1

Design for Safety

Paradigms

Dev Raheja, Louis J. Gullo, and Jack Dixon

1.1 Why Design for System Safety?

Only through knowledge of a specific system’s performance can a person understand how to design for safety for that particular system. Anyone designing for safety should realize that there is no substitute for first-hand knowledge of a system’s operating characteristics, architecture, and design topology. The most important parts of this knowledge is understanding the system—learning how it performs when functioning as designed, verifying how the system performs when applied under worst-case conditions (including required environmental stress conditions), and experiencing faulty conditions (including mission-critical failures and safety-critical failures).

1.1.1 What Is a System?

A system is defined as a network or group of interdependent components and operational processes that work together to accomplish the objectives and requirements of the system. Safety is a very important aim of a system while executing and accomplishing its objectives and requirements. The design process of any system should ensure that everybody involved in using the system or developing the system gains something they need, avoiding the allure to sacrifice one critical
part of the system design in favor of another critical part of the system. This context includes customers, system operators, maintenance personnel, suppliers, system developers, system safety engineers, the community, and the environment.

1.1.2 What Is System Safety?

System safety is the engineering discipline that drives toward preventing hazards and accidents in complex systems. It is a system-based risk management approach that focuses on the identification of system hazards, analysis of these system hazards, and the application of system design improvements, corrective actions, risk mitigation steps, compensating provisions, and system controls. This system-based risk management approach to safety requires the coordinated and combined applications of system management, systems engineering, and diverse technical skills to hazard identification, hazard analysis, and the elimination or reduction of hazards throughout the system life cycle.

1.1.3 Organizational Perspective

Taking a systems approach enables management to view its organization in terms of many internal and external interrelated organization and company business connections and interactions, as opposed to discrete and independent functional departments or processes managed by various chains of command within an organization. (Note: The term “organization” will be used throughout the book to refer to all system developer and customer entities to include businesses, companies, suppliers, operators, maintainers, and users of systems.) When all the connections and interactions are properly working together to accomplish a shared aim, an organization can achieve tremendous results, from improving the safety of its systems, products, and services to raising the creativity of an organization to increasing its ability to develop innovative solutions to help mankind progress.

1.2 Reflections on the Current State of the Art

System safety is defined as the application of engineering and management principles, criteria, and techniques to achieve acceptable risk within the constraints of operational effectiveness and suitability, time, and cost throughout all phases of the system life cycle [1]. We have come a long way since the early days of system safety in the 1960s. System safety in many organizations has been successfully integrated into the mainstream of systems engineering and is vigorously supported by management as a discipline that adds value to the product development process. Many analysis techniques have been created and revised numerous times to make them more effective and/or efficient. The
application of system safety in product design and development has proven valuable in reducing accidents and product liability.

However, there are still many challenges facing system safety engineers. First and foremost, even after over 50 years, system safety is still a small and somewhat obscure discipline. It needs more visibility. While many organizations successfully implement system safety, many continue to ignore its benefits and suffer the consequences of delivering inferior, unsafe products.

Other challenges include the continually increasing complexity of systems being developed. Now, instead of only worrying about one system at a time, we must worry about building safe systems of systems. This additional complexity has introduced new challenges of how to address the interactions of all the systems that might make up a system-of-systems.

Inadequate specifications and requirements continue to plague the discipline. Too often weak, generic specifications are provided to the designers leading to faulty designs because the requirements were vague or ill defined.

The management of change is often another weakness in the product life cycle. As changes are made to the product or system, system safety must be involved to ensure that the changes themselves are safe and that they do not cause unintended consequences that could lead to accidents.

The human often causes safety problems by the way he uses, or abuses, the product. All too often the user can be confused by the complexity of a product or system or by the user interface provided by the software that operates it. Taking the human into consideration during the design process is paramount to its successful deployment.

The goal of this book is to help remedy some of these problems and build upon the many years of success experienced by system safety. In this chapter we present 10 paradigms we believe will lead to better and safer product designs. Throughout this chapter, and the book, we provide both good and bad examples so the reader can identify with real-world cases from which to learn.

### 1.3 Paradigms for Design for Safety

Forming an ideal system’s approach to designing new systems involves developing paradigms, standards, and design process models for a developer to follow and use as a pattern for themselves in their future design efforts. These paradigms are often called “words of wisdom” or “rules of thumb.” The word “paradigm,” which originated from the Greek language, is used throughout the content of this book to describe a way of thinking, a framework, and a model to use to conduct yourself in your daily lives as a system safety engineer, or any type of engineer. A paradigm becomes the way you view the world, perceiving, understanding,
and interpreting your environment and helping you formulate a response to what you see and understand.

This book starts by focusing on 10 paradigms for managing and designing systems for safety. These 10 paradigms are the most important criteria for designing for safety. Each of these paradigms is listed next and is explained in detail in separate clauses of this chapter following the list:

- Paradigm 1: Always aim for zero accidents.
- Paradigm 2: Be courageous and “Just say no.”
- Paradigm 3: Spend significant effort on systems requirements analysis.
- Paradigm 4: Prevent accidents from single as well as multiple causes.
- Paradigm 5: If the solution costs too much money, develop a cheaper solution.
- Paradigm 6: Design for Prognostics and Health Monitoring (PHM) to minimize the number of surprise disastrous events or preventable mishaps.
- Paradigm 7: Always analyze structure and architecture for safety of complex systems.
- Paradigm 8: Develop a comprehensive safety training program to include handling of systems by operators and maintainers.
- Paradigm 9: Taking no action is usually not an acceptable option.
- Paradigm 10: If you stop using wrong practices, you are likely to discover the right practices.

These paradigms, which are referenced here, are cited throughout the course of this book. Table 1.1, at the back of this chapter, provides a guide to where in this book the various paradigms are addressed.

1.3.1 Always Aim for Zero Accidents

Philip Crosby, the former Senior Vice President (VP) at ITT and author of the famous book *Quality is Free*, pioneered the zero defects standard. Philip Crosby considered “zero defects” as the only standard you needed. This applies even more to safety. It is a practice that aims to prevent defects and errors and to do things right the first time. The ultimate aim is to reduce the level of defects to zero. The overall effect of achieving zero defects is the maximization of profitability.

To experience high profitability, an organization has to compare the life cycle costs of designing the product using current methods versus improving the design for zero accidents using creative solutions. Such creative solutions are usually simple. Because of this, they are often called elegant solutions. It may be a cheaper option to develop an elegant solution over a complex solution. As the great Jack Welch, former CEO of General Electric (GE), said in his book *Get Better or Get Beaten* [2], doing things simply is the most elegant thing one can do. In a company
in Michigan, a shaft/key assembly for a heavy-duty truck transmission was designed for zero failures for at least 20 years by changing the heat-treating method. Heat treatment (e.g., annealing) is a process where heating and cooling of a metallic item alter the material properties for the purpose of improving the design strength and reducing the risk of hazards or failures. In this case, the temperature range and the heating/cooling rates were varied to achieve the optimum design strength. The cost of the new heat treatment method was the same as the previous method. The only additional cost was the cost to run a few experiments to determine the temperature range and the heating/cooling rates that were needed to get the desired strength. The new heat treatment method became a cheaper method when the company eliminated warranty costs, risk of safety-related lawsuit costs, and projected maintenance costs. They received more business as a result of customer satisfaction through achieving high quality and safety. The Return on Investment (ROI) was at least 1000%. This was an elegant solution. This solution was much cheaper than paying legal penalties and maintenance costs for accidents.

Similarly, consider another example involving de Havilland DH 106 Comet aircraft. A redesign of the de Havilland DH 106 Comet aircraft resulted from the first three de Havilland DH 106 Comet aircraft fuselage failures in 1952. The failures were located around the perimeter of the large square windows on the fuselage, which were manufactured without increasing the thickness of the fuselage in this area. The metal fatigue from high stress concentration on the sharp corners was causing the failures. They eliminated the sharp corners by designing oval-shaped windows. This redesign was much cheaper than paying for the accidents that would have resulted if the design was not changed [3].

1.3.2 Be Courageous and “Just Say No”

Paradigm 2 is to be courageous and “Just say no” to those who want to rush designs through the design review process without exercising due diligence and without taking steps to prevent catastrophic events. Say “No” at certain times during the system development design process to prevent future possible catastrophic events as they are discovered. Many organizations have a Final Design Review. Some call it the Critical Design Review. This is the last chance to speak up if anyone is concerned about anything in the design. A very important heuristic to remember is “Be courageous and just say no.” The context here is that if the final design is presented with known safety design issues, and everyone votes “yes” to the design approval without seriously challenging it, then your answer should be “no.” Why? Because there are almost always new problems lingering in the minds of the team members, but they don’t speak up at the appropriate time. They are probably thinking that it is too late to interfere or they want
to be a part of the groupthink process where everyone thinks alike. No matter how good the design is, an independent facilitator can find many issues with it. Ford Motor Company hired a new VP during the design of the 1995 model of the Lincoln Continental car. The company had been making this car for years, and everyone on the team had at least 10 years of experience. The design was already approved, but the new VP insisted on questioning every detail of the design with a cross-functional team composed of engineers from each subsystem.

Though its redesign began four months later than had been intended, the 1995 Lincoln Continental was available on the market one month ahead of schedule. The team made over 700 design changes. Since they made these improvements while the design was still on paper, the team completed their project using only a third of their budgeted 90 million dollars, resulting in savings of 60 million dollars [4].

It takes courage to be a real change agent and a true believer that a change has critical importance. Jack Welch stated in his book *Winning* [5], that real change agents comprise less than 10% of all business people. They have courage—a certain fearlessness about the unknown. As Jack says, “Change agents usually make themselves known. They’re typically brash, high energy, and more than a little bit paranoid about the future. Very often, they invent change initiatives on their own or ask to lead them. Invariably, they are curious and forward looking. They ask a lot of questions that start with the phrase: “Why don’t we....?”’

To recommend design for safety changes at critical points in the system design cycle, and be successful in implementing the design changes, you will need to win people to your way of thinking. As Dale Carnegie stated in his book *How to Win Friends and Influence People* [6], you need to exercise 12 principles to win people to your way of thinking. Paraphrasing these principles here, we have:

Principle 1: Get the best of an argument by avoiding it.
Principle 2: Respect the other person’s opinions and avoid saying, “You are wrong.”
Principle 3: If you realize that you are wrong, admit you are wrong immediately.
Principle 4: Begin a discussion in a friendly way and nonconfrontational.
Principle 5: Provoke “yes, yes” responses from the other person immediately.
Principle 6: Allow the other person to do the most talking.
Principle 7: Let the other person think your idea is also their idea.
Principle 8: See things from the other person’s point of view and perspective.
Principle 9: Consider and sympathize with the other person’s ideas and desires.
Principle 10: Appeal to nobler motives.
Principle 11: Dramatize your ideas to display your passion.
Principle 12: Throw down a challenge when nothing else works.
1.3.3 Spend Significant Effort on Systems Requirements Analysis

Most accidents or system failures originate from bad requirements in specifications. The sources of most requirement failures are incomplete, ambiguous, and poorly defined device specifications. They result in making expensive engineering changes later, one at a time called “scope creep.” Often robust changes cannot be made because the project is already delayed and there may not be resources for implementing new features.

Look particularly hard for missing functions in the specifications. Usually there are insufficient requirements for modularity (to minimize interactions), reliability, safety, serviceability, logistics, human factors, testability, diagnostics capability, and the prevention of old failures in current designs. Specifications need to address interoperability functions in more detail such as for the requirements for internal interfaces, external interfaces, user–hardware interfaces, and user–software interfaces. Specifications also need to address how the product should behave if and when there is an unexpected behavior such as the device shutdown from a power failure or from an unexpected human error. Those who are trying to design around a faulty specification should only expect a faulty design. Unfortunately, most companies still discover design problems when the design is already in production. At this stage there are usually no resources and no time available for major design changes.

To identify missing functions in a specification, performing requirements analysis is necessary. This analysis is always conducted by a cross-functional team. At least one member from each discipline should be present, such as R&D, design, quality, reliability, safety, manufacturing, field service, marketing, and if possible, a customer representative. New products have missing and vague requirements as much as 60% [7]. Therefore, writing accurate and comprehensive performance specifications is the prerequisite for a safe design. Personal experiences during interviews with various people attending safety training reveal that the troubleshooting technicians on complex electronic products are unable to diagnose about 65% of the problems (i.e., cannot duplicate the fault). Obviously, improved fault isolation requirements in the specifications are necessary for such products, but the cost may be too prohibitive. In large complex systems, it is incredibly expensive to obtain 100% test coverage and harder still to obtain 100% fault coverage isolating to a single item that caused a failure.

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

- Functions of the product
- What undesired functions the product should never do (such as sudden acceleration)
- Range of applications
• Range of environments
• Active safety (inherent safety controls during the use of the device)
• Duty cycles during life
• Reliability for lifetime
• Robustness for user/servicing mistakes
• Logistics requirements to avoid adverse events
• Manufacturability requirements for defect-free production
• Internal interface requirements
• External interface requirements
• Installation requirements to assure safe functioning (an MRI was found inaccurate after the installation)
• Shipping/handling capabilities to keep the device safe (electronic components fail in humid environments)
• Serviceability/diagnostics capabilities
• PHM to warn users in case of an anomaly
• Interoperability with other products
• Sustainability
• Potential accidents and abuses
• Human factors

Most designers are likely to miss some of the aforementioned requirements. Almost all of them affect safety. This knowledge is not new; it can be incorporated by inviting experts in these areas to brainstorm. Requirements should include how the system should behave when a sneak failure occurs. A tool called Sneak Circuit Analysis (SCA) can be used for predicting sneak failures. A good specification will also address what the system shall not do, such as “there shall be no sudden fires in automobiles caused by any reason.” The point here is, if we do everything right the first time, we avoid pain and suffering to users.

1.3.4 Prevent Accidents from Single as well as Multiple Causes

Many Failure Modes and Effects Analysis (FMEA) assess designs to determine the possibility of accidents that can happen from a single point failure cause, such as a component failure or a human error, or from multiple failure causes. According to the system safety theory [8], at least two events happen in any accident. There has to be a latent hazard in the system and a trigger event, such as human error, to activate an accident. A latent hazard can be an oversight in design, poor supervision, poor specification, inadequate risk analysis, or inadequate procedure. The Fault Tree Analysis (FTA) tool can reveal the combinations of causes for the hazard and the trigger event. To design out the accident, you could either prevent the latent hazard or prevent the trigger event.
An example concerning the Space Shuttle Challenger accident makes this clear. The NASA Challenger carrying eight astronauts exploded shortly after lift-off in 1986, causing the deaths of all personnel on board. The rubber o-ring seal that was not capable of functioning properly below 40°F caused the hazard. The trigger event was that management decided to fly at a lower temperature (human error) when the specification was clear about not launching below 40°. The whole accident could have been prevented if either the hazard (seal weakness) was designed out or the decision-making process by management was able to prevent errors in decisions to launch. Since human errors are very hard to control, the best strategy would be to design the seal that is good at any temperature feasible for a launch. This is the change NASA made after the accident. They added a heating wire around the seal so that the seal would be flexible enough to provide a good seal at any temperature.

The most difficult thing is to know the latent hazards prior to design release and to prevent hazards by design. We can predict many hazards if we use the tools such as Preliminary Hazard Analysis (PHA), FTA, SCA, Operating & Support Hazard Analysis (OS&HA), and FMEA.

1.3.5 If the Solution Costs Too Much Money, Develop a Cheaper Solution

The previous example of the Challenger shows that at very little cost, the seal would have been designed to fly at any temperature. The cost of heating the wire was probably less than $1000, while the cost of the accident was in the millions of dollars and cost eight lives. The lesson to be learned is that a cross-functional team should not accept any design without challenging the design. Apparently this design (not to fly below 40°) was not challenged enough. If someone would have challenged the seal as a hazard, they had a chance of preventing the accident at very low cost. Another example of an inexpensive change was covered previously in Paradigm 1 where just changing the heat treatment method resulted in at least 20 years life for a heavy-duty truck transmission.

The solutions are simple and not costly if a cross-functional team engages in creative brainstorming and comes up with at least 10 possible solutions for every problem. One of them is likely to be very cost effective. In the previous example where just changing the heat-treating method resulted in failure-free long life, this approach was used. This organization required 500% ROI even on safety to encourage very robust and cost-effective designs. It required system hazard analysis before approving the specifications and required design mitigation for all catastrophic hazards. The trick is to make all the big safety changes early during the concept stage where the cost of change is insignificant. The precedence for mitigating risk during the concept stage is as follows:

- Change the requirements to avoid the hazard.
- Introduce fault tolerance.
• Design to complete the mission safely.
• Provide early prognostic warnings.

Note that inspecting and testing are not included in the previous list. One cannot depend on inspection and testing for safety. However, it is always wise to inspect or test just to be confident in the solutions and to watch out for new defects inadvertently introduced during engineering changes.

These strategies almost always result in drastic reductions in life cycle costs, such as reduction in warranty costs, reduction in fatalities, reduction in maintenance/repair costs, reduction in accidents costs, and reductions in environmental damage costs and several other costs. Safety must always make a good business case if a best solution is chosen from creative brainstorming.

1.3.6 Design for Prognostics and Health Monitoring (PHM) to Minimize the Number of Surprise Disastrous Events or Preventable Mishaps

PHM technology and application enhances system safety, efficiency, availability, and effectiveness. In complex systems such as telecommunications and aerospace systems, most of the system failures are caused by fundamental limitations in the design strength and mechanical degradation mechanisms that propagate with time and lead to physical wear-out effects. Design strength must be able to withstand worst-case application and environmental stress conditions. There is a need for innovative solutions for discovering hidden problems, which usually turn up in rare events as probabilistic nondeterministic faults. Through the use of embedded sensors for health monitoring and predictive analytics within embedded processors, prognostic solutions for predictive maintenance are a possibility. PHM is an enabler for system reliability and safety. We need innovative tools for discovering hidden problems, which usually turn up in rare events, such as an airbag that does not deploy when needed in a crash. In the case of an airbag design, some brainstorming needs to be done on questions such as “Will the air bag open when it is supposed to?” “Will it open at the wrong time?” “Will the system give a false warning?” or “Will the system behave fail-safe in the event of an unknown component fault?”

The bottom line is that no matter how much analysis we do, it is impossible to analyze millions of combinations of events or faults that might occur. These faults might be due to incomplete test coverage or fault coverage because of limited time and funds to provide 100% fault detection capability through Built-In-Test (BIT) or external support test equipment. There will always be a certain percentage of faults that are unknown unknowns. There are an unknown number of failures that will remain hidden, latent, and unknown until they occur and are
detected, most likely during a mission when a function that is critical to a mission is disabled. These are called unpredicted failures. The following data on a major airline, announced at a FAA/NASA workshop [9], shows the extent of unpredicted failures:

- Approximately 130 problems known to FAA
- Approximately 260 actual problems in airline files
- Approximately 13,000 problems reported confidentially by the employees of the airline

The sneak failures are more likely to be in the embedded software where it is impractical to do a thorough analysis. Frequently the specifications are faulty because they are not derived from the system performance specification or they are not based on the system-of-systems. Peter Neumann, a computer scientist at SRI International, highlights the nature of damage from software defects [10]:

- Wrecked a European satellite launch
- Delayed the opening of the new Denver airport by one year
- Destroyed NASA Mars mission
- Induced a US Navy ship to destroy an airliner
- Shut down ambulance systems in London leading to several deaths

To counter such risks, we need an early prognostic warning, with health monitoring through embedded sensors to prevent a major mishap. The design process to incorporate these early warnings consists of postulating all the possible mishaps and designing intelligence to detect unusual behavior of the system. The intelligence may consist of measuring important features and making a decision on their impact. For example, a sensor input occasionally occurs after 30ms instead of 20ms as the timing requirement states. The question is: “Is this an indication of a disaster?” If so, the sensor should be replaced before the failure manifests itself to a critical state.

1.3.7 Always Analyze Structure and Architecture for Safety of Complex Systems

The systems today are connected directly and indirectly to many other systems. In large systems such as weapon systems, safety is linked to other systems, vehicles, soldiers, and satellites. There are almost an unlimited number of interactions possible. Tweaking in one place is bound to create some change in another place. As a result, all latent hazards are impossible to predict with high confidence. Therefore the organizations following a good process and doing the right things
need to rethink how to deal with enormous complexity. It is hard enough to do the right things, but it is even harder to know what the right things are!

So, what can we do to control complexity? An organization must analyze not just the safety of its own system but also the safety of the system-of-systems made up of interconnecting systems. Safety needs to pay a lot more attention to hazard analysis on the structure and architecture of the complex system. The architecture must have modularity and traceability to safety-related interactions. Unfortunately, by the time safety engineers get involved, the structure and architecture are usually already chosen. Therefore, early involvement by safety engineering is critically important when dealing with large, complex systems.

1.3.8 Develop a Comprehensive Safety Training Program to Include Handling of Systems by Operators and Maintainers

Development of a complete safety training program for certifying the operators and maintainers requires not only recognizing the components and subsystems but also understanding the total system. Many safety training programs are focused only on the subsystem training. When this occurs, it means that the certification of the person operating or maintaining the equipment is limited. The operating and maintenance personnel may therefore not realize that the total system can be affected by hidden hazards. For example, having only one power source may negatively affect a maintainer’s work. For safety-critical work, a redundant source would be a good mitigation to implement. However, if all sources of power are lost (prime, secondary, and emergency), the total system will not work until a correction is made. Safety training programs must include the process of making safe corrections. It should include total inter-connected system training addressing the worst potential secondary effect of hazards. Scenarios must be developed that provide instructors and students with a realistic understanding of the whole system and how to protect people from harm. Safety training represents a major mitigation method with complex systems; therefore it is imperative that complete training be provided for correctly operating and maintaining the system at all times and for all likely scenarios.

1.3.9 Taking No Action Is Usually Not an Acceptable Option

Sometimes, the teams cannot come up with a viable solution. They take no action or postpone the action hoping that a problem will be solved over time. This may be done out of denial, or because of fear to take action, or the fear of upsetting the superiors in management. Meanwhile, a product with a potential for fault finds its way to the market. Whatever be the case, all stakeholders including the
customer become victims. The goal is to prevent the customers from becoming victims and casualties from poor design practices. Paradigm 2 earlier in this chapter provides helpful guidance on how to influence change. Doing nothing about a known problem is unacceptable.

1.3.10 If You Stop Using Wrong Practices, You Are Likely to Discover the Right Practices

The cause of many product recalls is insufficient knowledge of what needs to be done before the product development begins. Some wrong things discussed in this chapter are worth repeating:

- The design team starts designing the product without a thorough requirements analysis. They accept requirements without much challenging. They pay very little attention to missing and vague requirements.
- The project is approved without proper selection of the right team. A good design team must, as a minimum, include a customer representative from marketing or field service, a person with a thorough knowledge of the science of designing for safety, a person from manufacturing to assure the ability to produce the defect-free product, and a reliability engineer to guide the team on failure-free performance over the expected product life.
- The design team trusts 100% testing and 100% inspection in production. They do not understand that 100% inspection is less than 80% effective over time according to the quality gurus such as Deming and Crosby. Testing rarely represents the actual use environment.
- The design team frequently make costly design changes instead of following a structured approach for robust designs that lower life cycle costs.

Encourage everyone on a design team to identify wrong things. Ask every team member to identify at least five wrong things. They are always able to do so. With a good leader, this can be done with each team member. Then, create a plan to stop working on the wrong things, and replace them with the right things. In one case, employees came up with 22 solutions for a single design problem. About five of them had more than a 600% ROI. Almost always, the right things seem to just appear by themselves. You just have to look for them.

1.4 Create Your Own Paradigms

The advantage of creative brainstorming for potential mishaps and potential solutions is that teams will have occasional “aha” moments. Someone will have insight into what should have been done instead of what was done. This situation
relates to a personal experience related to a system requirements analysis. In one particular example, at least 100 missing requirements were identified in the system specification during a brainstorming session on a complex product. At that time, a particular quote came to mind. This quote was “If you don’t know where you are going, you’ll probably end up somewhere else.” This can be a nice paradigm for R&D engineers and specification writers. Such ideas can come while reading a good book also. Dr. Deming, the father of the Japanese quality system, used to say, “Working hard won’t help if you are working on wrong things.” This is a very useful paradigm. This was realized in another example in which a company had 400-page document on FMEA, but they did not make a single design change. That is nonproductive! They relied on inspection and testing. They worked hard on this FMEA in one meeting per week for six months. It surely did not help the company. One can create a new paradigm from this incidence: Design out the problems, it is cheaper than inspecting and testing in production.

1.5 Summary

In conclusion, these paradigms help you in doing the right things at the right times. If we prevent hazards and failures by doing the right things right the first time, there is no such thing as the cost of safety since the cost is part of the initial design and cannot be separated. These hazard and failure prevention actions include doing the right safety analyses, using the 10 paradigms, and incorporating robust design risk mitigations early. The ROI is usually at least 1000-fold comparing the cost of doing nothing against the cost of the design for safety analysis and resultant design change preventive actions. With ROI this high, there is no reason a safety-critical design change to preventive hazards should not be accomplished. Table 1.1 provides a guide to where in this book the various paradigms are addressed.

References


<table>
<thead>
<tr>
<th>Paradigm number</th>
<th>Paradigm</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Always aim for zero accidents</td>
<td>X  X  X</td>
</tr>
<tr>
<td>2</td>
<td>Be courageous and “just say no”</td>
<td>X  X</td>
</tr>
<tr>
<td>3</td>
<td>Spend significant effort on systems requirements analysis</td>
<td>X  X</td>
</tr>
<tr>
<td>4</td>
<td>Prevent accidents from single as well as multiple Causes</td>
<td>X  X</td>
</tr>
<tr>
<td>5</td>
<td>If the solution costs too much money, develop a cheaper solution</td>
<td>X  X</td>
</tr>
<tr>
<td>6</td>
<td>Design for Prognostics and Health Monitoring (PHM) to minimize the number of surprise disastrous events or preventable mishaps</td>
<td>X  X</td>
</tr>
<tr>
<td>7</td>
<td>Always analyze structure and architecture for safety of complex systems</td>
<td>X  X</td>
</tr>
<tr>
<td>8</td>
<td>Develop a comprehensive safety training program to include handling of systems by operators and maintainers</td>
<td>X  X</td>
</tr>
<tr>
<td>9</td>
<td>Taking no action is usually not an acceptable option</td>
<td>X  X</td>
</tr>
<tr>
<td>10</td>
<td>If you stop using wrong practices, you are likely to discover the right practices</td>
<td>X  X</td>
</tr>
</tbody>
</table>

**Table 1.1** Paradigm locations