System Safety in Healthcare

by Dev Raheja

Lessons Learned From Designing for Safety

Designing for safety is a process. When the right process is followed, results can be great, such as eliminating most of the warranty costs. 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 these "right things" comes from using "lessons learned." One must use previously accumulated knowledge to arrive at correct decisions. Theory is not enough. We must become 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 2,000 times before the success of the light bulb. His answer? "I never failed. There were 2,000 steps in this process."

As system safety professionals, we can use the following lessons learned.

Spend a Lot of Time on Requirement Analysis

Most failures result from incomplete, ambiguous and poorly defined specifications. This is why we introduce unnecessary design changes and write deviations when we are in a hurry to ship a product.

Be alert, particularly, for missing functions in the specifications. There is often practically nothing in a specification about system safety, modularity, reliability, serviceability, logistics, human factors, 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.

To identify missing functions, a cross-functional team is necessary. At least one member from each discipline should be present, including 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 safety?

To ensure the accuracy and completeness of a specification, only those who have knowledge of the elements of a good specification should approve it. They must ensure that the specification is clear on what the product should never do, however obvious this 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 safety to give the product a competitive edge. To find the design flaws early, a team must 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 must be viewed from at least the following perspectives:

- Functions of the system
- Range of applications
- Range of environments
- Active safety
- Duty cycles during life
- Reliability
- Durability
- 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 systems
- Sustainability

There is a need to explain a sustainable design in this list. Good system design is about meeting current needs without compromising the needs of future

“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 was designed for twice the load (for 40 tons) or for a 60-year life, it should not fail at all during 30 years.”

generations, such as through impacts like pollution or global warming.

Design the System for Twice the Life

Why twice the life? The simple answer is that it's the fundamental taught in Engineering 101, yet often seems to be forgotten. Remember 100% design margin? Second, it is cheaper than designing for one life if we measure safety by the life-cycle cost savings. A division of Eaton Corporation required twice-the-life at 500% return on investment [Ref. 1]. It actually turned the situation into positive cash flow, since there is nothing to be monitored if the failures occur beyond the first life. Engineers tried 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 increased the size by a very minor amount, such as on wires or connectors, to expedite the solution. This was acceptable as long as the return on investment was at least 500%.

Another reason for twice the life is the obvious need to avoid engineering changes. 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 was 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 while designing for twice the load also results in twice the life most of the time, one must still use some engineering judgment. This is also an example of a 100% design margin. For this same reason, the electronic components in the aerospace industry are de-rated 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.

What if we cannot design 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 so that no one is injured. For automobiles, a safe mode could be that the car switches to a degraded performance to allow time 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.

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 to 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 expectation of 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 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 in earlier 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 common defective thinking. This is where the universities need to be more knowledgeable.

Design for Prognostics Health Monitoring

In complex systems such as telecommunications and fly-by-wire systems, most system failures are not from component failures; rather, they are from very complex interactions and sneak circuits. Failure rates are difficult to predict. The sudden acceleration experienced by Audi 5000 users during the 1980s, for example, 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. One 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) and National Aeronautics and Space Administration (NASA) workshop [Ref. 2], 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

Sneak failures are more likely to be in embedded software, where it is impractical to do a thorough analysis.

Frequently, software requirements are faulty because they are not derived completely from the system requirements. Peter Neumann, a computer scientist at SRI International, highlights examples of damage from software defects [Ref. 3]:

- 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. This intelligence may consist of measuring important features and making decisions regarding 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, sensor calibration may be required before the failure manifests as a mishap.

Conclusion

In this article, I've shared a few good lessons. In my book *Design for Reliability*, I share additional ideas. 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 lessons learned, including designing to avoid manufacturing problems.

Once, I was at a company meeting where the customers were asked to describe the warranty they wished 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 safety and reliability.

References

1. Raheja, D. G. and M. Allocco. *Assurance Technologies Principles and Practices: A Product, Process, and System Safety Perspective*, 2nd ed., Wiley, Hoboken, NJ, 2006, Appendix.
2. Farrow, D. R. presented at the Fifth International Workshop on Risk Analysis and Performance Measurement in Aviation, sponsored by FAA and NASA, Baltimore, August 19–21, 2003.
3. Mann, C. C. "Why software is so bad," *Technol. Rev.*, July–August 2002. https://www.technologyreview.com/s/401594/why-software-is-so-bad/