

Chapter 2

Modelling Quality and Warranty Cost

C. Sean Bohun¹, Olivier Dubois², Hongbin Guo³, Veena B. Mendiratta⁴, Nilima Nigam²,
Kostyantyn Stepankevych⁵, Tzvetalin Vassilev⁶

Report prepared by Nilima Nigam (nigam@math.mcgill.ca)

2.1 Problem Description and Methodology

The main aim of this project was to begin a modelling effort directed at optimizing the warranty and quality costs associated with the production of a system with both hardware and software components. This optimization would be constrained by the need to maintain reliability of the product, while staying within an operational budget. For a more detailed problem statement, see [1]. Our aim was to identify important quality attributes, and capture overall trends in costs and warranties. More concretely, our goals were:

- Identifying the major quality-related attributes of interest, denoted by a vector \mathbf{q} ,
- Modelling the key indicators of the reliability constraint: the failure rate ($\text{FR}(\mathbf{q})$) and the severity level ($\text{SL}(\mathbf{q})$),
- Modelling the cost of building a product to a certain quality level, $C(\mathbf{q})$,
- Modelling the warranty costs of a product built to a certain quality level, $W(\mathbf{q})$.

The optimization model is to minimize the sum of the quality and warranty costs over the entire class of admissible quality-related attribute vectors. This procedure is accomplished while simultaneously ensuring that the failure rate remains below a specified maximum FR_{\max} and the severity level remains above a given minimum SL_{\min} with a given probability level p . In other words, determine

$$\mathcal{F}_{opt} = \min_{\mathbf{q}}(C(\mathbf{q}) + W(\mathbf{q})), \quad (2.1)$$

¹Pennsylvania State University

²McGill University

³University of Alberta

⁴Lucent Technologies

⁵University of Calgary

⁶University of Saskatchewan

subject to

$$P(\text{FR}(\mathbf{q}) < \text{FR}_{\max}) < p, \quad P(\text{SL}(\mathbf{q}) > \text{SL}_{\min}) < p \quad (2.2)$$

over all admissible \mathbf{q} . In this project we did not perform this optimization, focusing instead on the modelling of the function involved.

It is important to note at this juncture that no raw data from Lucent was provided for this project, nor did we have specific information about the particular products being built. It was therefore not feasible to use existing hazard/risk models for the various components. Our modelling effort was thus critically dependent on discussions with the industrial contact, Prof. Veena Mendiratta. In the section on future directions we make a series of recommendations which will help refine the models involved.

We systematically identified the key quality-related attributes, described by a quality vector \mathbf{q} , which could be measured and quantified. We then developed reliability, warranty and cost models based on these. As our discussions progressed, it became clear that these quality attributes were not all independent. Nor were they all equally important indicators of overall quality. It is thus possible to simplify the models considerably by focusing on the effects of the most important attributes, making the optimization problem (2.1) simpler to solve. In practice, once cost functions and parameters have been picked on the basis of standard hazard models, it will be possible via scaling arguments to achieve further simplification.

With a view to illustrating qualitative trends predicted by our models, we generated some test data (see Subsection 2.8), and ran our models on them. The graphs presented in this report are therefore not linked to any true data, and serve only to provide qualitative information.

2.1.1 A Road Map

The following list details the strategy for this report. Figure 2.1 illustrates how the various sections of the report interconnect.

Section 2: In this section we identify the quality-related variables, \mathbf{q} , which drive the various costs associated with a product, and over which the optimization will occur. The fact that many of these variables are not independent will be dealt with later in this report.

Section 3: Here, we develop models for the reliability constraints, the failure rate FR and the severity levels SL. As well as providing some graphical insight into the dependence of these models on the quality \mathbf{q} , we also discuss how these failure rates determine the probability of the various modes of failure. These probabilities play a role in determining the warranty costs of a product.

Section 4: At this point in the report a model for $C(\mathbf{q})$, the cost of implementation of a given quality level \mathbf{q} is proposed.

Section 5: This contains the development of the warranty models for hardware W_{hw} and software W_{sw} aspects of a product.

Section 6: Here we combine the models to summarize the total proposed optimization problem.

Section 7: A sensitivity to parameters is discussed, providing insight into the relative importance of terms in the various models that have been introduced.



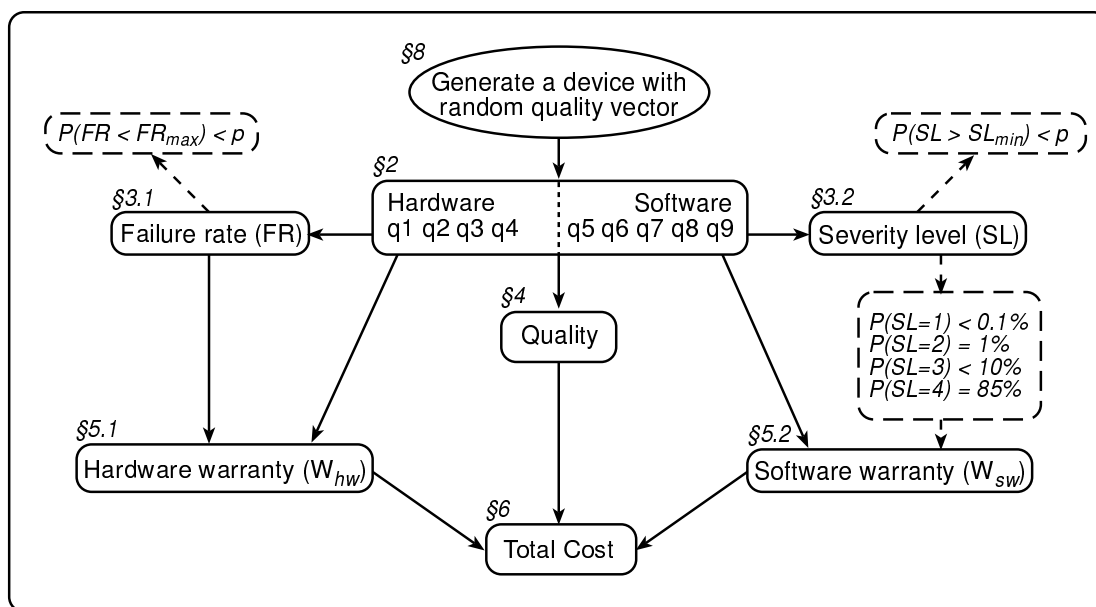


Figure 2.1: Illustrated are the various sections of this report and how they interconnect.

Section 8: Broad trends in the proposed models are generated by drawing the quality-related variables from a given probability distribution. Two cases are considered, each characterized by the probability distribution being used.

2.2 Quality Attributes Vector

We begin by identifying the important quality-related attributes which are both salient and measurable in the context of this project. These attributes fall into two broad categories – hardware-related and software-related – and the optimization of the total cost will be performed over these attributes. In practice, most of these attributes will be measured statistically. In the absence of raw data, we are unable to provide statistical models for these attributes, which will change depending on the product.

Mathematically, these attributes are gathered in a **quality vector**

$$\mathbf{q} = (q_1, q_2, q_3, \dots, q_9) \in D := [0, 1]^9.$$

The cost will be optimized as a function of $\mathbf{q} \in D$, subject to certain reliability constraints. We have scaled these attributes q_i to take on values between 0 and 1 for convenience. This enables us to compare, for example, a quantity originally measured as a percentage with one measured as a number between 0 and 10. When using the model in application, it will be important to identify the units used and convert them if necessary.

The various quality attributes are described below.

HARDWARE:

q_1 : *Component quality*. In practice measured as a failure rate percentage per year. Here, this rate is converted to a scale from 0 to 1, and is called q_1 .



- q_2 : *Infant mortality factor (IMF)*. Measured as the ratio of the initial failure rate to the steady-state failure rate, this is a number between 1 and 2. In this project, we use the scaling $q_2 = \text{measured infant mortality factor} - 1$.
- q_3 : *Diagnostics capability*. This attribute is denoted q_3 and lies between 0.8 and 1. In practice, it is measured as a percentage, typically between 80 and 100.
- q_4 : *Working environment range*. The variable q_4 is defined as the amount by which the constructed working range exceeds the specifications of the device. For example, suppose the device is intended to operate between 0°C and 100°C , but is built with a working range of -10°C to 125°C . The constructed working range exceeds the operational specifications by 10°C on the lower end, and by 25°C on the higher end. Thus, we would compute $q_4 = (10 + 25)/(\text{working range}) = 35/100 = 0.35$.

From the description of these hardware-related attributes, it is not immediately obvious which are the best indicators of overall quality.

SOFTWARE:

- q_5 : *Software development environment (SDE)*. Denoted q_5 , this describes the overall quality metric of the software development process.
- q_6 : *Code complexity*. This metric measures the complexity of a code based on a variety of indicators. Essentially, the more complicated the interactions between different parts of a large code, the harder it is to ensure reliability.
- q_7 : *Stability index*. Typically a number between 0.8 and 1, this metric describes the robustness of a code over longer periods of time.
- q_8 : *Coverage testing*. This attribute describes how comprehensively each module of the code has been checked.
- q_9 : *Fault density*. This measures the number of failures per 1000 lines of code. We express this as a fraction between 0 and 1.

The SDE index clearly seems to include, or be affected by, the other software-related attributes. We expect a good model will therefore be very sensitive to changes in q_5 . In particular cases, these quality attributes may be restricted to tighter “operating ranges” by the company’s production policies.

2.3 How do we Model the Reliability Constraints?

The optimization of the costs of quality and warranty would be straightforward in the absence of certain *reliability constraints*. These constraints are identified as benchmarks, or standards, which must be met by any product. The quality attributes must be chosen to meet or exceed these standards.

Prior to prescribing the nature of the constraints, we need to model the indicators of reliability which will be used. There are two major indicators, one for hardware and one for software.



HARDWARE:

Failure Rate (FR): this is described by the *system* failure rate per year, and includes the effect of the component failure rate q_1 .

SOFTWARE:

Severity Levels (SL): ranges in scale from 1 to 4, where $SL = 1$ is a catastrophic failure, and $SL = 4$ is a minor error.

The reliability constraints will be interpreted in terms of these indicators – the failure rate FR must be below a certain prescribed value with high probability, and the severity level SL must stay away from the catastrophic failures with high probability. This is illustrated in expression (2.2).

2.3.1 Modelling the Failure Rate

The failure rate used in the characterization of reliability combines several factors including the failure rate of the components themselves, the robustness of the overall architecture, the infant mortality factor (IMF) and the working environment range.

We identified the broad trends that the failure rate exhibited in three of the quality attributes: component failure rate q_1 , the infant mortality factor q_2 and the working environment range q_4 . As the component failure rate q_1 increases, so does the overall failure rate. Likewise, if the IMF q_2 is high, the failure rate is large. The effect of the working range environment q_3 is opposite: if the constructed working range is larger than the specs, the device is more robust and thus the failure rate goes down.

We proposed two models with increasing complexity that exhibit this behaviour. Our discussions revealed that in this specific context the failure rate was described largely in terms of the component quality.

The first model $FR_1(\mathbf{q})$ is a simple one, with 3 free parameters f_1 , f_2 , and f_3 :

$$FR_1(\mathbf{q}) = FR_1(q_1, q_2, q_4) = f_1 q_1 + f_2 q_2 - f_3(1 - q_1)q_4. \quad (2.3)$$

The nonlinear term $-f_3(1 - q_1)q_4$ enters since the failure rate should *decrease* with larger working environment range q_4 , however the system will nevertheless be affected by poor component failure rates q_1 . These two effects are therefore competing.

Figure 2.2 below shows four graphs related to failure rate model FR_1 . The first three graphs exhibit the trends of the failure rate with respect to the individual attributes q_1 , q_2 , q_4 . The last graph depicts a surface plot describing failure rate trends when q_1 and q_4 are allowed to change.

The next failure rate model we propose is manifestly nonlinear, and aims to better capture the importance of the component failure rate q_1 on the system failure rate FR. The free parameters are denoted f_1 , f_2 , f_3 and f_4 . As before, the failure rate FR depends on the quality vector \mathbf{q} , but in particular on the attributes q_1 , q_2 , q_4 .

$$FR_2(q_1, q_2, q_4) = f_1 e^{f_2 q_1} + f_3 q_2 q_1^2 - f_4(1 - q_1)q_4. \quad (2.4)$$

The rationale for picking this model is as follows: first, the system failure rate $FR(\mathbf{q})$ increases with poorer component quality, with this rate of change depending on q_1 . Therefore the dependence of



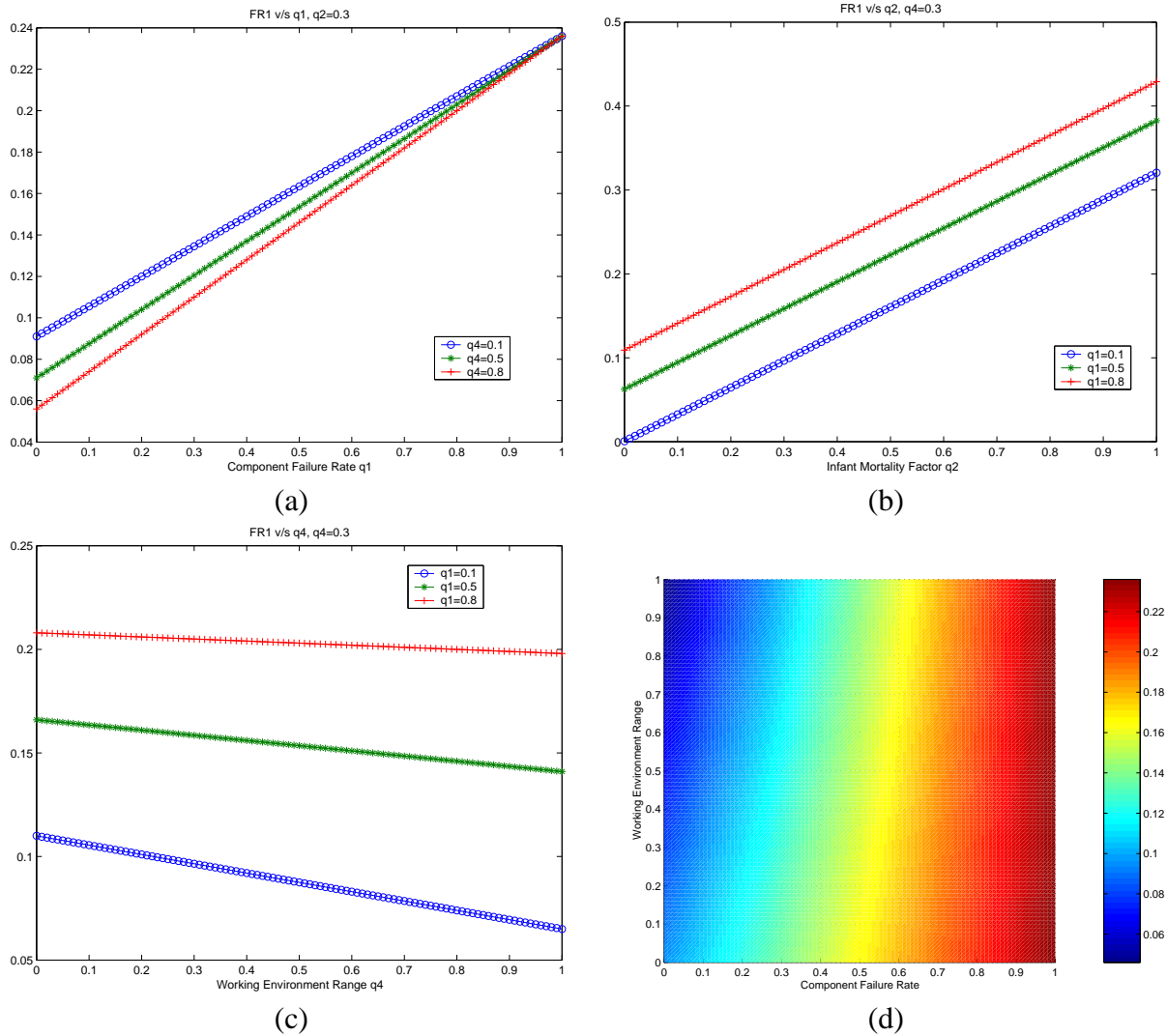


Figure 2.2: Failure Rate model FR_1 as a function of (a) component quality q_1 , (b) infant mortality q_2 , (c) working environment range q_4 , and (d) both q_1 and q_4 together, $q_2 = 0.3$.

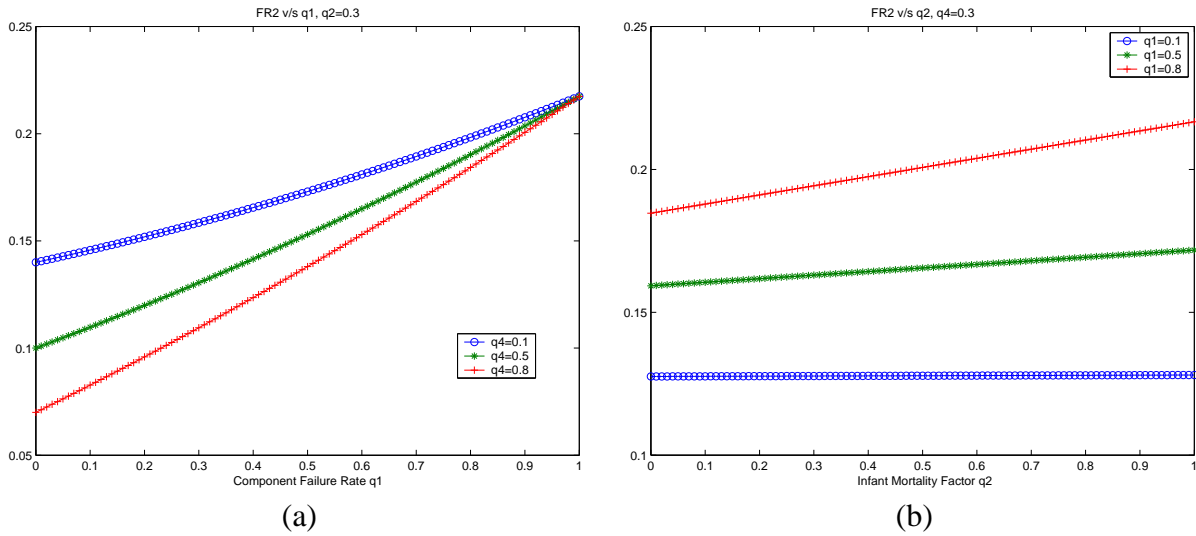


Figure 2.3: Failure Rate model FR_2 as a function of (a) component quality q_1 , and (b) infant mortality q_2 .

FR_2 on q_1 is modelled by an exponential. Second, the initial mortality rate q_4 impacts the overall failure rate, but even if this IMF is low, a poor-quality component will impact the failure rate adversely.

The two graphs in Figure 2.3 use the failure model (2.4) for FR to describe the broad trends in the model with component failure rate q_1 and infant mortality factor q_2 and can be compared to Figure 2.2. In Section 2.8 we show the effect of inputting several instances of \mathbf{q} , drawn from test data, into the model FR_2 .

2.3.2 Modelling the Severity Level

Software failures are characterized in terms of varying *severity levels* (hereafter denoted SL), where an $SL = 1$ is a catastrophic failure, while an $SL = 4$ is a minor failure. In this section we present some models describing the relationship between the quality vector \mathbf{q} and the SL.

In the context of this specific project, we determined that the severity levels of software failure were impacted by the software development environment q_5 , the code complexity q_6 , the stability index q_7 , the coverage testing q_8 and the fault density q_9 . The model we propose for the severity levels is not an additive/linear one. We believe that the chosen functional form captures well the trends in severity levels as functions of the individual attributes, as well as the relative importance amongst these factors. There are some free (nonnegative) parameters in the model, s_1, s_2, s_3, s_4, s_5 . The severity level SL as a function of \mathbf{q} is:

$$SL_1(\mathbf{q}) = SL_1(q_5, q_6, q_7, q_8, q_9) = s_1 e^{s_2 |q_9 - 0.8| - s_3 |q_8 - 0.1|} \left[\sqrt{q_5} + (1 + q_6 - q_6^2)^{s_4} q_7^2 \right]. \quad (2.5)$$

To describe the effects of coverage testing q_8 , we noted that as q_8 increases, the likelihood of catastrophic software error decreases since more of the software is validated. Similarly, as the number of faults per 1K lines, q_9 , increases, so does the risk of catastrophic error. Keeping in mind the scale

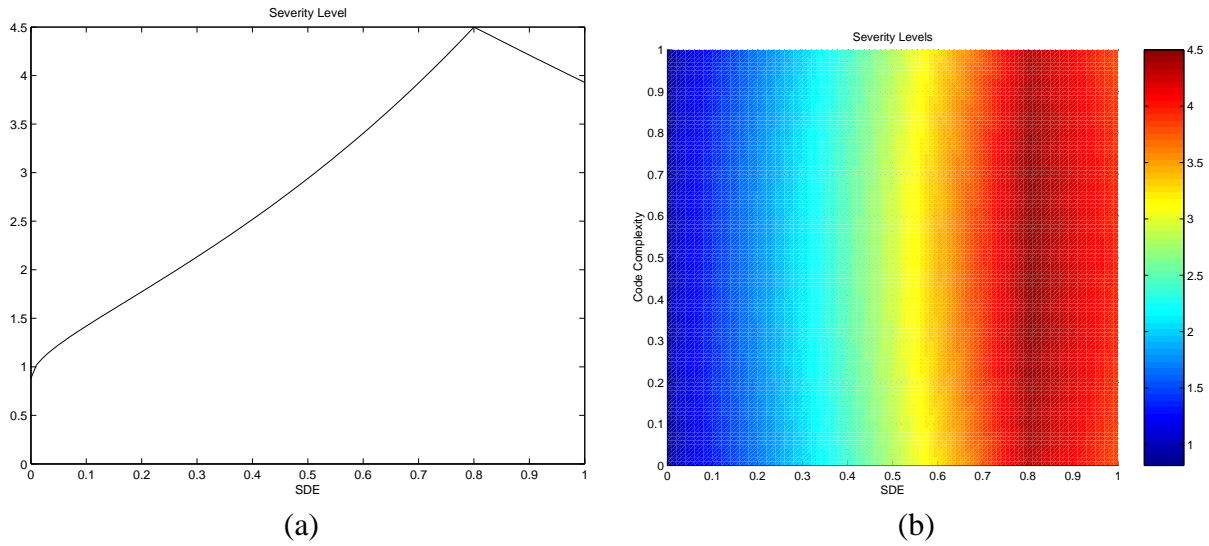


Figure 2.4: Severity levels as a function of the various components of the quality-related vector: (a) SDE q_5 , (b) SDE and code complexity (q_5, q_6) .

on which we measure SL, the dependence on q_8 and q_9 is modelled by exponentials with appropriate signs, penalizing deviations from high-quality.

Based on discussions, we modelled the dependence of SL on the stability index q_7 by a quadratic, since a more stable code is less prone to severe software failures.

As the software development environment indicator q_5 increases, the types of software failures get less severe and the SL increases. Poor quality development environment impacts the severity level more. That is, $\partial(\text{SL})/\partial q_5$ should be larger for small values q_5 . This behaviour is captured well by the square root function.

The opposite trend is exhibited as a function of code complexity q_6 . When the code complexity is low, the overall software is less prone to severe errors, putting the SL index in the high range. After a certain threshold complexity is exceeded, the effect of complexity on the severity levels becomes less dramatic. To capture this behaviour, the dependence of SL on q_6 is described by $(1 + q_6 - q_6^2)^{s_4}$ where $s_4 < 1$.

While discussing SL it appeared that the attributes q_8 , q_9 , the coverage testing and fault density, were well-predicted by the software development environment, q_5 . Therefore, we assumed that at least for the purpose of modelling severity levels as a function of \mathbf{q} , we could write

$$q_8 = K'_8 q_5, \quad q_9 = K'_9 q_5. \quad (2.6)$$

This suggests a possible simplification to the severity level model:

$$\text{SL}_2(\mathbf{q}) = \text{SL}_2(q_5, q_6, q_7) = s_1 e^{-s_5 |q_5 - 0.8|} \left[\sqrt{q_5} + (1 + q_6 - q_6^2)^{s_4} q_7^2 \right]. \quad (2.7)$$

where s_1 and s_4 are as in model (2.5), while $s_5 = s_2 K'_9 - s_3 K'_8$. The trends in the severity level are graphically described in the Figure 2.4.



Failure type	(s_1, s_4, s_5)			
	(2.46,0.4,1)	(2.46,0.6,1)	(2.46,0.4,2)	(2.46,0.4,3)
SL $\in (0, 1.5)$	0.1 %	0.01 %	0.01 %	0.02 %
SL $\in (1.5, 2.5)$	0.9 %	0.1 %	0.1 %	0.8 %
SL $\in (2.5, 3.5)$	12.7 %	14 %	29 %	43.5 %
SL $\in (3.5, 4.5)$	87.2 %	85.8 %	70.4 %	55.6 %

Table 2.1: Predicted distribution of severity levels for various sets of (s_1, s_4, s_5) .

Even with this simplification, the severity model described by equation (2.7) is highly nonlinear. How is one to choose the exponent s_4 ? Does this model actually capture the observed behaviour of software systems when they are built within a given range of quality?

To answer these questions, we first determined the heuristic trend: if the software development environment q_5 , the code complexity q_6 and the stability index q_7 were in the high-end, then the number of software failures classified as SL = 1 (catastrophic) should be less than 0.1%, SL = 2 failures should be about 1%, SL = 3 failures should be less than 10% and SL = 4 failures should be about 85%. A good reality check for our SL model (2.7) is to draw (q_5, q_6, q_7) from a given set of distributions. For our first simulation we take (q_5, q_6, q_7) from normal distributions with means $\mu_5 = 0.8$, $\mu_6 = 0.4$, $\mu_7 = 0.8$ and a common variance $\sigma^2 = 0.05$ so that $q_5 \sim N(0.8, 0.05)$, $q_6 \sim N(0.4, 0.05)$ and $q_7 \sim N(0.8, 0.05)$. The probability of each SL failure type can be computed through

$$\text{Probability of an SL type } i \text{ failure} = \frac{\mu \{ (q_5, q_6, q_7) \in \Omega \mid \text{SL}(q_5, q_6, q_7) = i \}}{\mu \{ (q_5, q_6, q_7) \in \Omega \}},$$

where $\Omega := \{(q_5^i, q_6^i, q_7^i)_{i=1}^{1000} \mid q_5 \sim N(0.8, 0.05), q_6 \sim N(0.4, 0.05), q_7 \sim N(0.8, 0.05)\}$ is a set of 1000 i.i.d. test data points drawn from the appropriate normal distributions, and $\mu(S)$ is the volume of a set S . We show in Table 2.1 these (approximate) percentages for a few choices of s_1, s_4 and, critically, s_5 . We note that these ranges are not obtained from Lucent, but are used because they seem consistent with a high-end product. The code-complexity q_6 was set to be mid-range since a marketable system would have a certain minimal level of complexity, but high complexity was undesirable.

2.4 The Cost of Quality Implementation

Having identified the constraints in the previous section, we now describe the costs associated with building a product with given quality vector \mathbf{q} . Our discussion revealed that the largest effects on the cost were due to maintaining a high software development environment q_5 , and a low component failure rate q_1 .

The model for the cost of quality, $C(\mathbf{q})$ which is proposed in this project is

$$\left. \begin{aligned} C(\mathbf{q}) &= C_{hw} + C_{sw} \\ &= C_1 e^{-C'_1 q_1} + c_2 (q_2 + c'_2)^2 + c_3 q_3 + c_4 (q_4 + c'_4)^2 + \text{fixed hardware costs} \\ &\quad + C_5 e^{C'_5 q_5} + c_6 q_6 + c_7 q_7 + \frac{c_9 q_9}{q_6 + c'_9} + \text{fixed software costs.} \end{aligned} \right\} \quad (2.8)$$



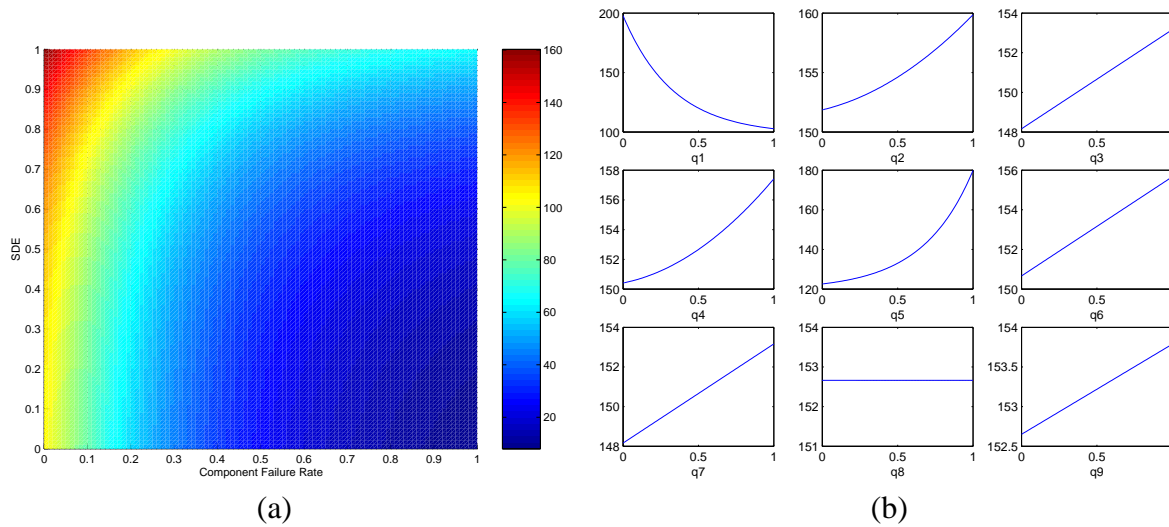


Figure 2.5: Behaviour of the quality function. (a) Trend in cost of quality as a function of component failure rate q_1 and SDE q_5 . (b) Trend of cost of quality in individual attributes q_i .

Here $C_1, C'_1, C_5, C'_5, c_2, c'_2, c_3, c_4, c'_4, c_6, c_7, c_9, c'_9$ are constant parameters which need to be determined by fitting actual data to the model.

The first term of the hardware and software portions in this model capture the importance of the component failure rate q_1 and the software development environment q_5 in the overall cost model. As q_1 decreases, the overall likelihood of failure decreases. This improvement costs more, especially after a certain threshold is achieved. Improvements beyond this level are increasingly expensive, as captured by an exponential function with the negative exponent. On the other hand, as SDE q_5 increases, the software development environment becomes better and thus costs more. These trends are captured in Figures 2.5(a) and (b).

The terms collected in equation (2.8) under *hardware* describe the effects of the infant mortality rate q_2 , the diagnostics capability q_3 and the working environment range q_4 . In terms of the overall hardware quality costs, these are higher-order effects in the sense that their contribution may not be as significant as that of the component failure rate, q_1 . This reasoning dictated the functional relationships as being at best quadratic. Similarly, the costs collected under *software* describe the effects of controlling the code complexity q_6 and the stability index q_7 . These costs contribute less significantly to the overall quality costs for the software than the software development environment q_5 . Indeed, the effects of increasing the coverage testing and decreasing the fault density are captured (to a large extent) by the cost of q_5 , and are therefore ignored in this cost model. The trends of the cost as each of these attributes vary is pictured graphically in Figure 2.5. The results using the test data are described in Section 2.8.

2.5 The Warranty Costs

Warranty costs can be broadly broken up into the hardware and software costs, and we thus modelled each of these separately. As in the cost of quality, the dominant factors involved are the component



failure rates q_1 and the software development environment q_5 . We now present a model for the warranty costs $W(\mathbf{q})$ expressed as

$$W(\mathbf{q}) = W_{hw}(\mathbf{q}) + W_{sw}(\mathbf{q}).$$

In this project we do not attempt to present a model for finding an optimal warranty policy. Our attempt is to model the effect of changing quality on warranty costs for a *given, fixed* warranty policy. This distinction is an important one. Clearly, the policies themselves will change considerably if the average product quality attributes change a lot. This model does not account for this, at least in the hardware costs. However, as a first approximation, if we assume the \mathbf{q} vector stays within a certain range, the warranty policy may be considered fixed, and we can describe the effects on the warranty costs of changing \mathbf{q} within this range.

2.5.1 The Hardware Warranty Costs

Hardware warranty costs are characterized by the four major types of hardware failures seen: *no trouble found* (NTF), *repaired*, *junked*, and *further failure modes analysis* (FMA). Of these, the NTF costs are the smallest, but the supplier may wish to penalize these. The model should be flexible enough so that an optimization will ensure that most of the errors fall into the second (repaired) category.

Empirical data will be able to describe the observed probabilities p_1, p_2, p_3 and p_4 of seeing the various warranty-related costs (NTF, repair, junk, and FMA, respectively). These probabilities are computed based on the assumption that the quality vector \mathbf{q} lies within a particular range, but are not sensitive to variations within the range. There is also a standardized dollar amount w_1, w_2, w_3, w_4 associated with each of these.

The standard warranty cost model simply computes the expected cost $E(C) = \sum_{i=1}^4 p_i w_i$. As a result this standard model fails to capture the most important trend, that of changing component failure rate q_1 , on the hardware warranty costs.

To include the effects of quality, we *penalize* these four kinds of failures to various degrees. This will allow the user, for example, to explicitly adjust the model so that NTF failures are reduced. This allows for greater flexibility in optimization. For example, NTF failures are inexpensive, and thus optimizing a standard warranty cost model may result in choosing \mathbf{q} values which *increase* NTF failures, while reducing the expensive FMA failures. By contrast, the proposed hardware warranty model (2.9) allows the user to choose the penalty parameters k_1, k_2, k_3, k_4 so that the optimal \mathbf{q} values result in a small number of NTF.

$$W_{hw}(\mathbf{q}) = w_0 q_1 \left\{ \underbrace{p_1 [w_1 + k_1 (1 - q_3)]}_{\text{NTF costs}} + \underbrace{p_2 (w_2 + k_2 q_2)}_{\text{repair costs}} + \underbrace{p_3 (w_3 + k_3 q_1)}_{\text{junk costs}} + \underbrace{p_4 [w_4 + k_4 (1 - q_3)]}_{\text{FMA}} \right\} \quad (2.9)$$

where $p_1 = 0.1, p_2 = 0.8,$ and $p_3 = p_4 = 0.05$.

These probabilities are based on how many items which are returned to the supplier (and are ad hoc), while w_0 is used to scale costs appropriately by the number of items in service. We notice that the standard warranty cost model has been multiplied by q_1 , the component failure rate. In the proposed model if the component failure rates q_1 are at the high end of the operating range, then the warranty costs are higher. Figure 2.6 depicts the various dependencies.



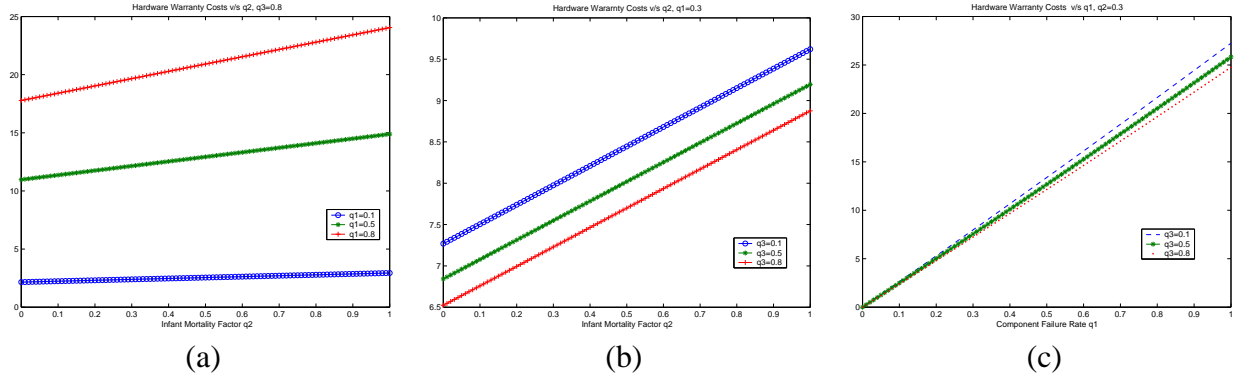


Figure 2.6: (a) Trends in W_{hw} with component quality q_1 and IMF q_2 , (q_3 fixed). (b) Trends in W_{hw} with IMF q_2 and diagnostics capability q_3 , (q_1 fixed). (c) Trends in W_{hw} with component quality q_1 and diagnostics capability q_3 , (q_2 fixed).

2.5.2 The Software Warranty Costs

The software warranty costs are characterized in terms of the severity levels of the software failures, SL (see Section 2.3.2). Therefore, given the operating range of the quality vector \mathbf{q} , we can calculate from our SL model the probabilities p_5, p_6, p_7, p_8 of failures of SL = 1 (catastrophic) through SL = 4 (minor) respectively (see Table 2.1). Associated with each of these types of failures is a warranty cost, w_5, w_6, w_7 and w_8 respectively. The warranty cost for software is expressed by:

$$W_{sw}(\mathbf{q}) = \widetilde{W}(1 - q_5)(p_5w_5 + p_6w_6 + p_7w_7 + p_8w_8). \quad (2.10)$$

The SDE most significantly impacts the warranty cost of the software. A higher SDE means fewer catastrophic failures. The probabilities $p_i, i = 5, \dots, 8$ are fixed for a given operational range of q_5, q_6, q_7 , and are chosen according to our model of the Severity Level function (2.7). Recall that q_8 and q_9 are given by expression (2.6). More precisely,

$$p_{i+4} = \frac{\mu \{(q_5, q_6, q_7) \in \Omega \mid \text{SL}(q_5, q_6, q_7) = i\}}{\mu \{(q_5, q_6, q_7) \in \Omega\}} + \epsilon_i$$

where $\Omega := (0.8, 1) \times (0.4, 0.7) \times (0.7, 1)$. The quantities ϵ_i are introduced to account for other lower-order effects, which are not accounted for during the production. These include effects such as a product being returned due to incorrect usage by the customer. One may also use a model allowing more flexibility, such as (2.9). Unfortunately, in such a model the probabilities will need to be computed as functions of q_i .

2.6 The Complete Model

We now summarize the models of the previous sections. Recall that the goal was to identify the various functions in the optimization problem (2.1) repeated here for convenience

$$\mathcal{F}_{opt} = \min_{\mathbf{q}}(C(\mathbf{q}) + W(\mathbf{q})),$$



subject to

$$P(\text{FR}(\mathbf{q}) < \text{FR}_{\max}) < p, \quad P(\text{SL}(\mathbf{q}) > \text{SL}_{\min}) < p$$

over all admissible \mathbf{q} and some preset probability p . The objective function consists of the combined costs of quality (2.8) and warranty (2.9), (2.10):

$$\begin{aligned} \mathcal{F} &= C(\mathbf{q}) + W(\mathbf{q}) = C(\mathbf{q}) + W_{hw}(\mathbf{q}) + W_{sw}(\mathbf{q}) \\ &= C_1 e^{-c'_1 q_1} + C_5 e^{c'_5 q_5} + w_0 q_1 \sum_{i=1}^4 p_i w_i + \widetilde{W}(1 - q_5) \sum_{i=5}^8 p_i w_i \\ &\quad + c_2 (q_2 + c'_2)^2 + c_3 q_3 + c_4 (q_4 + c'_4)^2 + c_6 q_6 + c_7 q_7 + c_9 \frac{q_9}{q_6 + c'_9} \\ &\quad + w_0 q_1 [p_1 k_1 (1 - q_3) + p_2 k_2 q_2 + p_3 k_3 q_1 + p_4 k_4 (1 - q_3)] + \text{fixed costs} \end{aligned} \quad (2.11)$$

where the terms have been reordered. The reliability constraints are given by (2.4) and (2.7):

$$\begin{aligned} \text{FR}_2(q_1, q_2, q_4) &= f_1 e^{f_2 q_1} + f_3 q_2 q_1^2 - f_4 (1 - q_1) q_4 < \text{FR}_{\max} \\ \text{SL}_2(q_5, q_6, q_7) &= s_1 \sqrt{q_5} e^{s_5 q_5} (1 - q_6)^{s_4} q_7^2 > \text{SL}_{\min} \end{aligned}$$

where FR_{\max} and SL_{\min} are specified by the user. One may also use other models proposed in Section 2.3. This objective function is quite complicated. However, by retaining only terms to leading order, we propose a simpler model $\widetilde{\mathcal{F}}$ which captures most of the behaviour:

$$\widetilde{\mathcal{F}}(\mathbf{q}) = c_1 e^{-c'_1 q_1} + c_5 e^{c'_5 q_5} + \widetilde{W}_1 q_1 + \widetilde{W}_2 (1 - q_5).$$

This simplification is justified numerically in Section 2.8.2.

2.7 Model Justification: Sensitivity to Parameters

Consider a function $\mathcal{G}(\mathbf{q}; f_1, f_2, \dots, f_N)$ depending on N continuously varying parameters on some domain Ω . We assume that \mathcal{G} is sufficiently regular to ensure that $\partial \mathcal{G} / \partial f_j$ exists throughout Ω . By considering the sequence of parameters $\{f_j\}_{j=1}^N$ as a vector $\vec{f} \in \mathbb{R}^N$ the total differential of \mathcal{G} can be written as

$$d\mathcal{G} = \nabla_{\vec{f}} \cdot d\vec{f} = \sum_{j=1}^N \frac{\partial \mathcal{G}}{\partial f_j} df_j.$$

As a consequence, assuming \mathcal{G} is positive valued⁷, one has

$$\frac{d\mathcal{G}}{\mathcal{G}} = \sum_{j=1}^N |f_j| \frac{\partial \log \mathcal{G}}{\partial f_j} \frac{df_j}{|f_j|} \quad (2.12)$$

illustrating that the proportion of the relative change in the \mathcal{G} due to the relative change in the parameter f_j is $f_j \partial \log \mathcal{G} / \partial f_j$. This simple formalism is used in the analysis that follows.

⁷The general expression is $\frac{d\mathcal{G}}{|\mathcal{G}|} = \sum_{j=1}^N |f_j| \text{sgn}(\mathcal{G}) \frac{\partial \log |\mathcal{G}|}{\partial f_j} \frac{df_j}{|f_j|}$.



2.7.1 Failure Rate Model

Recall that the second model for overall failure rate that we proposed in Section 2.3 was

$$\text{FR}_2(\mathbf{q}; f_1, f_2, f_3, f_4) = f_1 e^{f_2 q_1} + f_3 q_2 q_1^2 - f_4 (1 - q_1) q_4$$

where f_1, f_2, f_3, f_4 are positive constants. Because we have explicitly specified the dependence of FR on the parameters f_i expression (2.12) allows one to estimate which of these parameters have the most impact on the model. Indeed,

$$\frac{d\text{FR}_2}{|\text{FR}_2|} = \frac{(df_1 + f_1 q_1 df_2) e^{f_2 q_1} + q_2 q_1^2 df_3 - (1 - q_1) q_4 df_4}{|f_1 e^{f_2 q_1} + f_3 q_2 q_1^2 - f_4 (1 - q_1) q_4|}.$$

We claim that the effect of changing f_1 or f_2 by an amount δ has a larger effect than a similar change in f_3, f_4 . To see this, note that perturbations to f_1 and f_2 are amplified by a factor of $e^{f_2 q_1} > 1$ and $q_1 e^{f_2 q_1} \geq q_1$ respectively, whereas perturbations to f_3 and f_4 are only amplified by factors $q_2 q_1^2 \leq q_1 < 1$ and $(1 - q_1) q_4 \leq q_1 < 1$.

2.7.2 Severity Level Model

In order to gauge which of the free parameters most significantly affect the second SL model (2.7), we fix \mathbf{q} , and compute $\log(\text{SL}_2) = \log s_1 + (s_5 q_5 + \frac{1}{2} \log q_5) + s_4 \log(1 - q_6) + 2 \log q_7$, which helps us identify the relative sensitivity of the model to the parameters s_1, s_4 and s_5 :

$$\frac{d\text{SL}_2}{\text{SL}_2} = \frac{ds_1}{s_1} + s_5 q_5 \frac{ds_5}{s_5} + s_4 \log(1 - q_6) \frac{ds_4}{s_4}.$$

Thus, if all other parameters are fixed, a 1% change in s_5 will result in a $s_5 q_5$ % change in the SL value. If s_4 is changed by 1%, the resultant percentage change in the SL values is $s_4 \log(1 - q_6)$. Heuristically the q_6 values range between 0.4 and 0.7, and therefore $\log(1 - q_6)$ is negative. Increasing s_4 thus results in a decreasing SL. This also explains our findings regarding Table 2.1.

2.7.3 Cost of Quality Implementation Model

Following the same techniques as in the previous two subsections, we examine the model $C(\mathbf{q})$, given by equation (2.8) for the relative importance of the parameters

$$\vec{C} = \langle C_1, C'_1, C_5, C'_5, c_2, c'_2, c_3, c_4, c'_4, c_5, c_6, c_7, c_9, c'_9 \rangle.$$

Continuing with our prescription yields

$$\begin{aligned} dC(\mathbf{q}; \vec{C}) &= (dC_1 - q_1 C_1 dC'_1) e^{-C'_1 q_1} + (dC_5 + q_5 C_5 dC'_5) e^{C'_5 q_5} \\ &+ (q_2 + c'_2) [2c_2 dc'_2 + (q_2 + c'_2) dc_2] + (q_4 + c'_4) [2c_4 dc'_4 + (q_4 + c'_4) dc_4] \\ &+ q_3 dc_3 + q_6 dc_6 + q_7 dc_7 + \frac{c_9 q_9}{(q_6 + c'_9)^2} \left[(q_6 + c'_9) \frac{dc_9}{c_9} - dc'_9 \right]. \end{aligned}$$



2.8 Test Data, and Model Trends

The validation of a proposed model is an important step in any modelling effort. In the absence of real data from Lucent, we were unable to specify the nature of distributions from which to generate test data. Indeed, test data should be created on the basis of *hazard models* appropriate for the products. In the absence of these models, any test data used is for illustration purposes only.

As part of this project, we provide two test sets of data, drawn from a normal distribution (the mathematical interpretation of the popular Six-Sigma model) and from a Beta distribution. To illustrate the broad trends of our models, we generated several instances of \mathbf{q} , with individual attributes picked as independent random variables drawn from these two models.

2.8.1 Test Data Drawn from a Normal Distribution

Test data of 1000 instances of \mathbf{q} was created by considering the attributes q_i as independent random variables drawn from appropriate normal distributions $N(\mu_i, \sigma_i^2)$ (mean μ_i , variance σ_i^2), with the distribution parameters chosen to reflect a high quality product. Figure 2.7 illustrates a particular instance of this process. Note that the attributes q_i are scaled to reflect the *natural* quantities, e.g., component failure rate q_1 is scaled by 100 to yield a failure rate percentage. We input our simulated data \mathbf{q} into the failure rate model FR. The results are described in Figure 2.8.

We expect that with most products being built to high quality specifications, the warranty costs will be low, while the cost of implementation will be high. Figure 2.9 illustrates the histograms of the warranty costs W_{sw} , W_{hw} , and the implementation costs $C(\mathbf{q})$ when this instance of test data is applied to each of the relevant models.

2.8.2 Test Data Drawn from a Beta Distribution

A beta distribution was chosen because it is a two parameter distribution defined on the interval $[0, 1]$. The probability distribution function is given by

$$p(x; \alpha, \beta) = \begin{cases} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} & 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

where the mean and variance are

$$\mu = \frac{\alpha}{\alpha + \beta}, \quad \sigma^2 = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$

So as to match with the corresponding normal distribution $N(\mu_i, \sigma_i^2)$ one chooses the parameters α and β as

$$\alpha = \frac{\mu_i}{\sigma_i^2} (\mu_i - \mu_i^2 - \sigma_i^2), \quad \beta = \left(\frac{1}{\mu_i} - 1 \right) \alpha.$$

Once again test data of 1000 instances of \mathbf{q} was created and the simulated data applied to the failure rate model FR. The results are described in Figure 2.10. Since we assumed the attributes were drawn from a beta distribution (most instances are high quality), it is reasonable that the failure rate is skewed



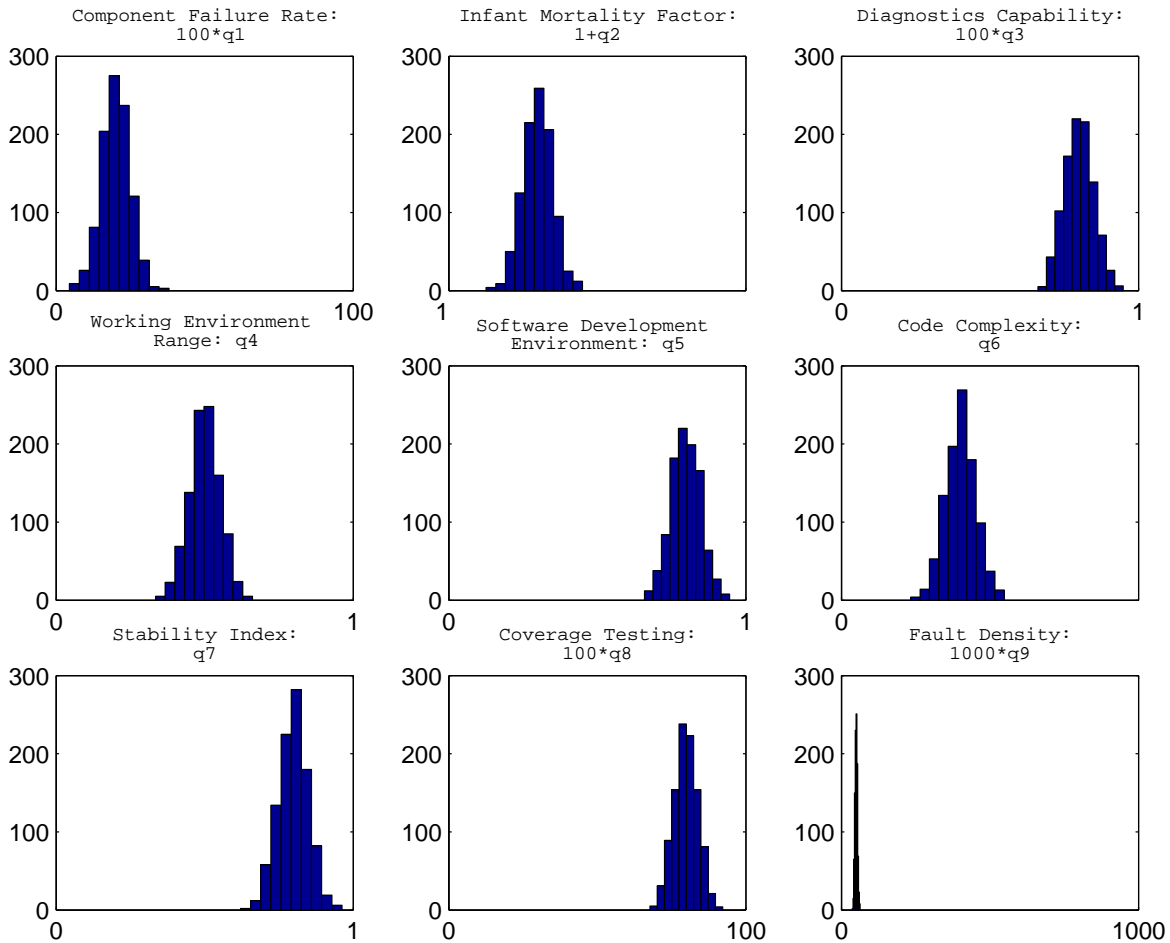


Figure 2.7: Simulated data drawn from normal distributions. In detail, $q_1 \sim N(0.2, 0.05)$, $q_2 \sim N(0.3, 0.05)$, $q_3 \sim N(0.8, 0.05)$, $q_4 \sim N(0.5, 0.05)$, $q_5 \sim N(0.8, 0.05)$, $q_6 \sim N(0.4, 0.05)$, $q_7 \sim N(0.8, 0.05)$, $q_8 \sim N(0.8, 0.04)$, $q_9 \sim N(0.05, 0.004)$. Any normalization to the respective variables is indicated.

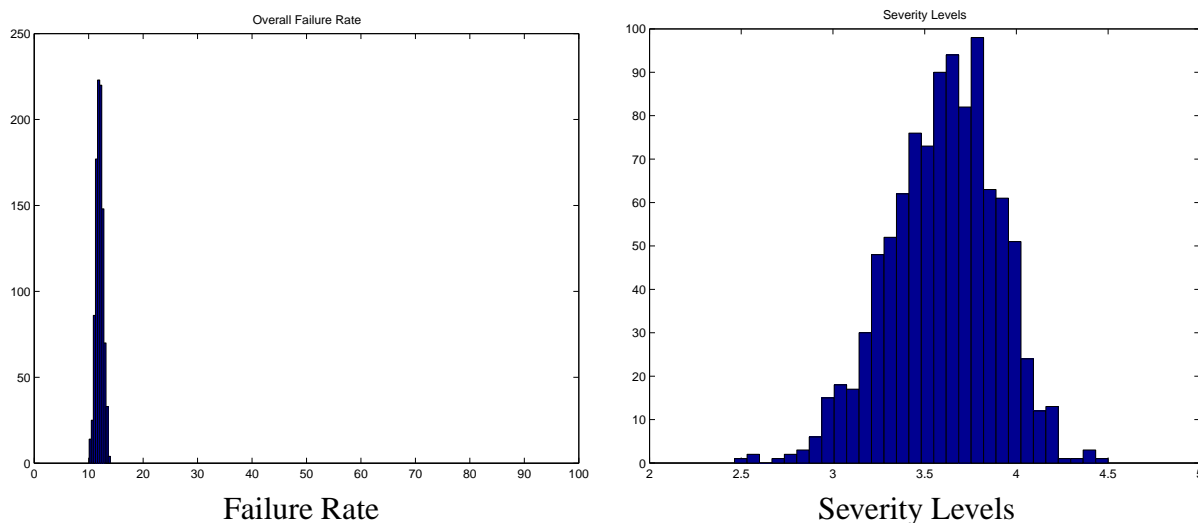


Figure 2.8: Effect of FR_2 and SL_2 on simulated data using a normal distribution: most instances of the product have a low failure rate.

towards the lower end. We expect that with most products being built to high quality specifications the warranty costs will be low while the cost of implementation will be high. This is borne out in Figure 2.11.

In Figure 2.12 we see that at least for this test data, the simplified objective function \tilde{F} described in Section 2.6 captures most of the behaviour of the complicated objective function (2.11).

2.9 Summary, Future Directions and Suggestions

We conclude this report by noting again that the models developed were based solely on discussions and heuristic arguments. In the absence of data, survival and hazard models, indeed even product information from Lucent, this report should not be interpreted as representative. Instead, we hope that the arguments will provide the basis for a more careful modelling effort by Lucent.

We note that despite the identification of nine quality attributes q_i , not all of these attributes are equally important. This is a crucial step in any modelling process: identifying the key elements. Based on the preceding discussion we can conclude that the most significant *independent* attributes are the *component failure rates* q_1 and the *software development environment* q_5 . These indices seem to outweigh the others. In fact, most of the other attributes are affected by these two. Thus, any further work should focus on the careful estimation of these attributes. The attributes associated with diagnostics capability q_3 , coverage testing q_8 and fault density q_9 are the least significant. Indeed, these do not even appear in the constraints.

Neither the constraint functions FR and SL nor the objective function contain complicated functional forms; the resultant model is nonlinear and awkward, but none of the individual components is more complicated than a quadratic or an exponential. These forms are deliberately chosen since the associated parameters can be easily fit, using real data and standard statistical software.

We suggest that the parameters be located based on true data which may be available to the industry.



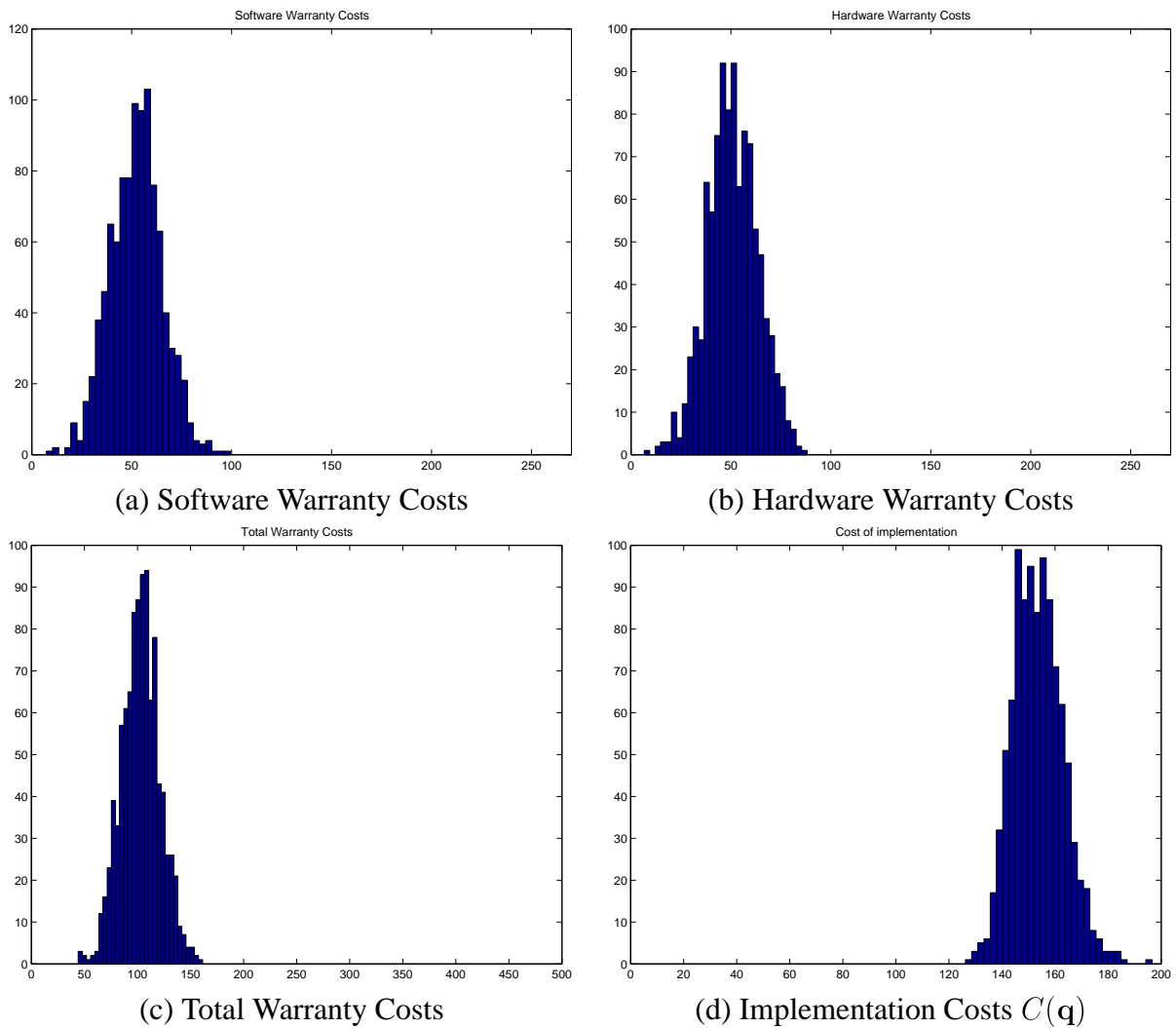


Figure 2.9: Warranty Costs and Implementation Costs using normally distributed test data. Scales range from 0 to maximum possible cost in each case.

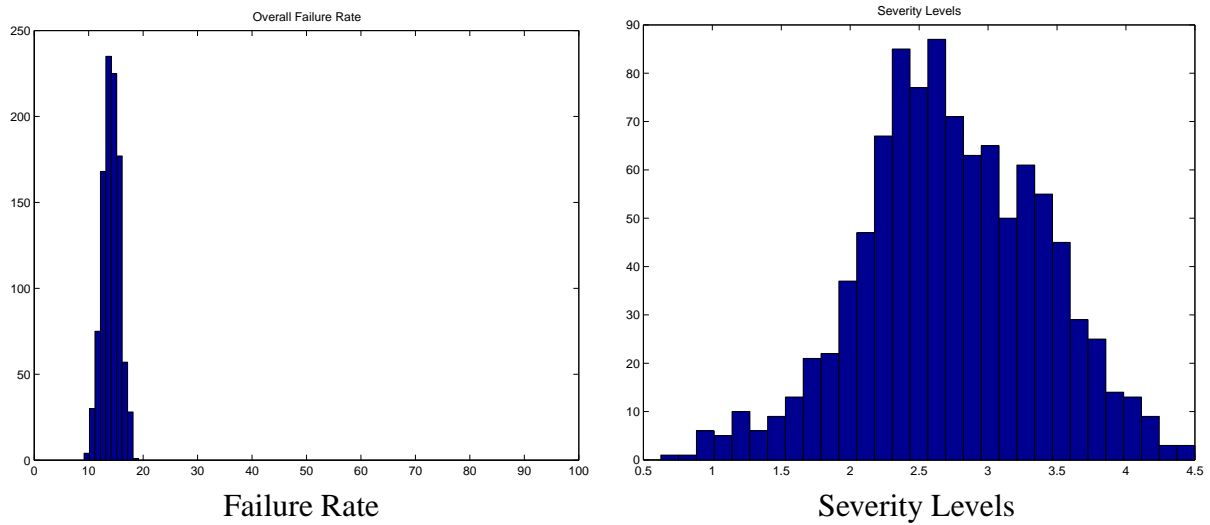


Figure 2.10: Effect of FR_2 and SL_2 on simulated data using a beta distribution: most instances of the product have a low failure rate.

These parameters will vary with various product and warranty policies. Simultaneously, test data drawn from hazard models appropriate to the specific product should be used as a reality check. The final optimization can be carried out using standard packages.

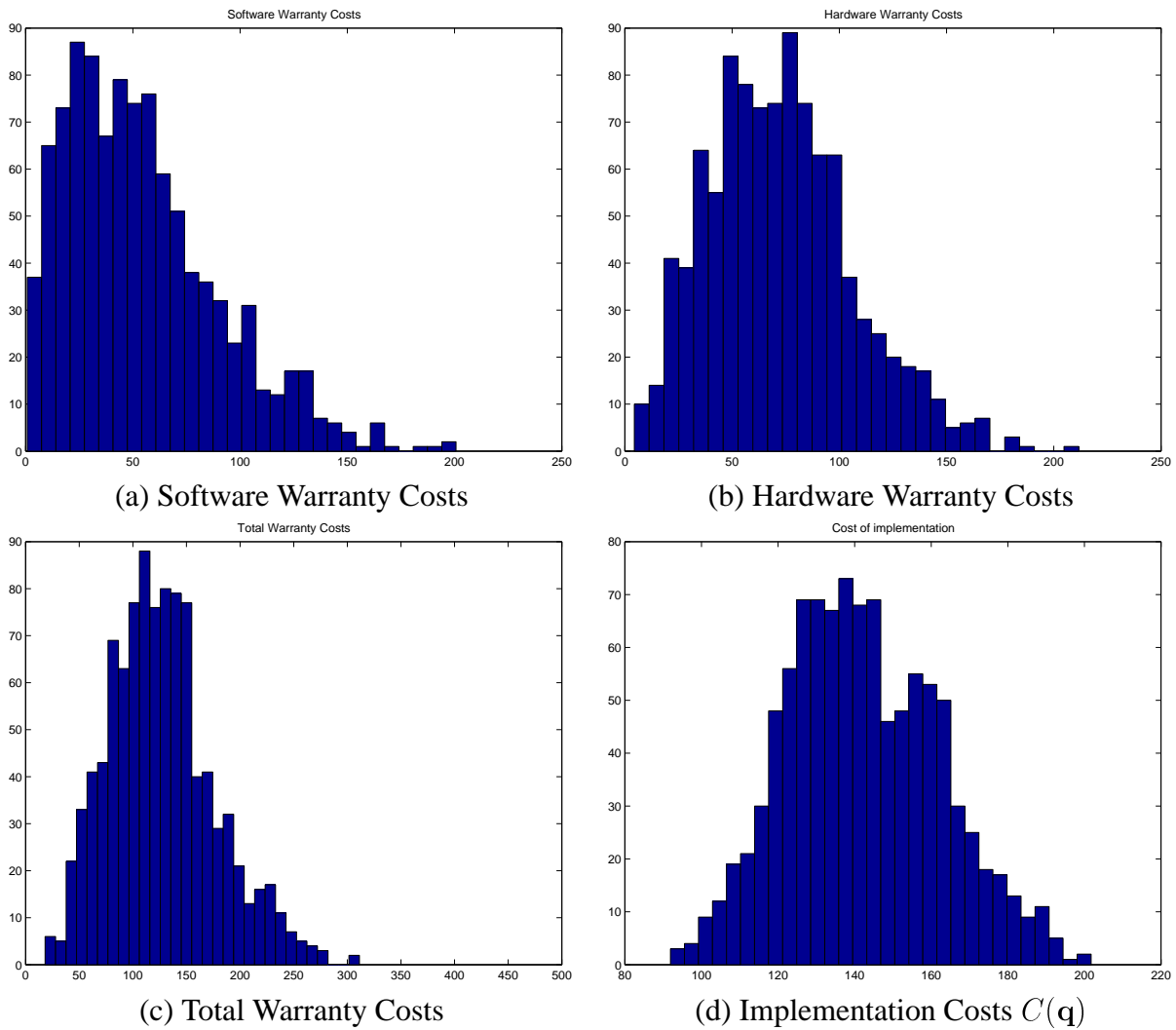


Figure 2.11: Warranty Costs and Implementation Costs using beta distributed test data. Scales range from 0 to maximum possible cost in each case.

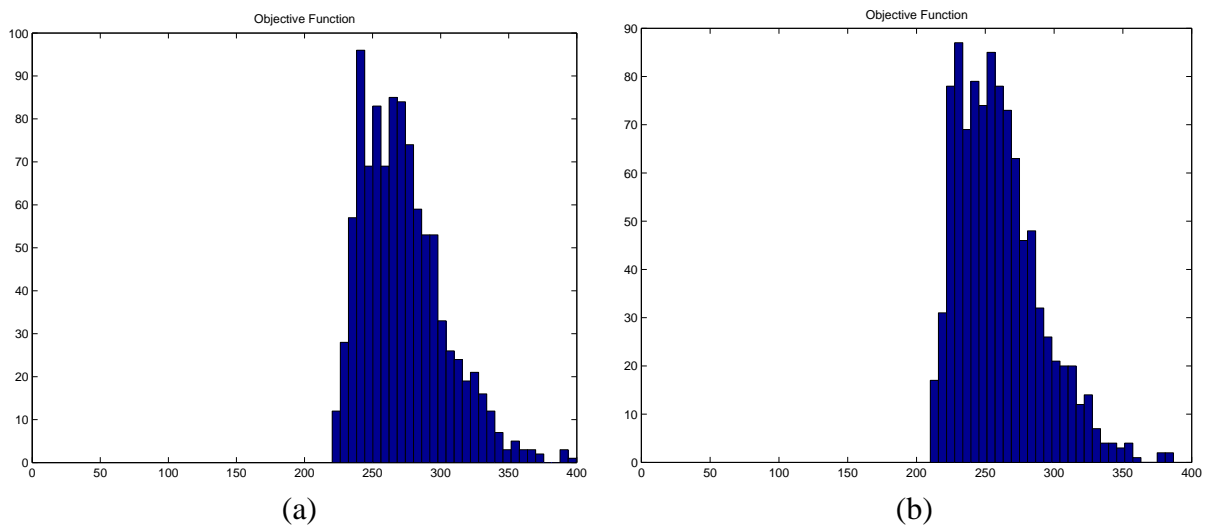


Figure 2.12: (a) Objective function $\mathcal{F}(\mathbf{q})$. (b) Simplified objective function $\tilde{\mathcal{F}}(\mathbf{q})$.



Bibliography

- [1] Mendiratta, V.B. (2002). Modeling Quality and Warranty Costs.
<http://www.pims.math.ca/industrial/2003/ipsw/Lucent.pdf>