你的产品可靠吗？

提高产品可靠性的50种方法

MIKE SILVERMAN

“How Reliable is Your Product?” Book Excerpt

50 Ways to Improve Product Reliability

By Mike Silverman
Foreword by Patrick O'Connor

superstar press

E-mail: info@superstarpress.com
20660 Stevens Creek Blvd., Suite 210
Cupertino, CA 95014

BOOK EXCERPT Table of Contents

Foreword by Patrick O'Connor
序 Patrick O'Connor

Modern engineering products, from individual components to large systems, must be designed, developed, and manufactured to be reliable in use. Designs must be robust in relation to the stresses and other factors that could cause damage or deterioration in transport, storage, use, and maintenance. Product development must include testing to ensure that this is achieved and to show up weaknesses for correction. The manufacturing processes must be performed correctly and with the minimum of variation. All of these aspects impact the costs of design, development, manufacture, and use, or, as they are often called, the product's life cycle costs. The challenge of modern competitive engineering is to ensure that life cycle costs are minimized while achieving requirements for performance and time to market. If the market for the product is competitive, improved reliability can generate very strong competitive advantages, as well as cost savings to manufacturers and to users. Today, this message is well understood by most engineering companies that face competitive pressures.

现代工程产品（无论是单个零件还是大型产品），都必须在设计、开发和生产过程中使用可靠性。在产品运输、储存、使用和维修中，由于应力和其它因素，会引起产品损坏或劣化，因此，设计必须稳健。产品开发必需包括测试环节，以确保产品可靠、显示产品缺陷并进行矫正。生产过程的实施必须正确，并尽可能减少变化。所有这些方面都影响设计、开发、生产和使用成本 。人们往往把这些成本叫产品生命周期成本。现代竞争工程的挑战要求生命周期成本最小，同时还应满足性能要求和上市时间要求。如果产品在市场上很有竞争性，那么可靠性的提高可以产生很强的竞争优势，同时还可以减少生产成本或使用成本。现在，面临竞争压力的工程企业可以很好地理解这一点。

The customers for major systems, particularly the U.S. military, drove the quality and reliability methods that were developed in the West from the 1950s onwards. They reacted to perceived low achievement by the imposition of standards and procedures. The methods included formal systems for quality and reliability management (MIL-Q-9858 and MIL-STD-758) and methods for predicting and measuring reliability (MIL-STD-721, MIL-HDBK-217, and MIL-STD-781). MIL-Q-9858 was the model for the international standard on quality systems (ISO9000). The methods for quantifying reliability were similarly developed and applied to other types of products and have been incorporated into other standards such as ISO60300. The application of these approaches has been controversial and not always effective.

自二十世纪 50 年代以来，大产品的客户（尤其是美国军队）推动了西方国家对质量和可靠性方法的开发。通过应用标准和程序，他们对他们认为的低可靠性做出反

应。这些方法包括质量和可靠性管理的形式系统（MIL-Q-9858 和 MIL-STD-758）以及可靠性预测和测试方法（MIL-STD-721、MIL-HDBK-217 和 MIL-STD-781）。MIL-Q-9858 是质量系统（ISO9000）的国际标准模型。与此相似，人们也给其它类型的产品开发和使用了量化的可靠性方法，这些方法已被纳入其它标准，如 ISO60300。这些方法并不总是有效，人们对他们的使用有争议。

In contrast, the Japanese quality movement that began in the 1950s was led by an industry that learned how manufacturing quality provided the key to greatly increased productivity and competitiveness, principally in commercial and consumer markets. The methods that they applied were based upon understanding of the causes of variation and failures, as well as continuous improvements through the application of process controls and motivation and management of people at work. It is one of history's ironies that the foremost teachers of these ideas were Americans, notably P. Drucker, W.A. Shewhart, W.E. Deming, and J.R Juran. The Japanese also applied methods for design for reliability, notably Quality Function Deployment (QFD) and failure modes and effects analysis (FMEA).

于此相反，始于上个世纪 50 年代的日本质量运动由一个行业引领，该行业知道，生产质量对大量提高生产力和竞争力非常关键，尤其在商业和消费者市场。他们所使用的方法来源于对变异和故障的理解，以及通过流程控制使用而获得的持续改进。具有讽刺意味的是，这些方法的先驱是美国人，如 P. Drucker、W.A. Shewhart、W.E. Deming 和 J.R Juran。日本人也为可靠性的设计提供了一些方法，如“质量功能部署（QFD）”和“失效模式和影响分析（FMEA）”。

By the turn of the century, methods of design for reliability and for manufacturing quality excellence had become refined. Most of the U.S. military standards were discontinued. More practical and effective methods were applied almost universally, particularly by industries whose products faced international competition or other drivers, particularly high costs of failures or strict customer requirements. However, some still cling to unrealistic mathematical precision for predicting and measuring reliability, as well as to bureaucratic approaches to quality management.

本世纪之初，可靠性设计方法和卓越生产质量设计方法变得很完善。大多数军队标准停止使用了。更实用有效的方法被广泛实用，尤其被面临国际竞争、故障率成本高、或客户要求严格的行业所使用。然而，在预测和衡量可靠性时，有些行业依然坚持不切实际的数学精确性，以及官僚的质量管理方法。

In the same time frame, there have been improvements in design capabilities with advances in computer-aided engineering, as well as in materials and in manufacturing processes. We have seen dramatic improvements in the reliability of products as diverse as automobiles, telecommunications, domestic equipment, and spacecraft. How many readers have experienced a failure of a microprocessor or an automobile engine?

同时，随着电脑辅助工程、材料和生产流程的进步，设计能力也得到了提高。在汽车、通讯、家用设备、及航天器行业，我们已经看到产品可靠性的巨大提高。因此现在很少有人遇到过微处理器或汽车发动机故障。

I am pleased to endorse and recommend this new book. Mike Silverman presents a wealth of practical, experienced-based wisdom in a way that is easy to read and apply. He has avoided detailed descriptions of methods, emphasizing instead the management and team aspects of applying cost-effective reliability improvement tools in ways that work.

我很高兴向大家推荐这本书。Mike Silverman 从经验中得到的智慧实用、易懂。他避免繁琐的细节描述，而是强调使用经济有效的可靠性提高工具的管理和团队方面。

The main methods he covers include reliability planning, design techniques (FMEA, fault tree analysis), test—particularly highly accelerated life test (HALT), and design of experiments, as well as methods for reliability prediction, stress derating, vendor reliability, failure reporting and analysis, and others. The whole product life cycle is considered, from initial design through prototype test, manufacturing, and field service, to obsolescence. He emphasizes the need for integration of reliability efforts to ensure their effective application. The fifty chapters all include brief case histories that illustrate this.

他介绍的方法包括可靠性计划、设计方法（FMEA、故障树分析）、高加速寿命测试（HALT）、试验设计、可靠性预测、应力降额、供货商可靠性、故障报告和分析等。涵盖了整个产品生命周期：从初始设计到样机测试、生产、现场服务、到报废。他强调为确保可靠性的有效应用，应该对可靠性进行整合。每章都用一个小故事来阐释这一点。

I recommend the book as an excellent guide for engineering project management and their teams, as well as for reliability specialists. It demystifies the sometimes difficult methods and helps specialists to communicate with managers, designers, and other engineers. It will make your products more reliable. November 2010

我建议将本书作为工程项目管理层、其团队以及可靠性专家的指南。它对困难的方法进行阐释 ，帮助专家与管理人员、设计师及其他工程师进行交流。它会让你的产品更可靠。2010 年 11 月。

Table of Contents

NOTE: This is the Table of Contents (TOC) from the book for your reference. The eBook TOC (below) differs in page count from the tradebook TOC.

目录表

Part 4: Prototype Phase
第 4 部分 样机阶段

Part 5: Manufacturing Phase
第5部分 生产阶段

Preface: Why Am I Writing This Book?

前言：我为什么写这本书？

I've read many reliability and quality textbooks, and very few approach reliability from the practical perspective. Instead, these books are filled with theory and formulas. However, many engineers are starting with almost no knowledge on the subject of reliability; they are in need of some basic education, but even more, they need the benefit of some practical experience and guidance. I wrote this book as a helpful guide, and I targeted the book at engineering professionals around the world in need of a practical guide to reliability.

我读过许多可靠性和产品质量方面的教材，可几乎没有任何一本书从实践的角度来解读可靠性。相反，这些书中满是理论和公式。而许多工程师开始时几乎不了解可靠性。他们需要一些基本知识，需要一些实用经验和指导。我希望本书成为一本有益的指南，目标读者是希望获得可靠性实践指导的工程师。

I wrote this book based on my 25 years' practicing reliability, including 10 years running a reliability test lab and 10 years running a reliability consulting firm called Ops A La Carte®. I started Ops A La Carte® because I saw the need to teach and help companies develop reliability programs. Most engineers I come across know basic concepts and have their favorite reliability techniques, but few understand how to put this into an overall reliability program.

根据 25 年的可靠性实践经验，包括 10 年经营可靠性实验室和 10 年经营可靠性咨询公司（Ops A La Carte®）的经验，我写成了这本书。我发现许多公司在制定可靠性方案时，需要培训和帮助，于是我成立 Ops A La Carte®。我见过的大多数工程师了解一些基本概念，并有最喜欢的可靠性方法，但很少有人知道如何把它们写进总可靠性方案。

Ops A La Carte® has worked with over 500 different companies in over 100 different industries in 30 different countries, so we have the ability to provide guidance from the experiential point of view. When I use the collective term "we" in this book, I am referring to an experience we have had at Ops A La Carte®.

Ops A La Carte®与 500 多家公司合作过，这些公司来自 30 个国家的 100 多个行业。因此我们有能力从实践的角度提供指导。本书中的“我们”指 Ops A La Carte®。

Just like any other discipline, there is no substitute for experience. Book knowledge is a good start, but until you are working on a design program or faced with a particular failure situation, you may not know what to do, or you may panic and resort to ineffective techniques you used in the past. In this book, I will show you different techniques and give you real-life situations that we faced and how we used particular techniques to solve problems.

与其它学科相同，在可靠性领域，经验的作用是不可取代的。可以从书本上获得可靠性知识，但在真正涉及到可靠性设计方案或具体故障的时候，你不知道该怎么办，因而或许会惊慌失措，或许会再次使用以前使用过的不起作用的方法。本书将介绍各种可靠性方法，列举一些我们遇到过的真实的情况，以及我们如何使用具体方法来解决问题。

I saw a movie recently called *Eagle Eye* that is quite applicable to reliability. The movie starts with the discovery of a possible terrorist plot. The Joint Chiefs of Staff of the United States consult their new supercomputer "Eagle Eye" to determine the probability that the terrorist plot is real. The supercomputer comes back with a probability of 39%. The commander in charge responded by saying, "39% and probability don't belong in the same sentence." (So true…and very appropriate for reliability as well.) In the next scene, the Joint Chiefs consult with the President, and by this time, "Eagle Eye" collects a bit more information and raises its probability to 51%. The President then decides to take action based on this and authorizes an attack on the alleged terrorist group. What is that really telling us? In fact, a probability of 51% means that there is a 49% chance that the conclusion is incorrect based on the data.

有一部叫《鹰眼》的电影对可靠性很适合。电影以发现一个可能的恐怖阴谋而开始。“美国参谋长联席会议”向新型超级电脑“鹰眼”咨询，以确定恐怖阴谋的真实性。“鹰眼”得出的结论是 39%。总司令对此结果的反应是，“39%”。（可靠性领域也是如此。）。接下来，和总统一起咨询，这次，“鹰眼”收集了更多的数据，得出的可能性是 51%。于是总统决定据此采取措施，袭击该恐怖组织。这个故事告诉我们什么？事实上，51%的可能性意味着该结论错误的可能性是 49%。

Likewise, with reliability tests, you need to make decisions based on test data from a sample of the population. You will never have enough data to be 100% certain of any decision, so you should gain as much confidence as you can with the time and money that you have. That is the art of reliability testing.

同样，对于可靠性测试，需要根据样品的测试数据作出决定。因为永远都不可能得到充足的数据来得出 100%正确的结论，因此，应该根据可利用的时间和资金获得尽可能大的可信度。这就是可靠性测试的艺术。

I structured the book in 50 easy-to-read chapters. Each chapter has some background on the reliability technique, its usefulness, and in some cases, its limitations. In addition, when applicable, I compare the technique in question to other techniques to show you when to use which technique. Starting in Chapter 3, I introduce the topic of Reliability Integration, and for each chapter onwards, I comment on how you can use the concept of Reliability Integration with that particular technique. I will talk a lot about Reliability Integration. It is one of the most valuable takeaways from this book.

本书内容浅显易懂。共包括 50 章，每章都有可靠性方法背景知识、有效性、实例举例、及其局限性。而且，我对正在讨论的方法和其它方法进行了比较，从而显示何时使用该方法。从第 3 章开始，每章都有“可靠性整合”。在这一部分，我会阐释如何对该章讨论的方法进行“可靠性整合”。“可靠性整合”是本书中最有价值的知识点之一。

In each chapter, I will provide one or more case studies from clients we have worked with and discuss how we utilized the specific technique in question. I didn't use the names of people or companies, but all of the case studies are real.

每一章都会提供一个或多个实例分析，来讨论如何使用可靠性方法。该部分没有提及公司名称，但所有这些例子都是真实的。

Tips on how to best use this book:

阅读须知：

- If a phrase is highlighted in bold italics, that means the term is a main technique of that chapter and is in the table of contents as well as the index. I will also capitalize the phrase throughout the rest of the book as an indication that it is an important technique. For all other important terms, check the index for other places I have used the same term.
- 用黑体字表示的术语是每章讨论的主要方法。在书中其它地方，这些术语每个单词的首字母都是大写的，从而表明这些方法很重要。可以使用索引找到在书中使用过这些方法的地方。
- I included a guide to acronyms. The field of reliability uses a lot of acronyms and I know how frustrating it can be reading a book filled with them.
- 书中还有首字母缩略词索引。可靠性行业使用许多缩略词，如果不提供这样的索引，我知道读一本充满缩略语的书是很头疼的一件事。
- I included a glossary of terms.
- 书中还有术语表

If you feel I missed something or you have information to add to a particular topic, I'd love to hear from you. I hope you enjoy my book.

如果书中有遗漏或不足之处，欢迎来信就某个具体话题进行讨论。希望您能喜欢本书。

Chapter 3: Reliability Integration Provides Integrity
第 3 章: 可靠性整合提供完整性

Reliability Integration is the process of seamlessly and cohesively integrating reliability techniques together to maximize reliability at the lowest possible cost. What this means is you should think of your reliability program as a set of techniques that are used together rather than just a bunch of individual activities.

可靠性整合是指无缝地、紧密地把不同可靠性方法融合在一起，从而以最小的成本得到最佳可靠性。也就是说，可靠性方案是几种方法协调使用的整体，而不是一堆无序的可靠性任务。

You are building a system, and a system is made up of different components and assemblies; there are different disciplines involved (some of the main disciplines are electrical, mechanical, software, firmware, optical, and chemical). All of the individual pieces make up the system, so don't forget about the interactions, and make sure that you think of the reliability from a system perspective. In Figure 3.1, we illustrate this point using the disciplines of electrical, mechanical, and software.

产品由各种部件和组建构成。产品生产涉及到的学科主要有：电子学、机械学、软件、固件、光学和化学。所有这些学科组成了产品这个系统。因此该重视这些学科之间的相互作用，并从整个产品的角度考虑可靠性（而不是从部件的角度）。表3-1 用电子学、机械学和软件学说明这一点。

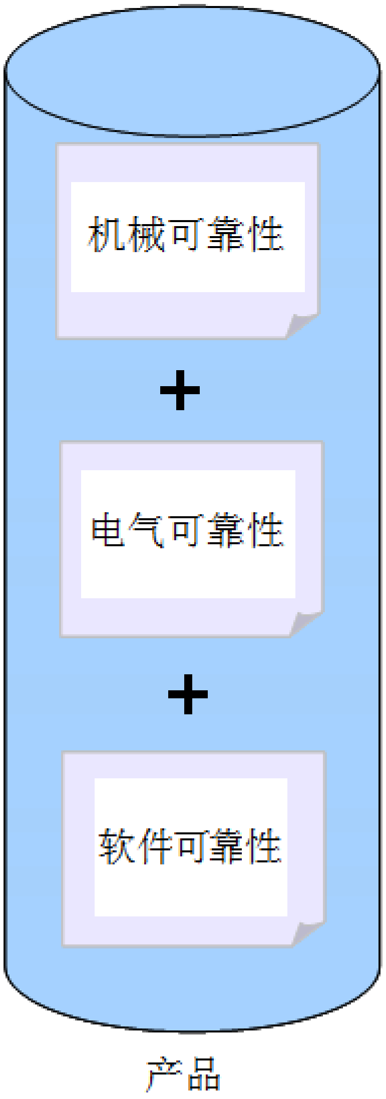

Figure 3.1: System View of Reliability

表 3.1： 产品级别的可靠性视图

This is especially true of software versus hardware disciplines. Most companies work on Software Reliability and Hardware Reliability separately and don't integrate the two. When failures occur, this then results in finger-pointing rather than synergy.

在软件和硬件方面尤为如此。在大多数公司里，软件可靠性和硬件可靠性是分别制定的，并没有进行整合。因此当故障发生时，人们只是互相指责，而不是互相协调。

This is equally true of electrical versus mechanical disciplines. We see more synergy between these two groups during programs than between software and hardware; however, at the beginning, they rarely get together to discuss common reliability goals and how to apportion them down to each major area of the system.

在电子学和机械学方面也是如此。与软件和硬件相比，电子部门和机械部门在产品开发过程中的协调使用更常见。然而，起初这两门学科的人几乎没有讨论过电子和机械的共同可靠性目标，及如何将这些目标在系统的主要领域进行分配。

Product development teams view reliability within each of the separate sub-domains of mechanical, electrical, and software issues. Your customers view reliability as a system-level issue, with minimal concern placed on the distinction between mechanical, electrical, and software issues. Your customer wants the whole product and all its parts to work together perfectly. Since the primary measure of reliability is made by your customer and their end users, engineering teams should maintain a balance of both views (system and sub-domain) in order to develop a reliable product.

产品开发部门从各个部件（如机械、电子和软件）出发考虑可靠性。客户从整个产品出发看待可靠性，而很少关注机械、电子和软件问题的差别。客户要的是完整的产品，并希望产品的各个部件在一起能正常运转。因为可靠性主要由客户和最终用户进行衡量，产品开发人员应该从产品和部件两个方面进行综合考虑，从而开发出可靠的产品。

3.1 Reliability versus Cost

Intuitively, the emphasis in reliability to achieve a reduction in warranty and in-service costs results in some minimal increase in development and manufacturing costs. Use of the proper techniques during the proper life cycle phase will help to minimize total life cycle cost (LCC).

3.1 可靠性与成本

通过提高可靠性来减少保修成本和使用成本，会造成开发成本和生产成本的少量增加。在生命周期的各个阶段使用适合的方法，会使生命周期的总成本（LCC）降至最低。

To minimize total LCC, your organization should do two things:

1. Choose the best techniques from all of the techniques available, and apply these techniques at the proper phases of the product life cycle.
2. Properly integrate these techniques by feeding information between different phases of the product life cycle.

降低生命周期的总成本可以采用以下方法：

1. 从可用的方法中选择最好的方法，并在适宜的生命周期阶段使用这些方法。
2. 在产品生命周期各个阶段，使用上个阶段获得的信息来正确整合这些方法。

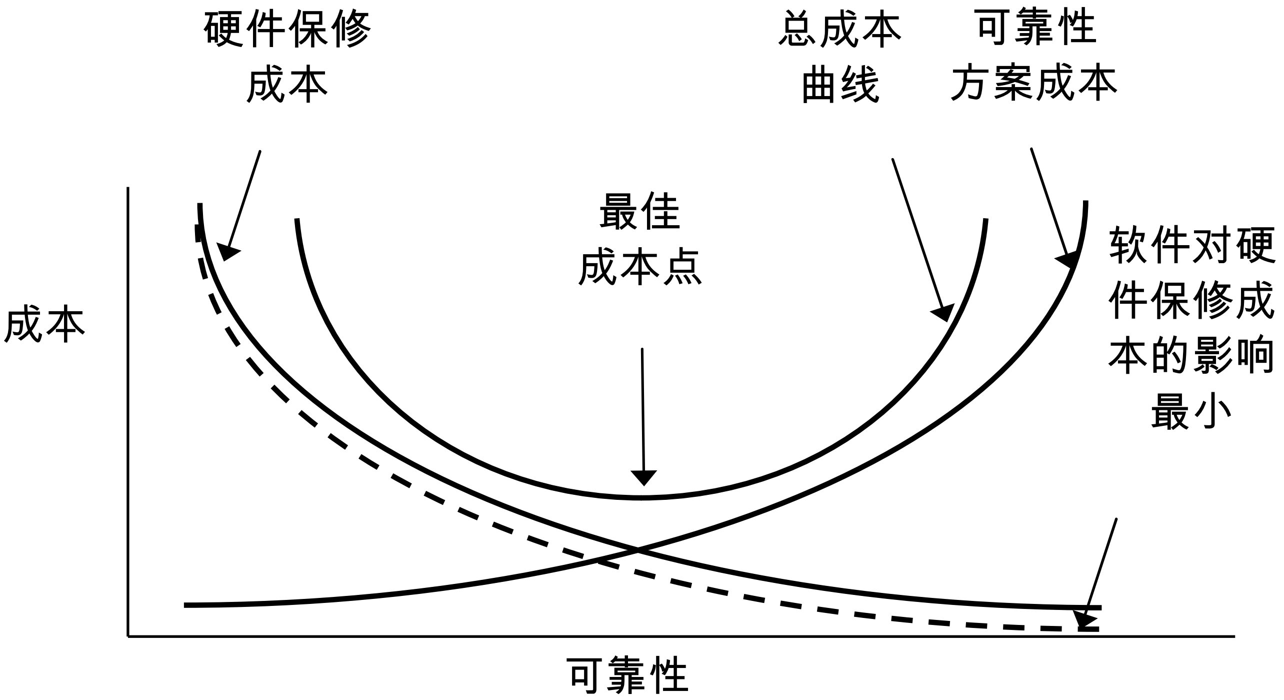

Figure 3.2: System Reliability versus Cost

图 3.2：产品可靠性与成本

In Figure 3.2, it is evident that:

1. Program costs go up as you spend more on reliability. At a certain point, you won't get your return on investment (ROI) because the reliability has reached a point where it is becoming increasingly more difficult to improve. That is why it is important to know what the goal is, and it can be just as detrimental to your company to produce a product that is too reliable as not reliable enough. The product that is too reliable usually comes with increased costs; your customers may not need this level of reliability and will opt for the less expensive product. When was the last time you purchased a $200 blender or toaster?
2. Warranty costs go up as reliability goes down.
3. Software has no associated manufacturing costs (other than perhaps the cost of CDs and manuals and the cost of personnel to test the product in production), so warranty costs and savings are almost entirely allocated to hardware. If there is no cost savings associated with improving Software Reliability, why not leave it as is and focus on improving hardware reliability to save money? You shouldn't do this for two reasons:

   - **a.** Our experience is that for typical systems, software failures outnumber hardware failures by a ratio of 10:1 (see Section 31.1 for more details). Customers buy integrated systems, not just hardware.
   - **b.** The benefits for a Software Reliability program aren't in direct cost savings. Instead, the benefits are in:
     - **i.** Increased software/firmware staff availability with reduced operational schedules, resulting in fewer corrective maintenance events.
     - **ii.** Increased customer goodwill based on improved customer satisfaction.

表 3.2 表明：

1. 增加可靠性成本会增加产品总成本。当可靠性提升到一定高度，很难再进一步提高时，投资报酬率不佳。因此弄清楚可靠性目标是什么很重要。产品的可靠性过高与过低都不好。产品的可靠性过高通常会增加成本。客户或许不需要这么高的可靠性，他们会选择较便宜的产品。你什么时候买过 200 美元的搅拌器或烤箱？

2. 可靠性下降时保修成本增加。

3. 除可能的 CD 成本、手册成本以及在开发过程中测试产品的人工费用之外，软件没有生产成本。因此保修成本和经费几乎全部用于硬件。如果提高软件可靠性与节约成本无关，为何不放弃软件可靠性，而只是提高硬件的可靠性，来节约成本？然而我们不能这样做，因为：

   - **a.** 事实证明普通产品的软件故障比硬件故障高出 10 倍（详细信息见 31.1 节）。客户购买的是产品，而不只是其中的硬件。
   - **b.** 软件可靠性方案的好处不在于直接节约成本，而在于：
     - **i.** 减少运作安排，减少故障维修事件，减轻软件/固件人员工作量。
     - **ii.** 提升客户满意度和商誉。

CASE STUDY: Linking Electrical, Mechanical, and Software Reliability Together

实例分析：整合电子、机械和软件可靠性

We were working with a semiconductor equipment company to help improve their reliability on their next generation product. First, we provided a Design for Reliability (DFR) seminar for each of the three different disciplines—the electrical group, the mechanical group, and the software group. Then, we met with the electrical, mechanical, and software team leads and developed reliability goals. We started with high level system goals and the apportioned the goals down to each subsystem—electrical, mechanical, and software.

我们曾帮助一个半导体设备公司提高新一代产品的可靠性。首先，我们为电子团队、机械团队和软件团队分别举行了 DFR 会议。然后我们与这三个团队的主管一起制定了可靠性目标。我们首先制定了产品可靠性目标，然后把目标分配到电子、机械和软件子系统。

Each group lead then took the goal for his subsystem and broke it down further within his area. We worked with each group lead to put together a reliability program plan to meet his subsystem goals. We rolled each of these different subsystem plans into an overall reliability plan for the product. We worked with each group lead to ensure he was on track for meeting his subsystem goals throughout the product development process. The end result was that our client was able to achieve reliability goals for each subsystem and for the system as a whole.

然后，在各自的子系统内，每个团队的主管负责对所分到的目标进行进一步分配。我们帮助每个团队主管制定了可靠性方案计划，以实现子系统目标。我们把这些子系统计划整合在一起制定出了产品的总可靠性计划。在产品开发过程中，我们和团队主管一起努力，使他们按计划达到各自的子系统目标。最后，他们既达到了子系统可靠性目标，也达到了产品可靠性目标。

Appendix A: Software Reliability Growth
附录 A: 提高软件可靠性

The following section of the book on Software Reliability Growth was provided by Mark Turner of Enecsys. This is referenced from Section 31.3.5 from the main section of this book.

本附录的内容*引用自本书的第 31 章 31.3.5 节*，由 *Enecsys 公司的* Mark Turner *提供。*

You can measure and manage the reliability growth of any software (or even hardware) development using an appropriate model, of which many exist, some of which are more suitable than others. Perhaps the most suitable for software development is the Rayleigh model.

可以使用适当的模型来测量并管理任何软件（甚至硬件）开发的可靠性增长。有很多这样的模型，其中有些比其它模型更适合，最适合软件开发的大概是 Rayleigh 模型。

Software design and development is a continuous process where you provide functionality using source code. Unfortunately, despite the best intentions of engineers, you may introduce defects as you create source code. Therefore, you will benefit from modeling the creation, identification, and elimination of code defects as a function of time.

软件设计和开发是一种连续的过程，它可以通过使用源代码来提供功能。让工程师们头疼的是，在创建源代码的同时，也有可能引进缺陷。因此，你将会因为创建、识别、以及消除以时间为函数的代码缺陷而获益。

Throughout the software development process, there are numerous opportunities for you to introduce defects. We provide a typical Rayleigh curve in Figure A.1, which illustrates the defect insertion and discovery process. This shows how you can identify and address defects that you introduce at an earlier phase in the software development. There will come a point in any software development program where you maximize the defect discovery rate, after which you reduce over time the quantity of remaining defects. Here the leading bar graph illustrates code defect insertion, which can often begin at the start of the development project. Because code defects are often related to the amount of engineering effort, the rate at which you introduce them often is directly proportional to that effort.

在软件开发过程中，有无数次可以引进缺陷的机会。图 A.1 中的典型 Rayleigh 曲线显示了缺陷的引入和发现过程。该图表明如何识别和处理在软件开发初期引入的

缺陷。在任何软件开发过程中，都有一个缺陷发现率最高的点，过了这个点之后，发现的剩余缺陷数量会减少。在图中，柱状图显示了代码缺陷的引入，缺陷的引入一般发生在项目的初期。因为代码缺陷往往和工程工作量有关，缺陷的引入率往往与该工作量成正比。

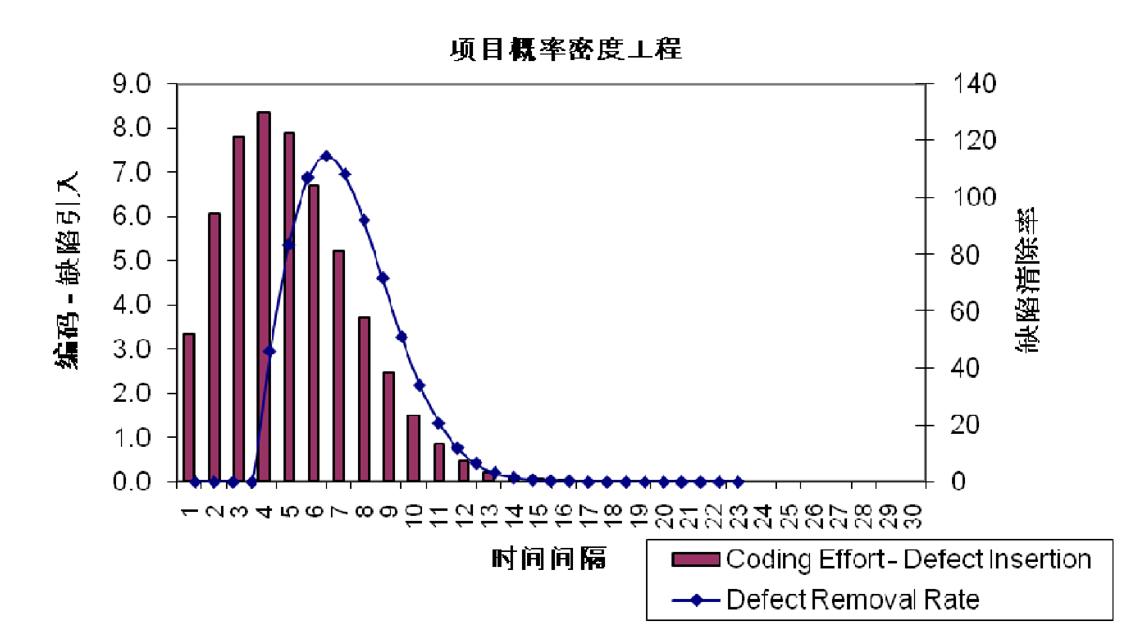

Figure A.1: Rayleigh Estimation Model with Effort and Defect Curves

*图 A.1：**Rayleigh 评估模型的工作量和缺陷曲线***

The lagging curve illustrates the defect removal rate, with problems being addressed at a later date than when you originally inserted them, which can hinder project progress and negatively impact customer satisfaction. You can partially mitigate this by conducting design reviews, code inspections, and early module testing, as these activities will often assist you in discovering inserted defects as early as possible, thus moving the defect discovery and correction curve to the left.

图中的滞后曲线显示了缺陷清除率。缺陷引入后，对缺陷的处理会阻碍项目进程，并会影响客户满意度。通过进行设计审核、代码检查、和早期模块测试，可以部分减轻这些负面影响，因为这些做法往往能帮助你尽早发现引入的缺陷，从而使缺陷的发现和校正曲线向左移。

Eventually, the quantity of defects still present in the code will equate to the original reliability target, and as you discover and address further defects, the Software Reliability increases, or grows. You can manage the rate at which you address defects by setting software defect targets. This has to begin by estimating how many defects are likely to occur, then addressing those defects by implementing a Software Reliability growth management program in which you plan and schedule the necessary resource to ensure you achieve your reliability target.

最后，代码中的缺陷数量应该等于原定可靠性指标，并且随着缺陷的进一步发现和清楚，软件可靠性会得到提升，或增长。你可以通过设定软件缺陷目标，对缺陷清除率进行管理。首先对可能出现的缺陷数量进行评估，然后通过实施软件可靠性增长管理方案来清除这些缺陷。在该方案中，应该对必要的资源做出计划，从而确保达到可靠性目标。

A.1 Implementing the Rayleigh Model

A.1 Rayleigh 模型的实施

You can use the Rayleigh function to forecast the rate at which you identify defects during the software development program as a function of time. It is a specific instance of one of the models in the Weibull family of reliability models.

在软件开发过程中，你可以使用 Rayleigh 函数来预测缺陷识别率。Rayleigh 模型是 Weibull 可靠性模型的一个具体模型。

This model is particularly suited to Software Reliability modeling as it provides a good representation of the vector sum of a large number of random sources of defects, none of which dominate. The Rayleigh model provides an effective iterative design process in which feedback is inherently part of the solution process, and in fact it closely approximates the actual profile of defect data that you collect during software development programs.

该模型尤其适合于软件可靠性的创建，因为它很好地表示了大量随机缺陷源的矢量和，没有任何一种缺陷源占主导地位。Rayleigh 模型提供有效的迭代设计过程，在这一过程中，Rayleigh 模型的反馈信息是问题解决过程中的一个部分，事实上，Rayleigh 模型非常接近在软件开发过程中收集到的真实的缺陷数据分析。

Monitoring software development defect metrics can provide you valuable input into planning engineering and Root Cause Analysis (RCA) efforts, and it helps you to quantify the maturity of the software you are developing. Collecting defect metrics related to engineering effort, project duration, and type over several development projects provides a great opportunity to analyze trends, which can then provide you with more accurate resource predictions for new projects. If you lack such trend data (which typically is a problem when you first deploy the Rayleigh model), then you may have to use industry data as an alternative guide. While this alternative approach may not factor in the abilities of your actual design team, it does at least provide a reasonable estimation to begin with.

对软件开发缺陷的衡量标准进行监控可以向规划工程和根因分析（RCA）提供有用信息，而且还能帮助你对正在开发的软件成熟度进行定量。收集与工程计划、工期、几个开发项目的种类有关的缺陷衡量标准可以提供很好的分析趋势的机会，而趋势分析可以向新项目提供更准确的资源预测。如果你缺少这些趋势数据（这是第一次使用 Rayleigh 模型时经常出现的一个问题），那么需要使用行业数据作为备选指南。虽然这一备选方法可能不会影响设计团队的真正实力，但它至少可以提供合理评估。

After your organization completes multiple projects, you will benefit from reviewing predicted versus actual defect counts, as this enables you to refine the original estimates and improve the model for future development projects. As the project progresses, compare the initial defect count estimates with the quantity of defects you actually address. If you find that the actual defect count is significantly higher than predicted, then the model has generated an early indication that a significant problem may exist. On the other hand, if you find that the actual defect count is significantly less than the initial prediction, then you should confirm that the identification process is indeed sufficient to detect the anticipated defects. Once you confirm this, then you can conclude that your defect insertion rate is actually less than predicted.

公司在完成多个项目之后，将会因为对预测缺陷数目进行评审，而不是对实际缺陷数目进行评审而获益。因为这会让原始估计精确，并为将来的项目开发而改善模型。随着项目的进行，对最初估计的缺陷数目和真正出现的缺陷数目进行对比。如果发现真正的缺陷数目远远大于预测值，那么该模型预示项目存在重大问题。如果发现发现真正的缺陷数目远远小于预测值，那么证明缺陷识别过程可以非常有效地检测出预期缺陷。一旦证明了这一点，就可以推断缺陷引入率低于预测值。

In using the Rayleigh model, you should determine the parameters for the total anticipated engineering effort, the total number of defects that you expect to insert into the code, and the time period to reach the peak estimate. Knowing these parameters will enable you to plot the cumulative distribution function (CDF). We've shown an example of a CDF Chart in Figure A.2 and a CDF Table in Figure A.3. The Rayleigh model parameters of Figure A.3 are 55 man-months of engineering effort, 755 inserted defects, and 4 months to reach the peak estimate. For the defect plot, you must define an additional value associated with the estimated lag behind the start of the project effort to account for defect detection and correction effort, which in Figure A.3 is 4 months.

在使用 Rayleigh 模型时，应该确定如下参数：总预期工程工作量、预计引入代码的总缺陷数量、以及达到估计峰值所需的时间。理解这些参数可以让你绘出累积分

布函数（CDF）。图 A.2 是一个 CDF 图，图 A.3 是一个 CDF 表。图 A.3 的 Rayleigh 模型的参数是 55 人月工程量、755 个引入缺陷、以及 4 个月达到估计峰值。对于缺陷规划，必须规定与估计延迟相关的附加值，从而计算缺陷检测和校正工作量，这在图 A.3 中是 4 个月。

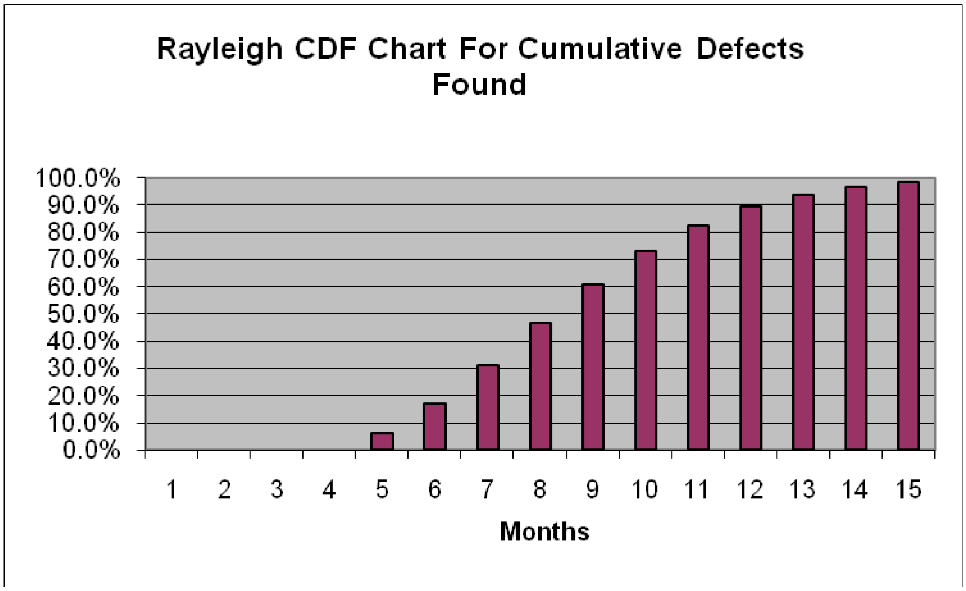

Figure A.2: Rayleigh Cumulative Distribution Function (CDF) Chart

图 A.2：Rayleigh 累积分布函数（CDF）图

Figures A.2 and A.3 illustrate the relationship between the project effort and the number of defects that you insert into the code, which enables you to make decisions regarding the impact that any code changes are likely to have and in changes to the code delivery date. From this example, you can conclude that a delivery schedule of eight months would be completely unrealistic, as a significant number of defects will still be present in your code, whereas a delivery schedule of twelve to fifteen months is more realistic. Delivery schedules in between require you to make a schedule versus reliability tradeoff.

图 A.2 和 A.3 显示了项目工作量和引入代码的缺陷数目之间的关系。这能够使你对任何代码变化以及代码发送日期变化所造成的可能影响做出决定。从这个例子中，可以推断 8 个月的发送计划根本无法实现，因为大量缺陷将仍然在代码中存在。然而 8 个月和 12 到 15 个月之间的计划更实际些。此发送计划需要基于可靠性综合标准制定计划。

Month	Effort	Inserted Defects	Cumulative defects found	CDF
1	3.3	0	0	0.0%
2	6.1	0	0	0.0%
3	7.8	0	0	0.0%
4	8.3	0	0	0.0%
5	7.9	79	79	10.5%
6	6.7	134	213	28.4%
7	5.2	153	366	48.7%
8	3.7	138	504	67.1%
9	2.5	106	610	81.2%
10	1.5	68	678	90.3%
11	0.9	39	717	95.5%
12	0.5	19	736	98.0%
13	0.2	8	744	99.1%
14	0.1	3	747	99.5%
15	0.0	1	748	99.6%
Total	55	748		

Figure A.3: Rayleigh CDF Table

图 A.3：Rayleigh CDF 表

However, if early delivery is unavoidable, the CDF can aid in planning reliability growth activities and managing customer expectations where multiple deliveries are viable.

然而，如果代码的早期发送不可避免，CDF 可以协助计划可靠性增长活动，并管理多代码发送情况下的客户期望。

About the Author

Mike Silverman Mike is founder and managing partner at Ops A La Carte LLC®, a professional consulting company that has an intense focus on helping clients with reliability throughout their product life cycle. Mike has over 25 years' experience in reliability engineering, reliability management, and reliability training. He is an experienced leader in reliability improvement through analysis and testing. Through Ops A La Carte®, Mike has had extensive experience as a consultant to high-tech companies. A few of the main industries are aerospace and defense, clean technology, consumer electronics, medical, networking, oil and gas, semiconductor equipment, and telecommunications. Most of the examples in this book have been taken from Mike's experiences.

关于作者 Mike Silverman Mike Silverman 是 Ops A La Carte LLC®公司的创始人兼股东。该公司位于美国硅谷，是一家专业咨询公司，致力于帮助客户在整个产品生命周期中提高可靠性。Mike 有 25 年以上的可靠性工程、可靠性管理和可靠性培训方面的经验。在通过分析和测试来提高产品可靠性方面，他具有资深地位。通过 Ops A La Carte®公司向高科技公司提供咨询服务，作为咨询师的 Mike 积累了极其丰富的经验。主要涉及的领域有：航空航天、国防、清洁技术、消费性电子产品、医疗、网络、石油和天然气、半导体设备和通讯。本书中的大多数例子都来自 Mike 的实际经验。

Mike is an expert in accelerated reliability techniques and owns HALT and HASS Labs®, one of the oldest and most experienced reliability labs in the world. Mike has authored and published 25 papers on reliability techniques and has presented these in countries around the world, including Canada, China, Germany, Japan, Korea, Singapore, Taiwan, and the USA. He has also developed and currently teaches over 30 courses on reliability techniques. Mike has a BS degree in electrical and computer engineering from the University of Colorado at Boulder, and is a Certified Reliability Engineer (CRE) through the American Society for Quality (ASQ). Mike is a member of the American Society for Quality (ASQ), Institute of Electrical and Electronics Engineers (IEEE), Product Realization Group (PRG), Professional and Technical Consultants Association (PATCA), and IEEE Consulting Society. Mike is currently the IEEE Reliability Society Santa Clara Valley Chapter Chair.

Mike 是加速可靠性技术方面的专家，并拥有世界上最早和最有经验的可靠性实验室 - HALT 和 HASS 实验室®。Mike 书写并出版了 25 份可靠性技术论文，并在世界上多个国家和地区就这些论文做了演讲，如加拿大、中国、德国、日本、韩国、新加坡、美国和台湾。他还开发并讲授 30 多们可靠性方法课程。Mike 在位于波尔得的美国科罗拉多大学获得了电气和计算机工程专业学士学位，是一位美国质量学会（ASQ）的注册可靠性工程师（CRE）。Mike 是美国质量学会（ASQ）、美国电气电子工程师协会（IEEE）、美国产品实现团体（PRG）、美国专业技术协会（PATCA）和美国 IEEE 咨询协会的会员。Mike 目前是 IEEE 可靠性协会圣克拉拉谷分会主席。

You can contact Mike via the Ops A La Carte® website at http://www.opsalacarte.com.

你可以通过 Ops A La Carte®公司的网站 http://www.opsalacarte.com 与 Mike 取得联系。

Getting “How Reliable is Your Product?” (**http://happyabout.com/productreliability.php**)

“How Reliable is Your Product?” can be purchased as an eBook for $19.95 or tradebook for $44.95 at http://happyabout.com/productreliability.php or at other online and physical book stores.

- Please contact us for quantity discounts at **sales@happyabout.info**
- If you want to be informed by email of upcoming Happy About® books, please email **bookupdate@happyabout.info**

Happy About is interested in you if you are an author who would like to submit a non-fiction book proposal or a corporation that would like to have a book written for you. Please contact us by email **editorial@happyabout.info** or phone (1-408-257-3000).

Other Happy About books available include:

- #LEADERSHIPtweet Book01: **http://www.happyabout.com/thinkaha/leadershiptweet01.php**
- #PARTNER tweet Book01: **http://www.happyabout.com/thinkaha/partnertweet01.php**
- Expert Product Management: **http://www.happyabout.com/expertproductmanagement.php**
- 42 Rules of Product Management: **http://www.happyabout.com/42rules/42rulesproductmanagement.php**
- 42 Rules to Increase Sales Effectiveness: **http://www.happyabout.com/42rules/increasesaleseffectiveness.php**
- 42 Rules for Driving Success With Books: **http://www.happyabout.com/42rules/books-drive-success.php**
- Agile Excellence for Product Managers: **http://www.happyabout.com/agileproductmangers.php**
- The Phenomenal Product Manager: **http://www.happyabout.com/phenomenal-product-manager.php**
- #QUALITYtweet Book01: **http://www.happyabout.com/thinkaha/qualitytweet01.php**
- Scrappy Project Management®: **http://www.happyabout.com/scrappyabout/project-management.php**
- #PROJECT MANAGEMENT tweet Book01: **http://www.happyabout.com/thinkaha/projectmanagementtweet01.php**

Through his 25 year career, Mike Silverman has maintained a singular focus on reliability. He is founder of and Managing Partner at Ops A La Carte, a reliability engineering consultancy that helps customers build end-to-end reliability into their products. He owns and operates HALT and HASS labs, a reliability laboratory in Northern California that has now tested over 500 products.

在25年的职业生涯中，Mike Silverman倾其全力于可靠性。他是Ops A La Carte公司的创始人和股东，该公司是一家可靠性工程咨询公司，致力于帮助客户在整个产品生命周期中提高产品可靠性。他的HALT和HASS可靠性实验室位于加利福尼亚北部，已经测试了500多种产品。Mike是一名注册可靠性工程师，已发表了十多份技术论文，目前是美国硅谷IEEE可靠性协会的会长。

SUPERSTAR press
http://superstarpress.com