2.009,shape.pd=1.73)  # pd prior, # field likelihood
```

 # Case 1 Stage 2

```
# Results of Stage 1 lognormal Distribution: Mean 4.34 SD 2.143
# Stage 2 uses stage 1 results as prior and engg judgment as likelihood

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- mu.enggjd
tau.analysis2<-1/pow(sd.enggjd,2)

}
data
```

list(mu.enggjd=1.73,sd.enggjd = 0.97,stage1.sd = 2.143)

 # Case 1x Stage 2
Results of Stage 1 lognormal Distribution: Mean 4.34 SD 2.143
Stage 2 uses stage 1 results as likelihood and engg judgment as prior

```
model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(sd.enggjd,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- stage1.mu
tau.analysis2<-1/pow(stage1.sd,2)
}

data
list(stage1.mu = 4.34,sd.enggjd = 0.97,stage1.sd = 2.143)
```

 # Case 2
 # Case 2x

```
model {

scale.posterior~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- log(scale.output)
tau.analysis <-1/pow(field.sd,2)

scale.output ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.pd)),shape.pd)

}

data
list(shape.pd=4.69,scale.pd=21.469,field.sd=2.009)  # pd likelihood, # field prior
```

 # Case 2 Stage 2
Results of Stage 1 lognormal Distribution : Mean 19.568 SD 5.218
Stage 2 uses stage 1 results as prior and engg.jd as likelihood

```
model {
```

```
scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- mu.enggjd
tau.analysis2<-1/pow(sd.enggjd,2)

}

data
list(mu.enggjd=1.43,sd.enggjd = 0.97,stage1.sd = 2.007)
```

 # Case 2x Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 2.905 SD 2.007
# Stage 2 uses stage 1 results as likelihood and engg judgment as prior

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(sd.enggjd,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- stage1.mu
tau.analysis2<-1/pow(stage1.sd,2)

}

data
list(stage1.mu = 2.905,sd.enggjd = 0.97,stage1.sd =2.007)
```

 # Case 3
 # Case 3x

```
model {

 scale.posterior ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.output)),shape.pd)


scale.output ~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- mu.enggjd
tau.analysis <-1/pow(engg.sd,2)
```

```
}

data
list(mu.enggjd=1.43,engg.sd=0.978,shape.pd=1.73)  # pd prior, # engg judgment likelihood
```

 # Case 3 Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 1.088 SD 1.23
# Stage 2 uses stage 1 results as prior and field data as likelihood

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- mu.field
tau.analysis2<-1/pow(sd.field,2)

}

data
list(mu.field=4.69,sd.field = 2.009,stage1.sd = 1.23)
```

 # Case 3x Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 1.088 SD 1.23
# Stage 2 uses stage 1 results as likelihood and field data as prior

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(sd.field,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- stage1.mu
tau.analysis2<-1/pow(stage1.sd,2)

}

data
list(stage1.mu=1.088 ,sd.field= 2.009,stage1.sd= 1.23)
```

 # Case 4

 # Case 4x

```
model {

scale.posterior~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- log(scale.output)
tau.analysis <-1/pow(engg.sd,2)

scale.output ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.pd)),shape.pd)

}

data
list(shape.pd=4.69,scale.pd=21.469,engg.sd=0.97)  # pd likelihood, # engg jd prior
```

 # Case 4 Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 2.923 SD 0.999
# Stage 2 uses stage 1 results as prior and field data as likelihood

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- mu.field
tau.analysis2<-1/pow(sd.field,2)

}

data
list(mu.field=4.69,sd.field = 2.009,stage1.sd = 0.999)
```

 # Case 4x Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 2.923 SD 0.999
# Stage 2 uses stage 1 results as likelihood and field data as prior

model {

scale.posterior2 ~ dlnorm(scale.output3,sd.analysis3)
sd.analysis3 <- 1/pow(sd.field,2)
scale.output3 <- log(scale.output2)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
```

```
mu.analysis2 <- stage1.mu
tau.analysis2<-1/pow(stage1.sd,2)

}

data
list(stage1.mu=2.923 ,sd.field= 2.009,stage1.sd= 0.999)
```

 # Case 5
 # Case 5x

```
model {

scale.posterior ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- log(scale.output)
tau.analysis2 <- 1/pow(field.sd,2)


scale.output ~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- mu.enggjd
tau.analysis <-1/pow(engg.sd,2)
}

data
list(mu.enggjd=1.43,engg.sd=0.978,field.sd=2.009)  # field prior, # engg judgment likelihood
```

 # Case 5 Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 1.43 SD 2.21
# Stage 2 uses stage 1 results as prior and pd data as likelihood

model {

scale.posterior2 ~ dlnorm(scale.analysis3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.analysis3 <- log(scale.output2)



scale.output2 ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.pd)),shape.pd)

}

data
list(shape.pd=4.69,scale.pd=21.469,stage1.sd=2.21)
```

 # Case 5x Stage 2
```
# Results of Stage 1 lognormal Distribution : Mean 1.43 SD 2.21
# Stage 2 uses stage 1 results as prior and pd data as likelihood

model {

scale.posterior2 ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.output2)),shape.pd)



scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- stage1.mu
tau.analysis2<-1/pow(stage1.sd,2)

}

data
list(stage1.mu = 1.43,shape.pd=1.749,stage1.sd = 2.21)
```

 # Case 6
 # Case 6x

```
model {

scale.posterior ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- log(scale.output)
tau.analysis2 <- 1/pow(engg.sd,2)


scale.output ~ dlnorm(mu.analysis,tau.analysis)
mu.analysis <- mu.field
tau.analysis <-1/pow(field.sd,2)
}

data
list(mu.field=4.69,engg.sd=0.978,field.sd=2.009)  # field likelihood , # engg judgment prior
```

 # Case 6 Stage 2
```
# Results of Stage 1 lognormal Distribution : Mean 4.69 SD 2.21
# Stage 2 uses stage 1 results as prior and pd data as likelihood
```

```
model {

scale.posterior2 ~ dlnorm(scale.analysis3,sd.analysis3)
sd.analysis3 <- 1/pow(stage1.sd,2)
scale.analysis3 <- log(scale.output2)


scale.output2 ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.pd)),shape.pd)

}

data
list(shape.pd=4.69,scale.pd=21.469,stage1.sd=2.21)
```

 # Case 6x Stage 2

```
# Results of Stage 1 lognormal Distribution : Mean 4.691 SD 2.213
# Stage 2 uses stage 1 results as prior and pd data as likelihood

model {

scale.posterior2 ~ dweib(shape.pd,scale.analysis)
scale.analysis <- pow((1/(scale.output2)),shape.pd)

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- stage1.mu
tau.analysis2<-1/pow(stage1.sd,2)

}

data
list(stage1.mu = 4.691,shape.pd=1.749,stage1.sd = 2.213)
```

 # Tesla Field

```
model {

scale.output2 ~ dlnorm(mu.analysis2,tau.analysis2)
mu.analysis2 <- mu.field
tau.analysis2<-1/pow(sd.field,2)

}
Data
list(mu.field=13.13,sd.field = 5.96)
```