6 Engineering and Operating Safer Systems Using STAMP

Part III of this book is for those who want to build safer systems without incurring enormous and perhaps impractical financial, time, and performance costs. The belief that building and operating safer systems requires such penalties is widespread and arises from the way safety engineering is usually done today. It need not be the case. The use of top-down system safety engineering and safety-guided design based on STAMP can not only enhance the safety of these systems but also potentially reduce the costs associated with engineering for safety. This chapter provides an overview, while the chapters following it provide details about how to implement this cost-effective safety process.

6.1 Why Are Safety Efforts Sometimes Not Cost-Effective?

While there are certainly some very effective safety engineering programs, too many expend a large amount of resources with little return on the investment in terms of improved safety. To fix a problem, we first need to understand it. Why are safety efforts sometimes not cost-effective? There are five general answers to this question:

1. Safety efforts may be superficial, isolated, or misdirected.
2. Safety activities often start too late.
3. The techniques used are not appropriate for the systems we are building today and for new technology.
4. Efforts may be narrowly focused on the technical components.
5. Systems are usually assumed to be static throughout their lifetime.

Superficial, isolated, or misdirected safety engineering activities: Often, safety engineering consists of performing a lot of very costly and tedious activities of limited usefulness in improving safety in the final system design. Childs calls this "cosmetic system safety" [37]. Detailed hazard logs are created and analyses

performed, but these have limited impact on the actual system design. Numbers are associated with unquantifiable properties. These numbers always seem to support whatever numerical requirement is the goal, and all involved feel as if they have done their jobs. The safety analyses provide the answer the customer or designer wants—that the system is safe—and everyone is happy. Haddon-Cave, in the 2009 Nimrod MR2 accident report, called such efforts *compliance only exercises* [78]. The results impact certification of the system or acceptance by management, but despite all the activity and large amounts of money spent, the safety of the system has been unaffected.

A variant of this problem is that safety activities may be isolated from the engineers and developers building the system. Too often, safety professionals are separated from engineering design and placed within a mission assurance organization. Safety cannot be assured without its already being part of the design; systems must be constructed to be safe from the beginning. Separating safety engineering from design engineering is almost guaranteed to make the effort and resources expended a poor investment. Safety engineering is effective when it participates in and provides input to the design process, not when it focuses on making arguments about the artifacts created after the major safety-related decisions have been made.

Sometimes the major focus of the safety engineering efforts is on creating a *safety case* that proves the completed design is safe, often by showing that a particular process was followed during development. Simply following a process does not mean that the process was effective, which is the basic limitation of many process assurance activities. In other cases the arguments go beyond the process, but they start from the assumption that the system is safe and then focus on showing the conclusion is true. Most of the effort is spent in seeking evidence that shows the system is safe while not looking for evidence that the system is *not* safe. The basic mindset is wrong, so the conclusions are biased.

One of the reasons System Safety has been so successful is that it takes the opposite approach: an attempt is made to show that the system is *unsafe* and to identify hazardous scenarios. By using this alternative perspective, paths to hazards are often identified that were missed by the engineers, who tend to focus on what they want to happen, not what they do *not* want to happen.

If safety-guided design, as defined in part III of this book, is used, the "safety case" is created along with the design. Developing the certification argument becomes trivial and consists primarily of simply gathering the documentation that has been created during the development process.

Safety efforts start too late: Unlike the examples of ineffective safety activities above, the safety efforts may involve potentially useful activities, but they may start too late. Frola and Miller claim that 70–80 percent of the most critical decisions

related to the safety of the completed system are made during early concept development [70]. Unless the safety engineering effort impacts these decisions, it is unlikely to have much effect on safety. Too often, safety engineers are busy doing safety analyses, while the system engineers are in parallel making critical decisions about system design and concepts of operation that are not based on that hazard analysis. By the time the system engineers get the information generated by the safety engineers, it is too late to have a significant impact on design decisions.

Of course, engineers normally do try to consider safety early, but the information commonly available is only whether a particular function is safety-critical or not. They are told that the function they are designing can contribute to an accident, with perhaps some letter or numerical "score" of how critical it is, but not much else. Armed only with this very limited information, they have no choice but to focus safety design efforts on increasing the component's reliability by adding redundancy or safety margins. These features are often added without careful analysis of whether they are needed or will be effective for the specific hazards related to that system function. The design then becomes expensive to build and maintain without necessarily having the maximum possible (or sometimes any) impact on eliminating or reducing hazards. As argued earlier, redundancy and overdesign, such as building in safety margins, are effective primarily for purely electromechanical components and component failure accidents. They do not apply to software and miss component interaction accidents entirely. In some cases, such design techniques can even *contribute* to component interaction accidents when they add to the complexity of the design.

Most of our current safety engineering techniques start from detailed designs. So even if they are conscientiously applied, they are useful only in evaluating the safety of a completed design, not in guiding the decisions made early in the design creation process. One of the results of evaluating designs after they are created is that engineers are confronted with important safety concerns only after it is too late or too expensive to make significant changes. If and when the system and component design engineers get the results of the safety activities, often in the form of a critique of the design late in the development process, the safety concerns are frequently ignored or argued away because changing the design at that time is too costly. Design reviews then turn into contentious exercises where one side argues that the system has serious safety limitations while the other side argues that those limitations do not exist, they are not serious, or the safety analysis is wrong.

The problem is not a lack of concern by designers; it's simply that safety concerns about their design are raised at a time when major design changes are not possible—the design engineers have no other option than to defend the design they have. If they lose that argument, then they must try to patch the current design; starting over with a safer design is, in almost all cases, impractical. If the designers had the

information necessary to factor safety into their early decision making, then the process of creating safer designs need cost no more and, in fact, will cost less due to two factors: (1) reduced rework after the decisions made are found to be flawed or to provide inadequate safety and (2) less unnecessary overdesign and unneeded protection.

The key to having a cost-effective safety effort is to embed it into a system engineering process starting from early concept development and then to design safety into the system as the design decisions are made. Costs are much less when safety is built into the system design from the beginning rather than added on or retrofitted later.

The techniques used are not appropriate for today's systems and new technology: The assumptions of the major safety engineering techniques currently used, almost all of which stem from decades past, do not match the assumptions underlying the technology and complexity of the systems being built today or the new emerging causes of accidents: They do not apply to human or software errors or flawed management decision making, and they certainly do not apply to weaknesses in the organizational structure or social infrastructure systems. These contributors to accidents do not "fail" in the same way assumed by the current safety analysis tools.

But with no other tools to use, safety engineers attempt to force square pegs into round holes, hoping this will be sufficient. As a result, nothing much is accomplished beyond expending time, money, and other resources. It's time we face up to the fact that new safety engineering techniques are needed to handle those aspects of systems that go beyond the analog hardware components and the relatively simple designs of the past for which the current techniques were invented. Chapter 8 describes a new hazard analysis technique based on STAMP, called STPA, but others are possible. The important thing is to confront these problems head on and not ignore them and waste our time misapplying or futilely trying to extend techniques that do not apply to today's systems.

The safety efforts are focused on the technical components of the system: Many safety engineering (and system engineering, for that matter) efforts focus on the technical system details. Little effort is made to consider the social, organizational, and human components of the system in the design process. Assumptions are made that operators will be trained to do the right things and that they will adapt to whatever design they are given. Sophisticated human factors and system analysis input is lacking, and when accidents inevitably result, they are blamed on the operators for not behaving the way the designers thought they would. To give just one example (although most accident reports contain such examples), one of the four causes, all of which cited pilot error, identified in the loss of the American Airlines B757 near Cali, Colombia (see chapter 2), was "Failure of the flight crew to revert

to basic radio navigation when the FMS-assisted navigation became confusing and demanded an excessive workload in a critical phase of the flight." A more useful alternative statement of the cause might have been "An FMS system that confused the operators and demanded an excessive workload in a critical phase of flight."

Virtually all systems contain humans, but engineers are often not taught much about human factors and draw convenient boundaries around the technical components, focusing their attention inside these artificial boundaries. Human factors experts have complained about the resulting *technology-centered automation* [208], where the designers focus on technical issues and not on supporting operator tasks. The result is what has been called "clumsy" automation that increases the chance of human error [183, 22, 208]. One of the new assumptions for safety in chapter 2 is that operator "error" is a product of the environment in which it occurs.

A variant of the problem is common in systems using information technology. Many medical information systems, for example, have not been as successful as they might have been in increasing safety and have even led to new types of hazards and losses [104, 140]. Often, little effort is invested during development in considering the usability of the system by medical professionals or of the impact, not always positive, that the information system design will have on workflow and on the practice of medicine.

Automation is commonly assumed to be safer than manual systems because the hazards associated with the manual systems are eliminated. Inadequate consideration is given to whether new, and maybe even worse, hazards are introduced by the automated system and how to prevent or minimize these new hazards. The aviation industry has, for the most part, learned this lesson for cockpit and flight control design, where eliminating errors of commission simply created new errors of omission [181, 182] (see chapter 9), but most other industries are far behind in this respect.

Like other safety-related system properties that are ignored until too late, operators and human-factors experts often are not brought into the early design process or they work in isolation from the designers until changes are extremely expensive to make. Sometimes, human factors design is not considered until after an accident, and occasionally not even then, almost guaranteeing that more accidents will occur.

To provide cost-effective safety engineering, the system and safety analysis and design process needs to consider the humans in systems—including those that are not directly controlling the physical processes—not separately or after the fact but starting at concept development and continuing throughout the life cycle of the system.

Systems are assumed to be static throughout their lifetimes: It is rare for engineers to consider how the system will evolve and change over time. While designing

for maintainability may be considered, unintended changes are often ignored. Change is a constant for all systems: physical equipment ages and degrades over its lifetime and may not be maintained properly; human behavior and priorities usually change over time; organizations change and evolve, which means the safety control structure itself will evolve. Change may also occur in the physical and social environment within which the system operates and with which it interacts. To be effective, controls need to be designed that will reduce the risk associated with all these types of changes. Not only are accidents expensive, but once again planning for system change can reduce the costs associated with the change itself. In addition, much of the effort in operations needs to be focused on managing and reacting to change.

6.2 The Role of System Engineering in Safety

As the systems we build and operate increase in size and complexity, the use of sophisticated system engineering approaches becomes more critical. Important system-level (emergent) properties, such as safety, must be built into the design of these systems; they cannot be effectively added on or simply measured afterward.

While system engineering was developed originally for technical systems, the approach is just as important and applicable to social systems or the social components of systems that are usually not thought of as "engineered." All systems are engineered in the sense that they are designed to achieve specific goals, namely to satisfy requirements and constraints. So ensuring hospital safety or pharmaceutical safety, for example, while not normally thought of as engineering problems, falls within the broad definition of engineering. The goal of the system engineering process is to create a system that satisfies the mission while maintaining the constraints on how the mission is achieved.

Engineering is a way of organizing that design process to achieve the most cost-effective results. Social systems may not have been "designed" in the sense of a purposeful design process but may have evolved over time. Any effort to change such systems in order to improve them, however, can be thought of as a redesign or reengineering process and can again benefit from a system engineering approach. When using STAMP as the underlying causality model, engineering or reengineering safer systems means designing (or redesigning) the safety-control structure and the controls designed into it to ensure the system operates safely, that is, without unacceptable losses. What is being controlled—chemical manufacturing processes, spacecraft or aircraft, public health, safety of the food supply, corporate fraud, risks in the financial system—is irrelevant in terms of the general process, although significant differences will exist in the types of controls applicable and the design

of those controls. The process, however, is very similar to a regular system engineering process.

The problem is that most engineering and even many system engineering techniques were developed under conditions and assumptions that do not hold for complex social systems, as discussed in part I. But STAMP and new system-theoretic approaches to safety can point the way forward for both complex technical *and* social processes. The general engineering and reengineering process described in part III applies to all systems.

6.3 A System Safety Engineering Process

In STAMP, accidents and losses result from not enforcing safety constraints on behavior. Not only must the original system design incorporate appropriate constraints to ensure safe operations, but the safety constraints must continue to be enforced as changes and adaptations to the system design occur over time. This goal forms the basis for safe management, development, and operations.

There is no agreed upon best system engineering process and probably cannot be one—the process needs to match the specific problem and environment in which it is being used. What is described in part III of this book is how to integrate system safety into any reasonable system engineering process. Figure 6.1 shows the three major components of a cost-effective system safety process: management, development, and operations.

6.3.1 Management

Safety starts with management leadership and commitment. Without these, the efforts of others in the organization are almost doomed to failure. Leadership creates culture, which drives behavior.

Besides setting the culture through their own behavior, managers need to establish the organizational safety policy and create a safety control structure with appropriate responsibilities, accountability and authority, safety controls, and feedback channels. Management must also establish a safety management plan and ensure that a safety information system and continual learning and improvement processes are in place and effective.

Chapter 13 discusses management's role and responsibilities in safety.

6.3.2 Engineering Development

The key to having a cost-effective safety effort is to embed it into a system engineering process from the very beginning and to design safety into the system as the design decisions are made. All viewpoints and system components must be included

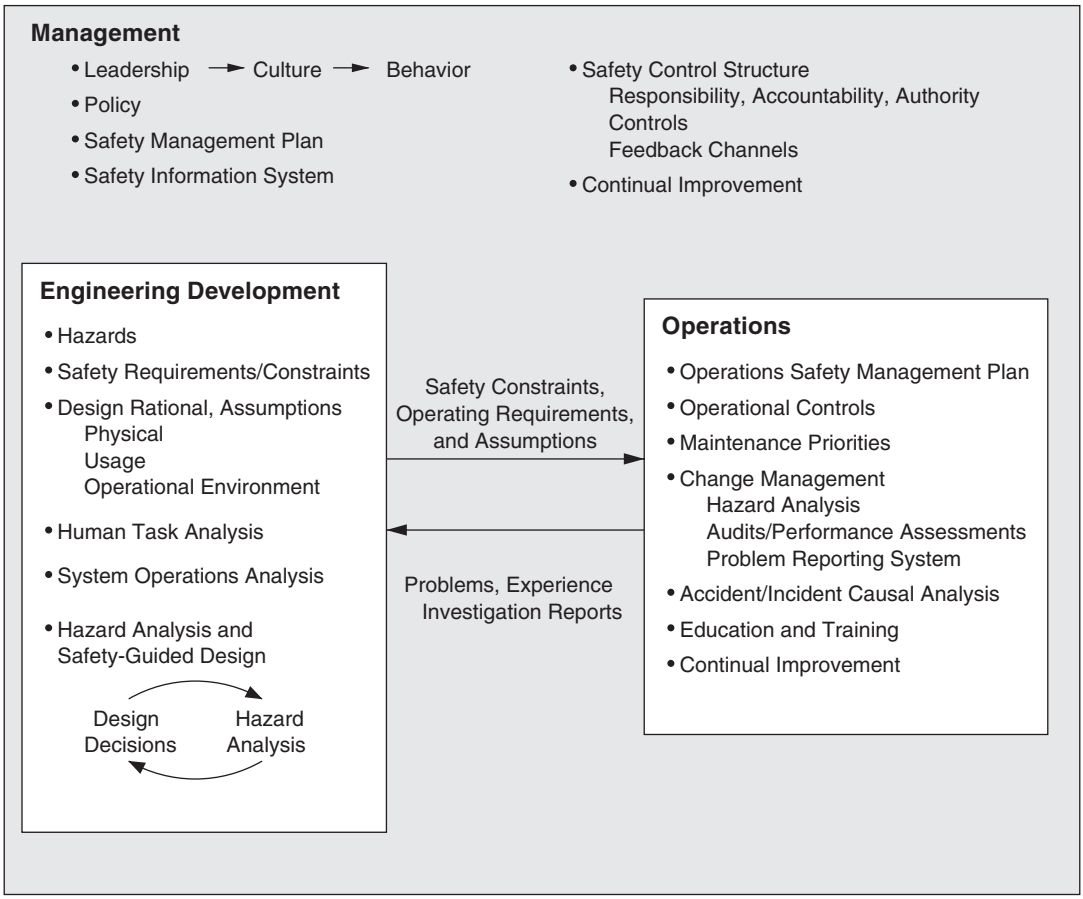

Figure 6.1
The components of a system safety engineering process based on STAMP.

in the process and information used and documented in a way that is accessible, understandable, and helpful.

System engineering starts with first determining the goals of the system. Potential hazards to be avoided are then identified. From the goals and system hazards, a set of system functional and safety requirements and constraints are identified that set the foundation for design, operations, and management. Chapter 7 describes how to establish these fundamentals.

To start safety engineering early enough to be cost-effective, safety must be considered from the early concept formation stages of development and continue throughout the life cycle of the system. Design decisions should be guided by safety

considerations while at the same time taking other system requirements and constraints into account and resolving conflicts. The hazard analysis techniques used must not require a completed design and must include all the factors involved in accidents. Chapter 8 describes a new hazard analysis technique, based on the STAMP model of causation, that provides the information necessary to design safety into the system, and chapter 9 shows how to use it in a safety-guided design process. Chapter 9 also presents general principles for safe design including how to design systems and system components used by humans that do not contribute to human error.

Documentation is critical not only for communication in the design and development process but also because of inevitable changes over time. That documentation must include the rationale for the design decisions and traceability from high-level requirements and constraints down to detailed design features. After the original system development is finished, the information necessary to operate and maintain it safely must be passed in a usable form to operators and maintainers. Chapter 10 describes how to integrate safety considerations into specifications and the general system engineering process.

Engineers have often concentrated more on the technological aspects of system development while assuming that humans in the system will either adapt to whatever is given to them or will be trained to do the "right thing." When an accident occurs, it is blamed on the operator. This approach to safety, as argued above, is one of the reasons safety engineering is not as effective as it could be. The system design process needs to start by considering the human controller and continuing that perspective throughout development. The best way to reach that goal is to involve operators in the design decisions and safety analyses. Operators are sometimes left out of the conceptual design stages and only brought in later in development. To design safer systems, operators and maintainers must be included in the design process starting from the conceptual development stage and considerations of human error and preventing it should be at the forefront of the design effort.

Many companies, particularly in aerospace, use integrated product teams that include, among others, design engineers, safety engineers, human factors experts, potential users of the system (operators), and maintainers. But the development process used may not necessarily take maximum advantage of this potential for collaboration. The process outlined in part III tries to do that.

6.3.3 Operations

Once the system is built, it must be operated safely. System engineering creates the basic information needed to do this in the form of the safety constraints and operating assumptions upon which the safety of the design was based. These constraints

and assumptions must be passed to operations in a form that they can understand and use.

Because changes in the physical components, human behavior, and the organizational safety control structure are almost guaranteed to occur over the life of the system, operations must manage change in order to ensure that the safety constraints are not violated. The requirements for safe operations are discussed in chapter 12.

It's now time to look at the changes in system engineering, operations, and management, based on STAMP, that can assist in engineering a safer world.