Chapter 1

Design for Reliability Paradigms

Dev Raheja

WHY DESIGN FOR RELIABILITY?

The science of reliability has not kept pace with user expectations. Many corporations still use MTBF (mean time between failures) as a measure of reliability, which, depending on the statistical distribution of failure data, implies acceptance of roughly 50 to 70% failures during the time indicated by the MTBF. No user today can tolerate such a high number of failures. Ideally, a user does not want any failures for the entire expected life! The life expected is determined by the life inferred by users, such as 100,000 miles or 10 years for an automobile, at least 10 years for kitchen appliances, and at least 20 years for a commercial airliner. Most commercial companies, such as automotive and medical device manufacturers, have stopped using the MTBF measure and aim at 1 to 10% failures during a self-defined time. This is still not in line with users' dreams. The real question is: Why not design for zero failures if we can increase profits and gain more market share? Zero failures implies zero mission-critical failures or zero safety-critical system failures. As a minimum, systems in which failures can lead to catastrophic consequences must be designed for zero failures. There are companies that are able to do this. Toyota, Apple, Gillette, Honda, Boeing, Johnson & Johnson, Corning, and Hewlett-Packard are a few examples.

The aim of design for reliability (DFR) is to design-out failures of critical system functions in a system. The number of such failures should be

Design for Reliability, First Edition. Edited by Dev Raheja, Louis J. Gullo.

zero for the expected life of the product. Some components may be allowed to fail, such as in redundant systems. For example, in aerospace, as long as a system can function at least for the duration of the mission and the failed components are replaced prior to the next mission to maintain redundancy, certain failures can be tolerated. This is, however, insufficient for complex systems where thousands of software interactions, hundreds of wiring connections, and hundreds of human factors affect the systems' reliability. Then there are issues of compatibility [1] among components and materials, among subsystems, and among hardware and software interactions. Therefore, for complex systems we may find it impossible to have zero failures, but we must at least prevent the potential failures we know about. Since failures can come from unknown and unexpected interactions, we should try to design-in fallback modes for unexpected events. A "what-if" analysis usually points to some events of this type. To minimize failures in complex systems, in this book we describe techniques for improving software and interface reliability.

As indicated earlier, some companies have built a strong and long-lasting reputation for reliability based on aiming at zero failures. Toyota and Sony built their world leadership mostly on high reliability; and Hyundai has been offering a 10-year warranty and increasing its market share steadily. Progress has been made since then. In 1974, when nobody in the world gave a warranty longer than one year, Cooper Industries gave a 15-year warranty to electric power utilities on high-voltage transformer components and stood out as the leader in profitability among all Fortune 500 electrical companies. Raytheon has established a culture at the highest level in the corporation of providing customers with mission assurance through a "no doubt" mindset. Says Bill Swanson, chairman and CEO of Raytheon: "[T]here must be no doubt that our products will work in the field when they are needed" (Raytheon Company, *Technology Today*, 2005, Issue 4). Similarly, with its new lifetime power train warranty, Chrysler is creating new standards for reliability.

REFLECTIONS ON THE CURRENT STATE OF THE ART

Reliability is defined as the probability of performing all the functions (including safety functions) satisfactorily for a specified time and specified use conditions. The functions and use conditions come from the specification. If a specification misses or is vague 60% or more of the time, the reliability predictions are of very little value. This is usually the case [2]. The second big issue is: How many failures should be tolerable? Some readers may not agree that we can design for zero critical failures, but the evidence supports the contrary conclusion. We may not be able to prevent failures that we did not

foresee, but we can design out all the critical failure modes that we discover during the requirements analysis and in the failure mode and effects analysis (FMEA). In over 30 years' experience, I have yet to encounter a failure mode that cannot be designed-out. The cost is usually not an issue if the FMEA is conducted and the improvements are made during the early design stage. The time specified for critical failures in the reliability definition should be the entire lifetime expected.

In this chapter we address how to write a good system specification and how to design so as not to fail. We make it clear that the design for reliability should concentrate on the critical and major failures. This prevents us from solving easy problems and ignoring the complex ones. The following incident raises issues that are central to designing for reliability.

> *The lessons learned from the Interstate 35 bridge collapse in Minnesota on August 1, 2007 into the Mississippi River on August 1, killing 13, give us some clues about what needs to be done. Similar failure mechanisms can be found in many large electrical and mechanical systems, such as aircraft and electric power plants.*
>
> *The bridge was expanded from four lanes to six, and eventually to eight. Some wonder whether that might have played a role in its collapse. Investigators said the failure resulted because of a flaw in its design. The designers had specified a metal plate that was too thin to serve as a junction of several girders.*
>
> *Like many products, it gradually got exposed to higher loads, adding strain to the weak spot. At the time of the collapse, the maintenance crews had brought tons of equipment and material onto the deck for a repair job. The bridge was of a design known as a nonredundant structure, meaning that if a single part failed, the entire structure could collapse. Experts say that the pigeon dung all over the steel could have caused faster corrosion than was predicted.*

This case history challenges the fundamentals of engineering taught in the universities.

- *Should the design margin be 100% or 800%*? "How does the designer determine the design margin?"
- *Should we design for pigeons doing their dirty job?* What about designing for all the other environmental stressors, such as chemicals sprayed during snow emergencies, tornados, and earthquakes?
- *Should we design-in redundancy on large mechanical systems to avoid disasters?* The wisdom says that redundancy delays failures but may not avoid disasters. The failure could occur in both the redundant paths, such as in an aircraft accident where the flying debris cut through all three redundant hydraulic lines.
- *Should we design for sudden shocks experienced by the bridge during repair and maintenance?*

These concerns apply to any product, such as electronics, electrical power systems, and even a complex software design. In software, the corrosion can be symbolic for applying too many patches without knowing the interactions. Call it "software corrosion."

The answers to the questions above should be a resounding "yes." An engineering team should foresee all these and many more failure scenarios before starting to design. The obvious strategy is to write a good system specification by first predicting all major potential failures and avoiding them by writing robust requirements. Oversights and omissions in specifications are the biggest weakness in the design for reliability. Typically, 200 to 300 requirements are generally missing or vague for a reasonably complex system such as an automotive transmission.

Analyses techniques covered in this book for hardware and software help us discover many missing requirements, and a good brainstorming session for overlooked requirements always results in discovering many more. What we really need is perhaps the paradigms based on lessons learned.

THE PARADIGMS FOR DESIGN FOR RELIABILITY

Reliability is a process. If the right process is followed, results are likely to be right. The opposite is also true in the absence of the right process. There is a saying: "If we don't know where we are going, that's where we will go." It is difficult enough to do the right things, but it is even more difficult to know what the right things are!

Knowledge of the right things comes from practicing the use of lessons learned. Just having all the facts at your fingertips does not work. One must utilize the accumulated knowledge for arriving at correct decisions. Theory is not enough. One must keep becoming better by practicing. Take the example of swimming. One cannot learn to swim from books alone; one must practice swimming. It is okay to fail as long as mistakes are the stepping stones to failure prevention. Thomas Edison was reminded that he failed 2000 times before the success of the light bulb. His answer, "I never failed. There were 2000 steps in this process."

One of the best techniques is to use lessons learned in the form of paradigms. They are easy to remember and they make good topics for brainstorming during design reviews.

Paradigm 1: Learn To Be Lean Instead of Mean

When engineers say that a component's life is five years, they usually imply the calculation of the mean value, which says that there is a 50% chance of failure during the five years. In other words, either the supplier or the customer has

to pay for 50% failures during the product cycle. This is expensive for both: a lose–lose situation. Besides, there are many indirect expenses: for warranties, production testing, and more inventories to replace failed parts. This is mean management. It has a negative return on investment. It is mean to the supplier because of loss of future business and mean to the customer in putting up with the frustrations of downtime and the cost of business interruptions. Therefore, our failure rate goal should be *as lean as possible*. Engineers should promise *minimum life* to customers, not mean life. Never use averages in reliability; they are of no use to anyone.

Paradigm 2: Spend a Lot of Time on Requirement Analysis

It is worth repeating that the sources of most failures are incomplete, ambiguous, and poorly defined requirements. That is why we introduce unnecessary design changes and write deviations when we are in hurry to ship a product. Look particularly for missing functions in the specifications. There is often practically nothing in a specification about modularity, reliability, safety, serviceability, logistics, human factors, reduction of "no faults found," diagnostics capability, and prevention of warranty failures. Very few specifications address even obvious requirements, such as internal interface, external interface, user–hardware interface, user–software interface, and how the product should behave if and when a sneak failure occurs. Developing a good specification is an iterative process with inputs from the customer and the entities that are downstream in the process. Those who are trying to build reliability around a faulty specification should only expect a faulty product. Unfortunately, most companies think of reliability when the design is already approved. At this stage there is no budget and no time for major design changes. The only thing a company can do is to hope for reasonable reliability and commit to do better the next time.

To identify missing functions, a cross-functional team is necessary. At least one member from each disciple should be present, such as manufacturing, field service, and marketing, as well as a customer representative. If the specification contains only 50% of the necessary features, how can one even think of reliability? Reliability is not possible without accurate and comprehensive specifications. Therefore, writing accurate performance specifications is a prerequisite for reliability. Such specifications should aim at zero failures for the modes that result in product recalls, high downtime, and inability to diagnose. My interviews with those attending my reliability courses reveal that the dealers are unable to diagnose about 65% of the problems (no faults found). Obviously, fault isolation requirements in the specifications are necessary to reduce down time.

To ensure the accuracy and completeness of a specification, only those who have knowledge of what makes a good specification should approve it. They must ensure that the specification is clear on what the product should never do, however stupid it may sound. For example: "There shall be no sudden acceleration during landing" for an aircraft. In addition, the marketing and sales experts should participate in writing the specification to make sure that old warranty problems "shall not" be in the new product and that there is enough gain in reliability to give the product a competitive edge.

The "shall not" specification is not limited to failures. That would be too simple. We must be able to see the complexity in this simplicity. This is called *interconnectedness*. We need to know that reliability is intertwined with many elements of life-cycle costs. The costs of downtime, repairs, preventive maintenance, amount of logistics support required, safety, diagnostics, and serviceability are dependent on the level of reliability. In the same spirit, we should also analyze product friendliness and modularity, which are interconnected with reliability. For example, General Motors is designing its hydrogen cars to have a single chassis for all models instead of 80 different chassis as is the case with current production. This action influences reliability in many ways. Similarly, an analysis of downtime should be conducted by service engineering staff to ensure that each fault will be diagnosed in a timely manner, repairs will be quick, and life-cycle costs will be reduced by extending the maintenance cycles or eliminating the need for maintenance altogether. The specification should be critiqued for quick serviceability and ease of access. Until the specification is written thoroughly and approved, no design work should begin. An example of the need to identify missing requirements is that nearly 1000 people around the world lost their lives while the kinks were being removed from the 290-ton McDonnell Douglas DC-10 during the 1970s. Blown-out cargo doors, shredded hydraulic lines, and engines dropped during the flight were just a few of the behemoth's early problems. It is obvious that the company did not have the right system performance specification. We rely on customers to tell us what they want, but they themselves don't know many requirements until there is a breakdown. Customers are not going to tell us that the cargo doors should not blow out during a crowded flight. It is the design team's responsibility to figure out what the customers did not say.

To find the design flaws early, a team has to view the system from various angles. You would not buy a house by just looking at the front view. You want to see it from all sides. Similarly, a product concept has to be viewed from at least the following perspectives:

Functions of the product
Range of applications
Range of environments
Active safety
Duty cycles during life
Reliability
Robustness for user or servicing mistakes
Logistics requirements
Manufacturability requirements
Internal interface requirements
External interface requirements
Installation requirements
Shipping and handling capabilities
Serviceability and diagnostics capabilities
Prognostics health monitoring
Usability on other products
Sustainability

There is a need to explain a sustainable design in the list above. Good product design is about meeting current needs without compromising the needs of future generations, such as by pollution or global warming. Current electronic and computers are not designed for sustainability. They should have been designed for reuse—the ability to recycle is not enough. Not everyone makes an effort to recycle. According to NBC News on October 4, 2007, there are over 3 billion such devices and only 15% are recycled. About 200 million tons, with mercury in the monitors and lead in the solder, wind up in landfills and often in drinking water.

Most designers are likely to miss many of the requirements noted above. This knowledge is not new. It can be included by inviting experts in these areas to brainstorm. There is no mechanism for customers to specify all of these. Suppliers that want to do productive work will teach customers how to develop good requirements as a team member. This makes the customer understand what needs to be in the contract. The point here is that if we have to fix many mistakes later (expensively), we cannot be proud of reliability, as craftsmen once were.

Paradigm 3: Measure Reliability by Life-Cycle Costs

It is wrong to measure reliability in terms of failure rates alone. Such a negative index with unknown impact does not get much attention from management, except when there is a crisis. It is the cost of failures that is important. It should be measured by reduction in life-cycle costs. The fewer the failures, the lower is the life-cycle cost. The costs should be measured over the expected life. They are not just warranty costs; they include the cost of downtime, repairs, logistics, human errors, and product liability. When I was in charge of the reliability of the Baltimore Rapid Transit Train system design, the reliability performance was measured in terms of cost per track mile. Similarly, at Baltimore Gas & Electric, reliability is measured in terms of cost per circuit mile. Smart customers look for only one performance feature: the life-cycle cost per unit of use. Those who approve the specification should concentrate on this measure. Reliability must result in cheaper, faster, and better products.

Paradigm 4: Design for Twice the Life

Why twice the life? The simple answer is that it is the fundamental taught in Engineering 101, which seems to have been forgotten. Remember 100% design margin? Second, it is cheaper than designing for one life if we measure reliability by the life-cycle cost savings. A division of Eaton Corporation requires twice-the-life at 500% return on investment [3]. It actually turns the situation into a positive cash flow, since there is nothing to be monitored if the failures occur beyond the first life. The 50% failure rate is now shifted to the second life, when the product is going to be obsolete. Engineers try to design transmission components without increasing the size or weight, using alternative means such as heat treating in a different way or eliminating joints in the assemblies. Occasionally, they may increase the size by a very minor amount, such as on wires or connectors, to expedite the solution. This is acceptable as long as the return on investment is at least 500%.

Another reason for twice the life is the need to avoid engineering changes, which seems to be obvious. Imagine a bridge designed for 20-ton trucks and a 30-year life. It may have no problems in the beginning. But the bridge degrades over time. After 10 years it may not be strong enough to take even 15 tons, and it is very likely to collapse. If it had been designed for twice the load (for 40 tons) or for a 60-year life, it should not fail at all during 30 years. It should be noted that designing for twice the load also results in twice the life most of the time, but one must still use some engineering judgment. This is similar to a 100% design margin. For the same reason, the electronic components in the aerospace industry are derated 50%. In one assembly the load-bearing capability was more than doubled by using a cheaper round key instead of a rectangular key. The round key has practically no stress concentration points. In another design, twice the life as well as twice the load capability were achieved by molding two parts as a single piece, preventing stresses at the joint. The cost was lower because no assembly was required, there were fewer part numbers in the inventory, no failures, and no downtime for customers.

What if we cannot design for twice the life? There are times when we cannot think of a proper solution for twice the life. Then one can go to other options, such as:

- Providing redundancy on the weakest links, such as bolts, corroded joints, cables, and weak components.
- Designing to fail safely such that no one is injured. For automobiles a safe mode can be that the car can switch to a degraded performance with enough time left to reach home or a repair facility.
- Designing-in early prognostics-type warnings so that the user still has sufficient time to correct the situation—before failure occurs. One of the purposes of prognostics is to predict the remaining life.

Paradigm 5: Safety-Critical Components Should Be Designed for Four Lives

The rule of thumb in aerospace for safety-related components is to design for four times the life. A U.S. Navy policy (NAVAIR) is to design safety-critical components for four times the life and conduct a test for a minimum of twice the life. The expected life should include future increases in load. Many airlines use their aircraft beyond the design life by performing more maintenance. This indirectly exposes many components to work beyond the normal one life. This is the main reason for designing for four times the life, to maintain 100% design margin all the time. Similarly, many consumers drive cars far beyond the expected 10-year life.

We should also design for peak loads, not the usual mean load. When a high-voltage cable used in power lines broke easily, engineers could not duplicate the failure with average loads. When they applied the peak loads, they could.

Designing for four times the life does not mean overdesigning. It is the art of choosing the right concept. If the attention is placed on innovation rather than marginal improvements, engineers can design for multiple lives with little or no investment, as shown earlier by several examples. They must encourage themselves to think differently rather than latching on to outdated traditional methods of increasing the size or weight. Engineers who talk of costs when solving problems usually block out creativity. They draw the boundary around the solution. Their first thought is to increase the size or weight to design for high loads. This is very common defective thinking. This is where the universities need to be more knowledgeable. We need to balance logic with creativity and should still be able to show a high return on investment.

Paradigm 6: Learn to Alter the Paradox of Cost and Performance into a Win–Win Situation

Most engineers are of the opinion that high reliability costs more. World-class organizations embrace the paradox of increasing reliability and lowering costs simultaneously. Trade-off between reliability and cost is not always necessary. Toyota has mastered this paradigm, where high reliability and lower life-cycle costs are a way of life. Toyota has learned over the years that preventing failures is always cheaper than fixing them if the failure prevention process starts early in the design. If we capture the potential failures during the requirements analysis, we can include design for reliability without making wasteful engineering changes later. Similarly, during detailed design reviews, such tools as design failure modes, effects, and criticality analysis (FMECA), process FMECA, and fault tree analysis, if used early, can help us discover

many missing, vague, and incomplete requirements. Engineering changes are the biggest source of waste in organizations, because most of them can be prevented. Here are some examples of achieving high reliability with very little or no investment. Since high reliability reduces life-cycle costs, the insignificant amount of investment does not negatively affect the win–win scenario.

Example 1

A company in Brazil had designed a large warning light bulb on a control console, with a plastic cover to reduce glare. They told me that they tried all kinds of plastics for the cap but that all of them melted after a few months. Someone suggested using a glass cover. We received the usual stupid answer: "Glass will cost three times as much as plastic. The cost of the product will be high." The bad part is that many engineers look only at the cost of the component and completely ignore the cost of losing customers and the warranty costs to the employer. They are unaware that the cost of getting a new customer is at least five times the cost of retaining a current customer. When the team calculated the life-cycle costs of plastic versus the glass cap, the return on investment (ROI) turned out to be 300% in favor of the glass material. The author requested them to put a hold on the solution because we had agreed on an ROI goal of at least 500%. The author advised the entire team to take long showers for three weeks in the hope that someone would come up with a better idea. Why? Because when you take a long shower, your brain is calmed. In this state it is able to use over 1000 billion neurons that you have never used.

It so happened that the present author (the facilitator) was the one taking the long shower. Suddenly I began to feel that the engineers were giving me a snow job! They said that they tried all the plastics and they all melted. This could not be true. There are fundamentally two types of plastics: thermoplastics, which melt with heat, and thermoset plastics, which harden with the heat. I sent them an e-mail suggesting that they try thermoset plastic. It worked. They could not melt it, no matter how much heat they put in. They sent a nice e-mail: "Thanks for the research you did for us." The cost of the new plastic was almost the same. Zero investment. One hundredfold life. One million percent ROI!

Example 2

The original European jet aircraft Comets were cracking around the windows. They were taken out of service for two years. The engineers, as usual, started to design thicker fuselage walls and proposed an enormous cost increase. Then someone suggested examining the failures and discovered that all the failures were around the corners of the widows. He suggested increasing the radius at

the corners. Problem solved quickly, with hardly any investment. The ROI was least 100,000% if you consider the ratio of the cost of thickening the fuselage and the investment in changing the radius on the corners of the windows.

Example 3

At a General Motors facility, the headlamps were failing after about 1000 hours of use. The supplier was going to raise the price 100% to design for twice the life. An engineer turned the filament in the headlamp 90° to avoid harmful vibration and the life increased at least sixfold. Practically zero investment.

Example 4

A dent in a Caterpillar tractor spring was causing premature breakdowns. The reason for the dent was that the spring under the tractor occasionally hit rocks on the ground. The engineers reduced the diameter of the spring such that it wouldn't hit rocks and replaced it with a tougher spring. With a very small investment they got a better than 10,000% ROI.

Paradigm 7: Design to Avoid Latent Manufacturing Flaws

We can design for reliability as much as we want, but if manufacturing processes are subject to operator error and to wide swings in variability, a good design is bound to have premature failures. We need to identify manufacturing features such as the correct torque for fasteners, vulnerability to installing components backward, or vulnerability to using the wrong components. These features could be certain dimensions, alignment, proper fit of mating parts, property of a lubricant, workmanship, and so on. A product should be designed to avoid such vulnerabilities or should be testable during manufacturing to detect abnormalities. For lack of current terminology, we can call it *design to avoid latent manufacturing flaws*.

Let's look at an example of designing to reduce vulnerability to manufacturing variations. A new motorcycle design involved over 50 different fasteners. Following process FMEA, the production operators discovered that a separate torque was required for each fastener joint. They approached design engineers to ask if they could choose about 20 different fasteners instead of 50. This would allows them to concentrate on fewer fasteners and fewer fastening standards. Engineers were flabbergasted: Such advice coming from the hourly workers was an aha! moment for them. They standardized on a few fasteners.

Another example is from Delco Electronics (now Delphi). A plastic panel required that a plating process have a conductive surface. The plating had been

peeling off in two to three years and six sigma team efforts failed to control the plating durability. Someone came up with the bright idea of adding carbon particles to the plastic to make it conductive. The entire plating process was eliminated. The cost went down by 70%. The reliability of the conductivity was now 100%! A good example of over 100,000% ROI.

The secret of controlling manufacturing flaws is to identify where inspection is needed and to design the process such that no inspection is required—if such a solution is possible.

One more example may help. In this case, the process is the focus. Assume that we want to design a dinner table with four legs such that the legs must be equal. If we cut one leg at a time, we cannot get them all equal because of the variability in the cutting process. But if we take all four legs together, and cut all of them with a single cut, they will all be equal.

Paradigm 8: Design for Prognostics Health Monitoring

In complex systems such as telecommunications and fly-by-wire systems, most system failures are not from component failures. They are from very complex interactions and sneak circuits. Failure rates are very difficult to predict. The sudden acceleration experienced by Audi 5000 users during the 1980s was a result of a software sneak failure. A bit in the integrated circuit register got stuck at zero value, which rapidly increased the speed when the gear was engaged in reverse mode. One way to prevent system failures is to monitor the health of critical features such as "stuck at" faults, critical functions, and critical inputs to the system. A possible solution is to develop a software program to determine prognostics, diagnostics, and possible fallback modes.

The following data on a major airline, announced at a Federal Aeronautics Administration (FAA) National Aeronautics and Space Administration (NASA) workshop [4] shows the extent of unpredicted failures:

- Problems reported confidentially by airline employees: about 13,000
- Number actually in airline files: about 2%, or 260
- Number known to the FAA: about 1%, or 130

The sneak failures are more likely to be in embedded software, where it is impractical to do a thorough analysis. Frequently, the software requirements are faulty because they are not derived completely from the system requirements. Peter Neumann, a computer scientist at SRI International, highlights the nature of damage from software defects in the last 15 years [5]:

- Wrecked a European satellite launch
- Delayed the opening of the new Denver airport by one year

- Destroyed a NASA Mars mission
- Induced a U.S. Navy ship to destroy an airliner
- Shut down ambulance systems in London, leading to several deaths

To counter such risks, we need an early warning, early enough to prevent a major mishap. This tool is prognostics health monitoring. It consists of tracking all the possible unusual events, such as signal rates, the quality of the inputs to the system, or unexpected outputs from the system, and designing in intelligence to detect unusual system behavior. The intelligence may consist of measuring important features and making a decision as to their impact. For example, a sensor input occasionally occurs after 30 milliseconds instead of 20 milliseconds as the timing requirement states. The question is: Is this an indication of a disaster? If so, the sensor calibration may be required before the failure manifests as a mishap.

SUMMARY

In summary we can say that we need to define functions correctly. We need to design not to fail, and we need to implement all the paradigms covered in this chapter, including designing to avoid manufacturing problems. Once I was at a company meeting where the customers were asked to describe the warranty they would wish to have. One of them said (and others agreed): No warranty is the best warranty. Very few understood the paradox—the best warranty would be one that would never experience a claim. In other words, the customers wanted a failure-free design for reliability.

REFERENCES

[1] Kuo, W., Compatibility and simplicity: the fundamentals of reliability, *IEEE Trans. Reliab*., vol. 56, Dec. 2007.
[2] Raheja, D. G., *Product Assurance Technologies*, Design for Competitiveness, Inc., 2002.
[3] Raheja, D. G., and Allocco M., *Assurance Technologies Principles and Practices: A Product, Process, and System Safety Perspective*, 2nd ed., Wiley, Hoboken, NJ, 2006, Appendix.
[4] Farrow, D. R., presented at the Fifth International Workshop on Risk Analysis and Performance Measurement in Aviation, sponsored by FAA and NASA, Baltimore, Aug. 19–21, 2003.
[5] Mann, C. C., Why software is so bad, *Technol. Rev*., July–Aug. 2002.