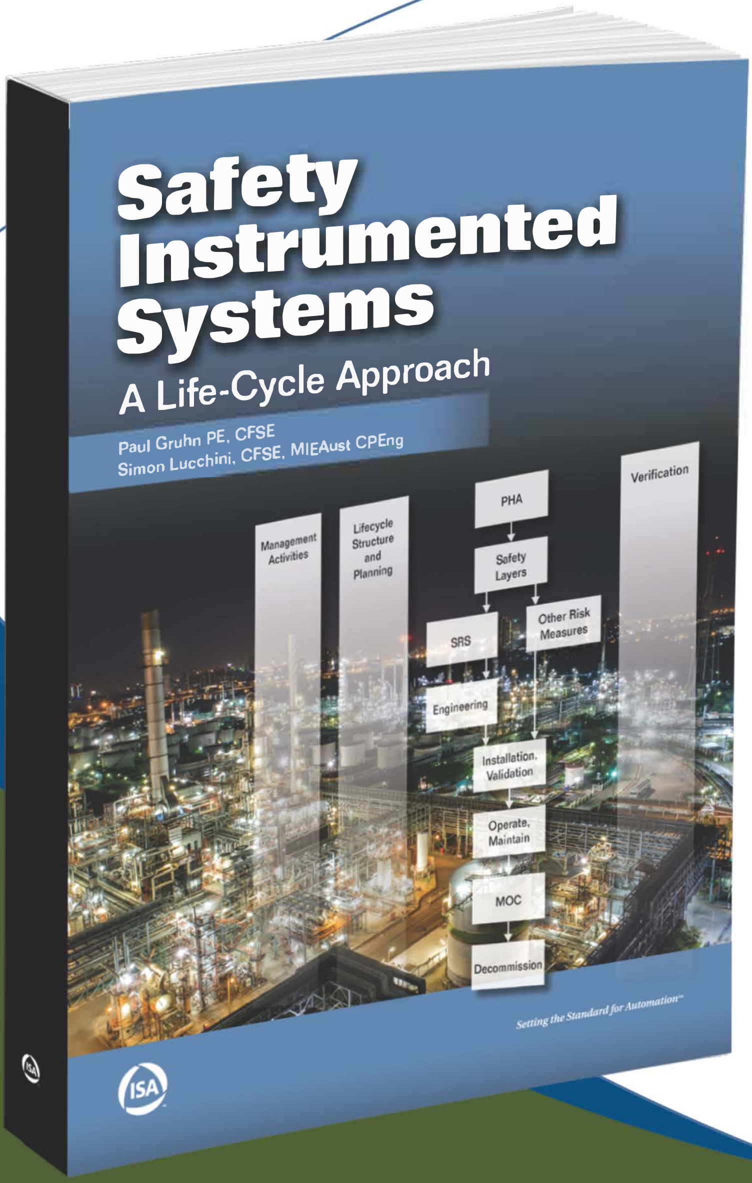
Standards

Certification

Education & Training

Publishing

Conferences & Exhibits



eBook available!

Table of Contents

View Excerpt

Buy the Book

Setting the Standard for Automation™

Safety Instrumented Systems:

A Life-Cycle Approach

By Paul Gruhn, PE, CFSE
Simon Lucchini, CFSE, MIEAust, CPEng

ISA™

Notice

The information presented in this publication is for the general education of the reader. Because neither the author nor the publisher has any control over the use of the information by the reader, both the author and the publisher disclaim any and all liability of any kind arising out of such use. The reader is expected to exercise sound professional judgment in using any of the information presented in a particular application.

Additionally, neither the author nor the publisher has investigated or considered the effect of any patents on the ability of the reader to use any of the information in a particular application. The reader is responsible for reviewing any possible patents that may affect any particular use of the information presented.

Any references to commercial products in the work are cited as examples only. Neither the author nor the publisher endorses any referenced commercial product. Any trademarks or tradenames referenced belong to the respective owner of the mark or name. Neither the author nor the publisher makes any representation regarding the availability of any referenced commercial product at any time. The manufacturer's instructions on the use of any commercial product must be followed at all times, even if in conflict with the information in this publication.

Printed in the United States of America.
10 9 8 7 6 5 4 3 2

ISBN: 978-1-945541-54-4

ISA
67 T. W. Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

Library of Congress Cataloging-in-Publication Data in process

Contents

About the Authors

Paul Gruhn, PE, CFSE

Paul Gruhn is a global functional safety consultant with aeSolutions in Houston, Texas.

Paul is an ISA Life Fellow, a 25+ year member and co-chair of the ISA84 standards committee (on safety instrumented systems), the developer and instructor of ISA courses on safety systems, the author of two ISA textbooks, and the developer of the first commercial safety system software modeling program over 20 years ago.

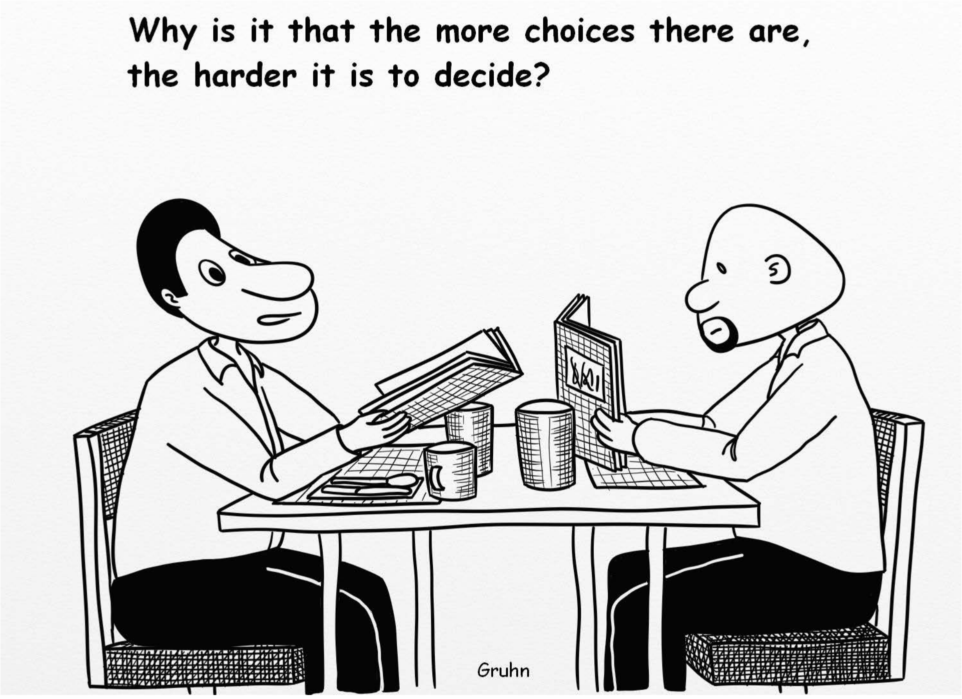
Paul has a BS in mechanical engineering from Illinois Institute of Technology. He is a licensed Professional Engineer (PE) in Texas, a member of the control systems engineering PE exam committee, and both a Certified Functional Safety Expert (CFSE) and an ISA84 Safety Instrumented Systems Expert.

Simon Lucchini, CFSE, MIEAust, CPEng

Simon Lucchini received Bachelor's degrees in both electrical engineering and science from Sydney University, Australia, and he is a Charted Professional Engineer in Australia. Lucchini has worked for 23 years in the petrochemical industry in roles ranging from operations and maintenance to corporate engineering and project engineering. He has worked at Fluor for the past 16 years in the Control Systems department. He is a Fluor Fellow in safety systems design and the chief controls specialist based at the Fluor Calgary, Alberta, Canada office.

Lucchini has written papers on safety systems for various industries and academic venues, including two chapters in Béla Lipták's *Instrument and Automation Engineers' Handbook: Process Measurement and Analysis*, fifth edition, published in 2016. Lucchini is currently the chair of the Safety Systems Committee for the International Society of Automation (ISA) Safety and Security Division for which he produces web articles on matters of importance for the safety systems industry. He is also an active contributor to local control system networks that include several global oil and gas operators.

1 Introduction


Why is it that the more choices there are, the harder it is to decide?
Gruhn

"Engineering responsibility should not require the stimulation that comes in the wake of catastrophe." ~ S. C. Florman

What Is a Safety Instrumented System?

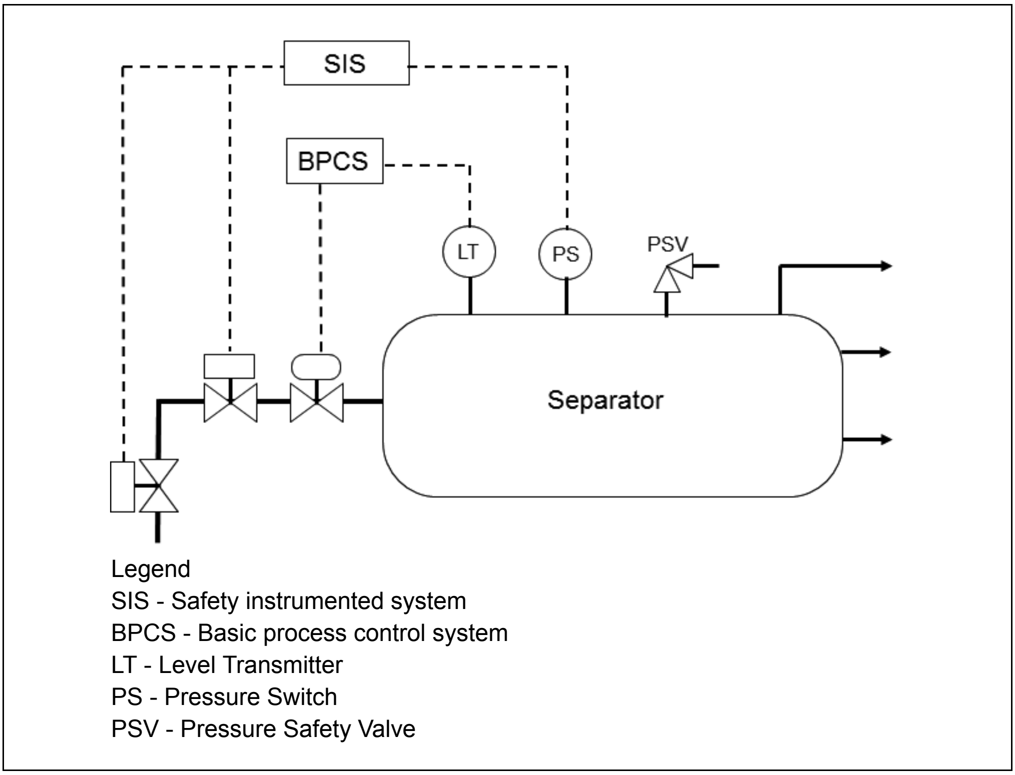

SIS
BPCS
LT
PS
PSV
Separator
Legend
SIS - Safety instrumented system
BPCS - Basic process control system
LT - Level Transmitter
PS - Pressure Switch
PSV - Pressure Safety Valve

Figure 1-1. Control Function and Safety Function

Figure 1-1 shows a control function and a safety instrumented function. As the name implies, control functions *control* pressure, level, temperature, flow, and the like. Early systems in the process industries were purely mechanical/pneumatic, then electronic, and are now software based. Do you believe control functions are perfect and will never fail? (That question usually draws giggles and grins in classes.) Do you believe designers and engineers can envision *every* possible hazardous situation that could occur and design control systems to prevent *all* of them? If that were the case, we would not need to install alarm systems (as there would never *be* an alarm), relief valves (as there would never *be* an overpressure), flare systems (as there would never *be* a process upset), or fire and gas systems (as there never would *be* a release). We obviously don't live in such a dream world. There are many reasons why process facilities are designed with multiple layers of protection.

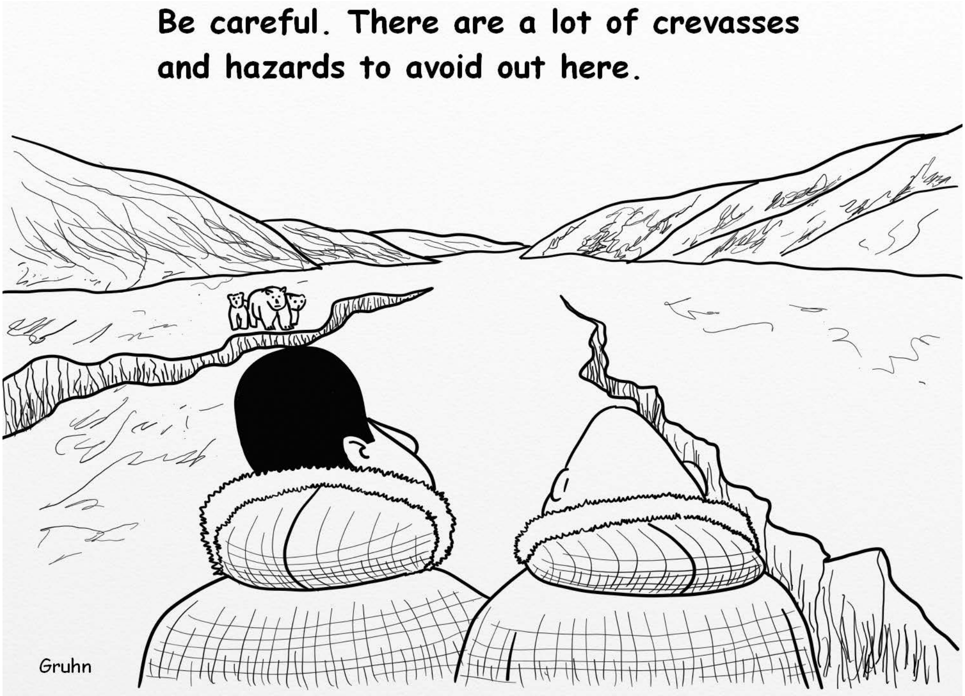
When a control function fails, the next layer of defense is often a safety instrumented function. The safety instrumented function in the process industry by and large does not *control* anything. It *monitors* many of the same variables, but only takes actions when a variable is outside its normal range, which generally means the control function has failed. The typical action of the safety function is to shut down the process or bring it to a predetermined safe state (e.g., recycle). This is a fundamentally different strategy compared to some other industries, such as aircraft. We don't really want to shut down the flying process at an altitude of 35,000 feet!

Control function failures most often conjure up notions of "things" breaking down (e.g., pressure transmitter electronics burning out or going out of calibration). However, as modern digital electronics have become more reliable with respect to random faults, other classes of failure may be prevalent. Systematic failures and human actions may be the initiating causes for a potential hazard. Furthermore, as the software-based control systems become more complex, hazards are frequently emergent properties and may not be related to any physical/permanent fault (i.e., a transient interaction between the process, control system, safety function, and the human operator). Questions may then include, "Does the safety function guard against these types of failures? Has the safety function been designed to be robust with respect to systematic failures?"

Systems performing safety functions have gone by many different names: emergency shutdown system, safety shutdown system, instrument protection system, safety interlock system, safety instrumented system, and more. Different companies within the process industry still use a variety of names for these systems. The shortest and perhaps most generic term might be *safety system*, but this too means different things to different people. For many chemical engineers, "safety systems" refer to management procedures and practices, not instrumented systems. One very common term has been *emergency shutdown system* (ESD), but to electrical engineers, ESD means electrostatic discharge. To some, ESD is a means of *manually* shutting down the process independent to the safety system. Many don't want the word *emergency* in the name at all, due to its negative connotation.

When the American Institute of Chemical Engineers' Center for Chemical Process Safety (AIChE CCPS) published the first edition of *Guidelines for Safe Automation of Chemical Processes* in 1993 [1], the term it used was *safety interlock system*—SIS. Some members of the ISA84 committee thought the term "interlocks" was only one subset of many different types of safety-related systems.

2
Design Life Cycle


Be careful. There are a lot of crevasses and hazards to avoid out here.
Gruhn

"If I had 8 hours to cut down a tree, I'd spend 6 hours sharpening the axe."
~ A. Lincoln

Designing a single component may be a relatively simple matter, one that a single person can handle. Designing any large *system*, however, whether it's a car, an airplane, or a refinery is beyond the ability of any single individual. The instrument or control system engineer should *not* feel that all the tasks associated with designing a safety instrumented system are his or her responsibility alone, because they're not. The design of a system, including a safety instrumented system, requires a multidisciplinary *team*.

Hindsight/Foresight

"Hindsight can be valuable when it leads to new foresight."
~ *P.G. Neumann*

Hindsight is easy. Everyone always has 20/20 hindsight. *Fore*sight, however, is a bit more difficult. Foresight is required, however, with today's large, high-risk systems. We simply can't afford to design large process facilities by trial and error. The risks are too great to learn that way. We have to try and prevent certain accidents, no matter how remote the possibility, even if they have never yet happened. This is the subject of *system safety*.

System safety was born out of the military and aerospace industries. The military have many obvious high-risk examples. The following case may have been written in a lighthearted fashion, but was obviously a very serious matter to the personnel involved. Luckily, there were no injuries.

> *An ICBM silo was destroyed because the counterweights, used to balance the silo elevator on the way up and down, were designed with consideration only to raising a fueled missile to the surface for firing. There was no consideration that, when you were not firing in anger, you had to bring the fueled missile back down to defuel. The first operation with a fueled missile was nearly successful. The drive mechanism held it for all but the last five feet when gravity took over and the missile dropped back down. Very suddenly, the 40-foot diameter silo was altered to about 100-foot diameter* [1].

A radar warning system in Greenland suffered an operational failure in the first month. It reported inbound Russian missiles, but what it actually responded to was...*the rising moon*.

If you make something available to someone, it will at some point be used, even if you didn't intend it to be. For example, there were two cases where North American Aerospace Defense (NORAD) and Strategic Air Command

(SAC) went on alert because radar systems reported incoming missiles. In reality, someone had just loaded a training tape by mistake. After the same incident happened a *second* time, it was finally agreed upon to store the training tapes in a different location. What might have originally been considered human error was actually an error in the *system* that allowed the inevitable human error to happen.

Findings of the HSE

The U.K. Health and Safety Executive (HSE) examined 34 accidents that were the direct result of control and safety system failures in a variety of different industries [2]. Their findings are summarized in Figure 2-1. Most accidents could have been prevented. The largest percentage of accidents (44%) was due to *incorrect and incomplete specifications*. Specifications consist of both the *functional* specification (i.e., what the system should do) and the *integrity* specification (i.e., how well it should do it).

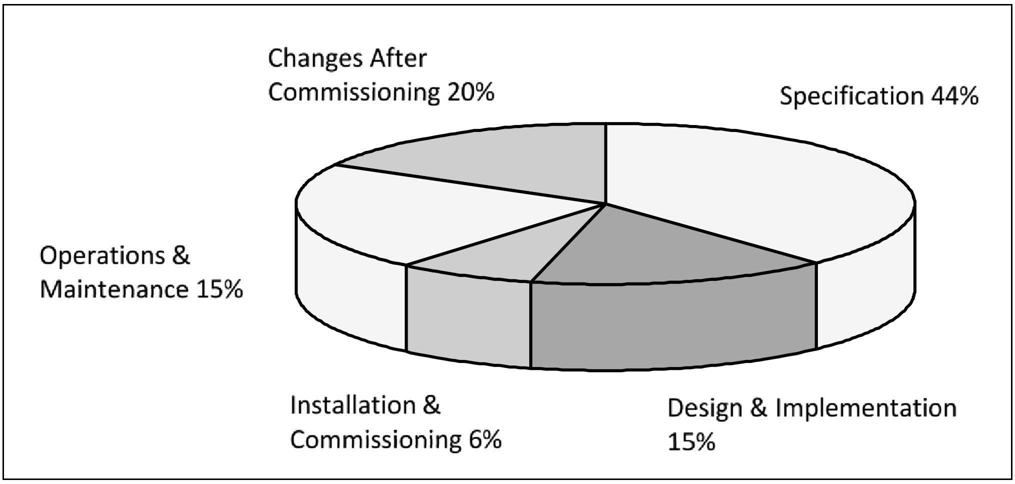
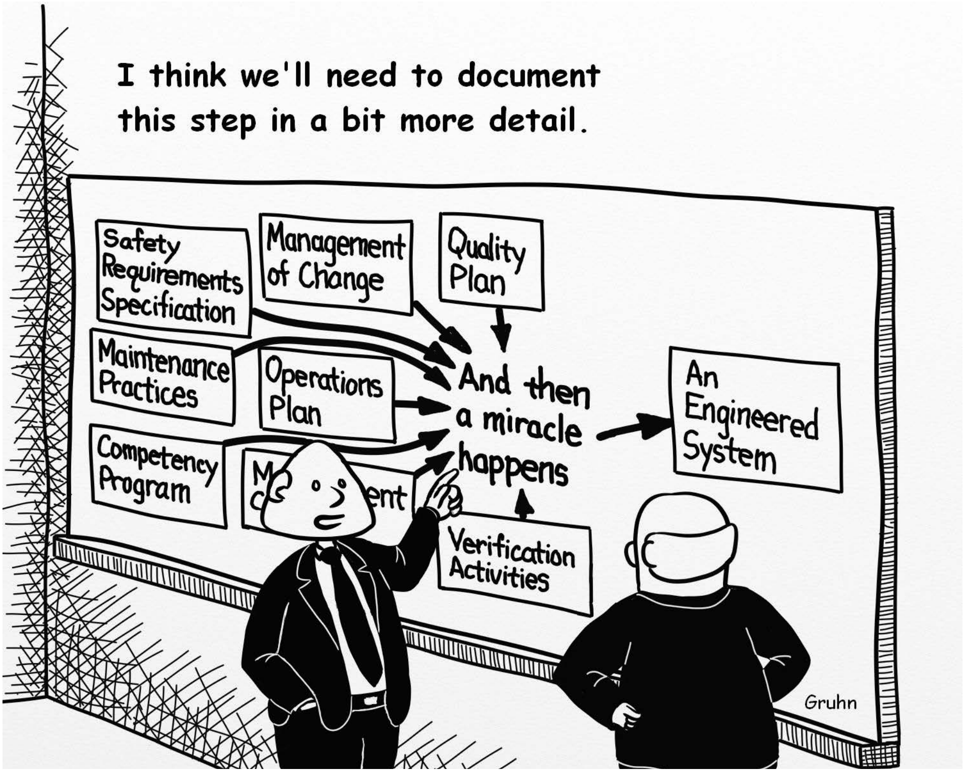


Figure 2-1. Finding of the U.K. HSE, Control and Safety System Failure Causes

Someone on the ISA safety listserv described a case where when their safety instrumented system shut down, it de-energized all the outputs. You might think that's what a system is normally supposed to do, but that's not always true. In this case the system de-energized the outputs to the lubricating oil pumps on their turbines and compressors. This specification error caused the rotating equipment to essentially self-destruct. Those outputs should have been specified to fail in their last state, which virtually any system could do.

3

Project Management



"One of the true tests of leadership is the ability to recognize a problem before it becomes an emergency." ~ Arnold H. Glasow

Everyone Has a Functional Safety Plan, Right?

The ISA/IEC 61511 standard provides guidance for implementing the safety instrumented system (SIS) life-cycle phases. However, the scope of a safety system project can vary considerably. The SIS may be part of a new multibillion dollar process plant, a facility revamp, or just involve the addition of a few safety functions to an existing installation. Even though the basic steps may be similar, the execution will vary considerably, depending on the overall scope and makeup of the project.

The overall project schedule and resourcing are most often governed by scope other than the safety system. A large project may take 4 to 7 years from concept to start-up. The functional safety engineer has to navigate many interfaces in order to formulate a solid SIS design basis (i.e., the safety requirements specification). It's important to understand the complexities that arise from these project interfaces since they need careful management. This chapter will consider how a project works, what are the critical interfaces for implementing safety systems, and when to make timely decisions.

Safety Life-Cycle and Real-World Project Complications

ISA/IEC 61511 (Figure 7 and Table 2) provides an overall plan for the SIS safety life-cycle phases [1, 2]. Armed with this information, the functional safety engineer may feel that he/she can tackle any project. However, even though the basic steps will be similar in concept, the execution will vary considerably depending on the overall scope and makeup of the project. To this point, Clause 6.1 of the standard qualifies the safety life cycle with the following statement:

> *"NOTE 1 The overall approach of the IEC 61511 series is shown in Figure 7. It can be stressed that this approach is for illustration and is only meant to indicate the typical SIS safety life-cycle activities from initial conception through decommissioning."*

Importantly, the overall project schedule and resourcing is most often governed by scope other than that of the safety system. The overall control systems content ranges from 8% to 12% of total installed cost (TIC). The safety system may be less than 10% to 20% of the control systems budget (i.e., approximately 1% to 2% of TIC). Engineering content may be less than 6% of TIC (i.e., safety system engineering may be as little as 0.06% of TIC). In other words, the main engineering effort is not focused on the safety system.

There are various project execution elements that can further complicate the design and implementation of a safety system. Some of these include:

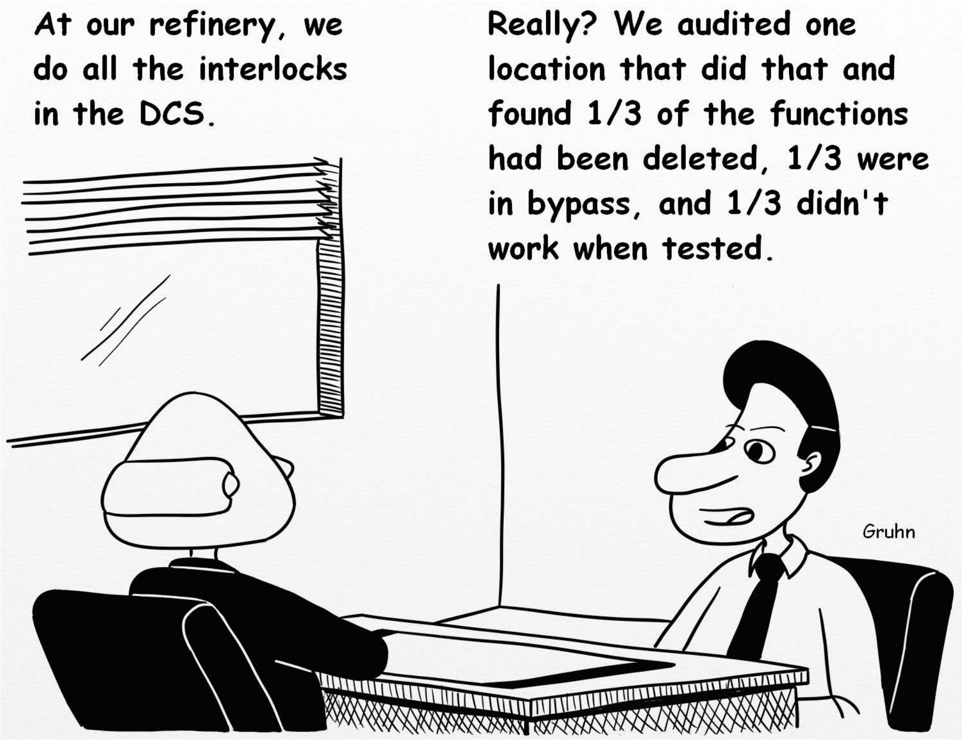
- Modularization and distributed safety systems (e.g., truckable modules to very large modules [VLMs], distributed SIS input/output [I/O] modules across different plant units)
- Process licensor packages and their specific safety requirements (e.g., polypropylene reactor, synthetic crude hydrocracker, coker unit, liquefied natural gas process, ethylene oxide reactor, and many more)
- Mechanical vendor packages with their diverse designs, hardware implementations, and code requirements (e.g., vessels, reactors, compressors, fired heaters, turbines, and effluent treatment)
- Multiple engineering contractors covering various plant units, depending on the overall plant split
- Varying design practices and procedures from the end user, process licensors, and engineering contractors

Furthermore, the automation scope for the whole plant may be undertaken by a specialist automation company (e.g., main automation contractor [MAC]). The MAC is typically an independent entity from the main engineering contractors (e.g., engineering procurement and construction [EPC]), who are responsible for the overall design of the individual plant units. This automation company may be the basic process control system (BPCS) and SIS vendor(s), or third-party system integrators. Therefore, there will be a number of groups, including the process licensors, who have a stake in the implementation of the safety system(s).

Thrown into this mix are the end-user standards and specifications that may not exactly align with the project execution practices of the various engineering companies, process licensors and other automation parties. Essentially, the safety engineers have to navigate many interfaces in order to formulate a solid SIS design basis (i.e., safety requirements specification as discussed in Chapter 6). It's important to understand the complexity that arises from these interfaces. They need careful management. Complexity can be considered as proportional to 2 to the power N, where N is the number of interfaces (i.e., $C=2^N$). This isn't a recognized equation to be found in textbooks. However, the authors have found this a useful model to explain to others how quickly things can get out of hand!

4

Process Control versus Safety Control


At our refinery, we do all the interlocks in the DCS.
Really? We audited one location that did that and found 1/3 of the functions had been deleted, 1/3 were in bypass, and 1/3 didn't work when tested.
Gruhn

"Nothing can go wrong, click… go wrong, click… go wrong, click…"
~ Anonymous

Process control used to be performed in mechanical/pneumatic, analog, single-loop controllers. Safety functions were performed in different hardware, typically hardwired relay systems. Electronic distributed control systems (DCSs) started to replace single-loop controllers in the 1970s. Programmable logic controllers (PLCs) were developed to replace relays in the late 1960s. Since both systems are software programmable, some people naturally concluded that there would be benefits in performing both control and safety functions within the same system, usually the DCS. The typical benefits touted included single source of supply, integrated communications, reduced training and spares, simpler maintenance, and potentially lower overall costs. Some believe that the reliability, as well as the redundancy, of modern DCSs are "good enough" to allow such combined operation. However, all domestic and international standards, guidelines, and recommended practices clearly recommend separation of the two systems. The authors agree with this recommendation and wish to stress that the reliability of the DCS is not the issue.

Control and Safety Defined

Critical systems require testing and thorough documentation. It's debatable whether normal process control systems require the same rigor of testing and documentation. When the U.S. government came out with their process safety management (PSM) regulation (29 CFR 1910.119) in 1992, many questioned whether the mandated requirements for documentation and testing applied to both the control systems as well as the safety systems. For example, most organizations have documented testing procedures for their safety instrumented systems, but the same may not be said for all their control system loops. Users in the process industry questioned OSHA representatives as to whether the requirements outlined in the PSM regulation applied to all 10,000 loops in their DCS, or just the 300 in their safety instrumented systems. OSHA's response was that it included everything. Users felt this was another nail in the proverbial coffin trying to put them out of business.

This helped fuel the development of ISA-91.01-1995, *Identification of Emergency Shutdown Systems and Controls that Are Critical to Maintaining Safety in Process Industries*. The ISA-91 standard was reaffirmed in 2001. This brief, two-page standard included definitions of process control, safety control, and safety critical control. The scope of ISA-91 was then passed to the ISA84 committee and ANSI/ISA ISA-84.91.01-2012 was published in 2012 with a new title *Identification and Mechanical Integrity of Safety Controls, Alarms, and Interlocks in the Process Industry*. Another industry acronym was introduced: SCAI (pronounced “sky”), standing for, as in the title of the standard, safety controls, alarms, and interlocks.

The basic process control system (BPCS)—as opposed to an “advanced” process control system—is the control equipment that performs the normal regulatory functions for the process (e.g., proportional-integral-derivative [PID] control and sequential control). Some have stated that this accounts for up to 95% of instrumentation for most land-based facilities. Most people accomplish this with a DCS, PLC, or hybrid system.

SCAI are essentially the instrumentation and controls used to achieve or maintain a safe state for a process. They provide risk reduction with respect to a specific hazardous event. Some have stated that this accounts for less than 10% of instrumentation for most land-based facilities.

Based on these definitions, users stated to OSHA that their 10,000 DCS loops were not safety-related and therefore did not require the same degree of rigor for documentation and testing as their 300 safety instrumented loops.

This is not meant to imply that the design of distributed control systems doesn’t require thorough analysis, documentation, and management controls. They obviously do, just not to the same extent as safety systems.

Process Control – Active/Dynamic

It’s important to realize and understand the fundamental differences between process control and safety control. Process control systems are active, or dynamic. They have analog inputs, analog outputs, perform math and number crunching, and have feedback loops. Therefore, most failures in control systems are inherently self-revealing. For example, consider the case of a robot on an automated production line. Normally the robot picks up part A and places it in area B. If the system fails, it’s obvious to everyone; it no longer places part A in area B. There’s no such thing as a “hidden” failure. The system either works or it doesn’t. There’s only one failure mode with such systems—revealed—and you don’t need extensive diagnostics to annunciate such failures. If a modulating process control valve were to fail fully open, fully closed, or stuck in place, it would most likely impact production and the problem would become evident to everyone fairly quickly. Again, extensive testing and diagnostics are usually not required to reveal such failures or problems.

5

Protection Layers

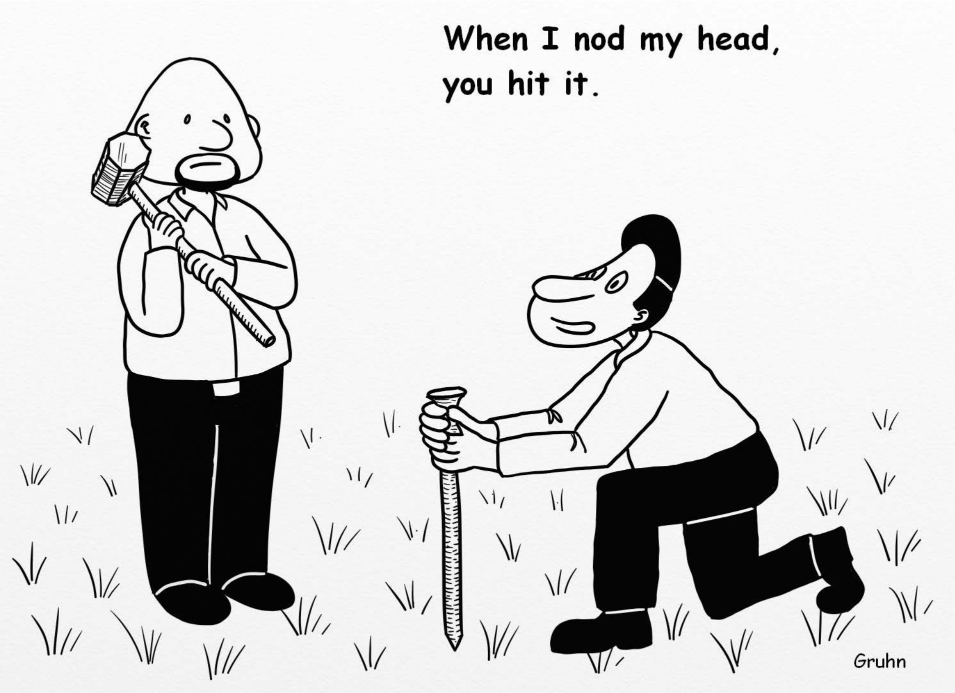
The last trip generated 17,000 alarms. The big guy wants your analysis and report in one hour.
Gruhn

Accidents rarely have a single cause. Accidents are usually a combination of rare events that people initially assumed were independent and would not happen at the same time. Take, as an example, the worst chemical accident to date, Bhopal, India, where an estimated 3,000 people died and 200,000 were injured [1].

The material that leaked in Bhopal was MIC (methyl isocyanate). The release occurred from a storage tank that held more material than allowed by company safety requirements. Operating procedures specified using the refrigerant system of the storage tank to keep the temperature of the material below 5°C. A temperature alarm would sound at 11°C. The refrigeration unit was turned off due to financial constraints and the material was usually stored at nearly 20°C. The temperature alarm threshold was changed from 11°C to 20°C. A worker was tasked to wash out some pipes and filters, which were clogged. Blind flanges were not installed as required. The worker with the hose knew this, yet he did not believe it was his job or responsibility to insert a flange. Water leaked past a check valve into the tank containing MIC. Temperature and pressure gauges that indicated abnormal conditions were ignored because they were believed to be inaccurate. A vent scrubber, which could have neutralized the release, was not kept operational because it was presumed not to be necessary when production was suspended (as it was at the time). The vent scrubber was inadequate to handle the size of the release anyway. The flare tower, which could have burned off some of the material, was out of service for maintenance; part of the piping to the flare was removed and being replaced. The flare was also not designed to handle the size of the release. Material could have been vented to nearby tanks, but gauges erroneously showed them to be partially filled. A water curtain was available to neutralize a release, but the MIC was vented from a stack 108 feet above the ground, too high for the water curtain to reach. Workers became aware of the release due to the irritation of their eyes and throats. Their complaints to management at the time were ignored. Workers panicked and fled, ignoring four buses that were intended to be used to evacuate employees and nearby residents. The MIC supervisor could not find his oxygen mask and broke his leg climbing over the boundary fence. When the plant manager was later informed of the accident, he said in disbelief, “The gas leak just can’t be from my plant. The plant is shut down. Our technology just can’t go wrong, we just can’t have leaks.” What is potentially even more egregious is that the company management had been warned many times of safety deficiencies. Many audits were performed outlining deficiencies and making recommendations. None of the recommendations were implemented.

The Texas City refinery disaster in 2005 is another classic example of multiple, diverse layers not functioning as designed or intended. The level control loop on the distillation column could not measure more than 10 feet and had not been calibrated for the fluid being run at that time. Operators did not follow start-up procedures and filled the column higher than called for. While the level transmitter indicated a high alarm, the separate high-level switch did not, as it had been damaged and was not functional. The sight glass had been dirty and unreadable for years, despite complaints and requests for replacement. Testing of all instrumentation prior to start-up—a company requirement—was checked off as being completed, yet personnel later admitted that the testing was *not* done due to the time pressures associated with the start-up. The start-up was split between shifts and there was an incomplete exchange of information between shifts. Operators had worked 12-hour shifts for *30 days* prior to start-up for the maintenance turnaround, so you can imagine they were most likely not performing optimally. A shift supervisor left due to a family emergency, leaving one lone operator to handle three different process units, while one of them was going through start-up. Requests had been made over the years for an additional operator, but the requests were denied for budgetary reasons. The distillation column was completely filled, but the operator had no idea anything was wrong, as the instrumentation could not measure beyond 10 feet. The column filled because an outlet valve that was supposed to be open was actually closed. Emergency relief valves did not lift at the appropriate gas pressure. They eventually lifted when there was liquid in the discharge piping. Liquid was vented to an atmospheric blowdown drum, one first installed in the 1950s, and later replaced in the 1990s. The high-level switch in the blowdown drum did not function since it had been damaged due to improper maintenance and testing. The blowdown drum was filled and emitted a geyser-like eruption for over a minute. An outdoor operator finally informed the indoor operator of the problem. Procedure said there were not supposed to be any vehicles in the unit, yet a running diesel truck just 25 feet from the blowdown drum was the source of ignition. Procedure said there were not supposed to be temporary trailers within 350 feet of a process unit, but the closest trailer where most of the contractors were killed was less than 130 feet away. Years prior, a hazard analysis had been started to examine the impact of having a trailer so close. The analysis was never completed, and it was never officially determined to be acceptable to have trailers that close. No unnecessary personnel were supposed to be in the area during start-up. The 15 contractors that were killed were there to start up a nearby unit, and were never even informed that this particular unit was

6 Safety Requirements Specification



"The wise man learns from the mistakes of others." ~ Otto von Bismarck

The Need to Specify versus the Desire to Design and Build

A "blueprint" is a nineteenth-century term describing the reproduction of a technical drawing used to document a naval, architectural, or engineering design. Today, it's sometimes used to describe the design of a process, vessel, automation system, or mechanical package. Very often, the focus of engineering is finalizing the blueprint. Engineers want to build things!

However, the important first step is to decide what to design and build. It's like designing an automobile. Does the end user need a truck, minivan, or sports car? There will be serious consequences if this requirement isn't confirmed before the design gets too far. Having to fix the design of the tailgate on a truck for a rancher using the vehicle on a cattle farm isn't too much of an issue. However, rectifying inadequate ground clearance because the assumed specification was for a sports car is an entirely different matter.

Similarly, discovering that the SIS logic solver should be SIL 3–capable when the architecture only supports SIL 2 will be problematic. Adding additional safety functions to a system is normally seen as design development and of lesser concern.

This chapter considers the development of the safety requirements specification (SRS) for an SIS. Important inputs to the SRS are the PHA/HAZOP (i.e., process hazard analysis/hazard and operability study to identify credible hazards) and the safety integrity level (SIL) allocation/determination review for the required safety instrumented functions (SIFs). The SRS documents the functional and integrity requirements for each SIF, which protects people, the environment, and the facility from harm caused by potential hazards.

ISA/IEC 61511 [1, 2] provides details for both the hardware and software components of the SRS. However, it's also important to consider its use for the various phases of the safety life cycle. The SRS should support engineering, installation, commissioning and operations. This may mean re-organizing the information to suit the different needs.

Process facilities are becoming larger and the controls are more tightly integrated between the various units. As a result, the design of safety systems is becoming more complex. It's becoming more difficult to analyze the requirements and to document them in the SRS. This chapter will review some common problem areas and potential solutions.

Specifications, Requirements, and Incidents

Clause 10.2, from ISA/IEC 61511, provides the general philosophy for developing an SRS. It should be noted that H&RA is an acronym for *hazard and risk assessment*.

"The safety requirements shall be derived from the allocation of SIF and from those requirements identified during H&RA. The SIS requirements shall be expressed and structured in such a way that they are:

- clear, precise, verifiable, maintainable, and feasible;
- written to aid comprehension and interpretation by those who will utilize the information at any phase of the safety life cycle."

Implicit in these general requirements is that the SRS should support a management of change process (MOC). In other words, change can't be controlled unless there is a clearly documented basis for the design.

Many publications, papers, and presentations about process safety include the results of a study by the U.K. Health and Safety Executive of 34 incidents caused by control and safety system failures. As shown in Chapter 2 (Figure 2-1), the main contribution to failures arose from the incorrect and incomplete specification of system requirements.

Similar statistics can be seen in Figure 6-1, by H. Thimbleby [3], demonstrating the distribution of the causes of software errors.

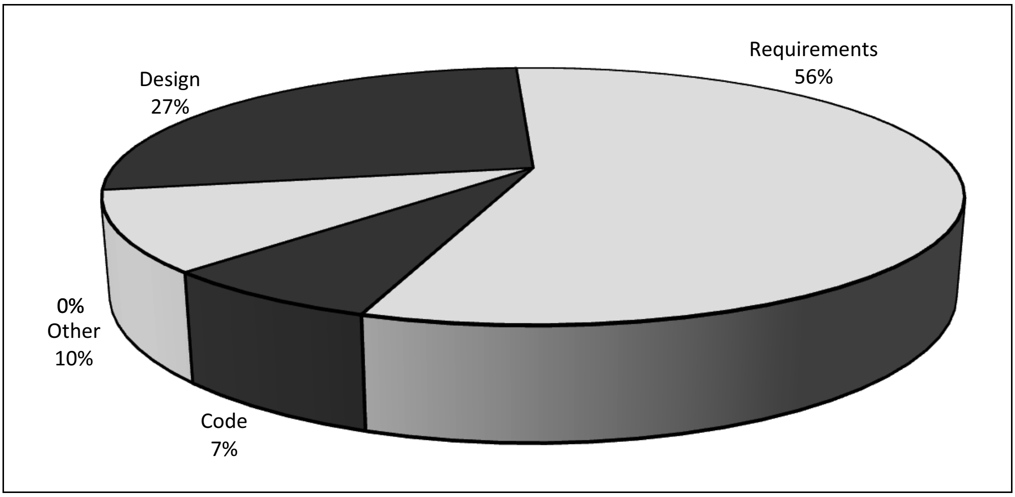


Figure 6-1. Failure Cause Distribution of Software Errors

7

Selecting Safety Integrity Levels (SIL)

Now let me get this straight. You smoke two packs a day, but still wear seat belts?
Gruhn

"The man who insists upon seeing with perfect clearness before deciding, never decides."
~ H.F. Amiel

Introduction

Today's safety instrumented system (SIS) standards are performance-based, not prescriptive. They do not mandate a technology, level of redundancy, test interval, or system logic. Essentially they state, "The greater the level of risk, the better the systems needed to control it." There are a variety of methods for evaluating risk. There are also a variety of methods for equating risk to the performance required from an SIS. One term used to describe safety system performance is safety integrity level (SIL).

Many industries have the need to evaluate and rank risk. Management decisions may then be made regarding various design options. For example, how remote, if at all, should a nuclear facility be to a large population zone? What level of redundancy is appropriate for a military aircraft weapons control system? What strength should jet engine turbine blades be for protection against flying birds? How long should a warranty period be based on known failure rate data? Ideally, decisions such as these would be made based on mathematical analysis. Realistically, quantification of *all* factors is extremely difficult, if not impossible, and subjective judgment and experience may still be considered.

Military organizations were some of the first groups to face such problems. For example, when someone has to press the button that might start or stop World War III, one would like to think that the probability of the electronic circuitry working properly would be rated as something other than "high." The U.S. military developed a standard for categorizing risk: MIL-STD 882 "Standard Practice for System Safety," [1] which has been adapted by other organizations and industries in a variety of formats.

Different groups and countries have come up with a variety of methods of equating risk to safety system performance. Some are qualitative, and some are more quantitative. No method is more correct or better than another.

It's important to clarify what SIL is and what it is not. SIL is a measure of the performance of a single safety instrumented function. A function consists of a sensor, logic solver, and final element(s). Therefore, it is incorrect to refer to the SIL of an individual component of a system (e.g., a logic solver in isolation). A chain is only as strong as its weakest link. A logic solver could be rated for use in SIL 3, but if connected with nonredundant field devices with infrequent testing, the overall system may only meet SIL 1. It is more appropriate to use phrases such as "SIL claim limit" and "SIL capability" in order to distinguish between component and system performance. It is useful to refer to the probability of failure on demand (PFD) of an individual component, but this will be based on assumed failure rates and test intervals that should also be stated. Unfortunately, PFD numbers are very small and are usually referred to using scientific notation (e.g., 5 E-3), which can be difficult for some to relate to. Many prefer the reciprocal of PFD, the risk reduction factor or RRF (e.g., 200).

SIL does not apply to an entire system consisting of dozens or hundreds of functions. That would be like asking, "What's the combined speed of all the cars in the parking lot?" SIL does not apply to a piece of equipment, such as a compressor. SIL is not directly a measure of process risk. It would be incorrect to say, "We've got an SIL 3 process."

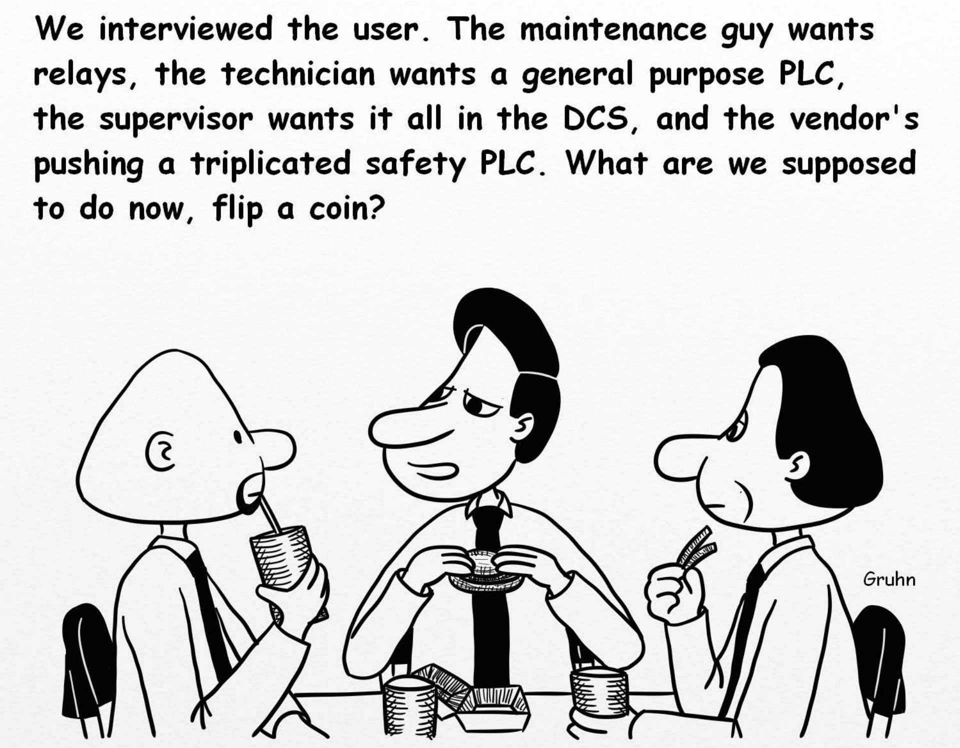
Who's Responsible?

Selecting or determining SILs is mentioned in a variety of safety instrumented system standards. Many therefore assume the task is the responsibility of the instrument or control system engineer. This is not the case. Evaluating the process risk and selecting the appropriate safety integrity level is a responsibility of a multidisciplinary *team*, not any one individual. A control system engineer may—and certainly should—be involved. Yet the review process will also require other specialists, such as those typically involved in any process hazards analysis (PHA), such as a hazard and operability study (HAZOP). Some organizations prefer to select SILs during the PHA. Others believe it's not necessary (or even desirable) to involve the entire HAZOP team. Therefore, some organizations perform SIL selection studies separately after the PHA with a subset of the same PHA team members. As a minimum, representatives from the following departments should participate in any SIL selection study: process, mechanical design, safety, operations, and control systems.

Which Technique?

When it comes to hazard and risk analysis and determining safety integrity levels, there are no answers that could be categorized as either right or wrong. There are many ways of evaluating process risk, none more correct than another. Various industry documents describe multiple qualitative and quantitative methods for evaluating risk and determining SIS performance [2–5]. The methods were developed at different times by different countries. No method should be viewed as more correct or accurate than another. All meth-

8 Choosing a Technology


We interviewed the user. The maintenance guy wants relays, the technician wants a general purpose PLC, the supervisor wants it all in the DCS, and the vendor's pushing a triplicated safety PLC. What are we supposed to do now, flip a coin?
Gruhn

"If architects built buildings the way programmers wrote software, the first woodpecker that came along would destroy civilization." ~ Unknown

There are several technologies available for use in safety instrumented systems: pneumatic, electromechanical relays, solid state, and programmable logic controllers (PLCs). There is no one overall best technology, just as there is no overall best car (vendor claims notwithstanding). Each technology has advantages and disadvantages. It's not so much a matter of which is best, but rather which is most appropriate, based on factors such as budget, size, level of risk, complexity, flexibility, maintenance, interface and communication requirements, and security.

Pneumatic Systems

Pneumatic systems are rarely used anymore. They were used in the past in offshore and in remote locations where systems needed to operate without electrical power. Pneumatic systems are relatively simple (assuming they're small) and relatively fail-safe. Fail-safe in this sense means that a failure or leak would usually result in the system depressurizing, which would initiate a shutdown. Clean, dry gas is generally necessary. If desiccant dust from instrument air dryers or moisture from ineffective drying or filtering enters the system, small ports and vents utilized in the pneumatic circuits will suffer from plugging and sticking. This can render the circuits prone to more dangerous failures where the system may not function on demand. Frequent operation and/or testing is usually necessary in order to prevent parts from sticking. Offshore operators in the United States Gulf of Mexico are required to test pneumatic safety systems on a monthly basis for this very reason. Pneumatic systems have been used in small applications where there is a desire for simplicity and intrinsic safety and where electrical power is not available.

Relay Systems

Relay systems offer a number of advantages:

- Simple (at least when they're small)
- Low capital cost
- Immune to most forms of electromagnetic or radio frequency interference (EMI/RFI)
- Available in different voltage ranges
- Fast response time

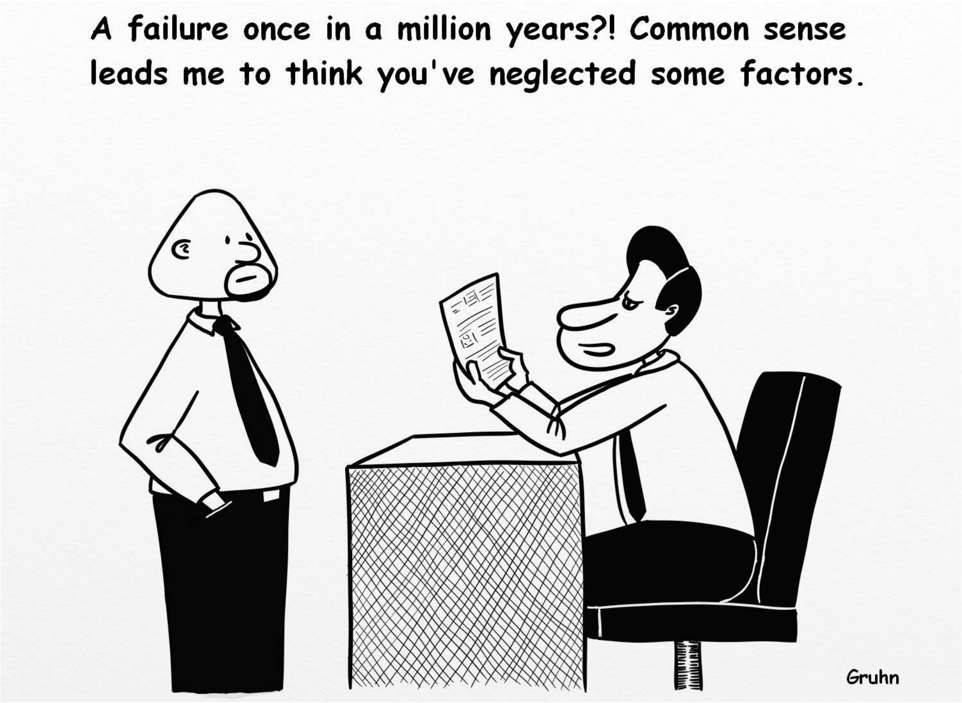
- No software
- Fail-safe (assuming normally energized)

Fail-safe means that the failure mode is known and predictable (usually with closed and energized circuits failing open and de-energized). However, nothing is 100% fail-safe, although there are safety relays offering 99.9+% fail-safe operation.

Relay systems also have a number of disadvantages:

- **Nuisance trips:** Relay systems are typically nonredundant. This means that a failure of a single relay can result in a nuisance trip of the process resulting in lost production. There is usually an elevated level of risk associated with shutting down and starting up any process as well.
- **Complexity of larger systems:** The larger a relay system gets, the more unwieldy it gets. A 10 I/O (input and output) relay system is manageable. A 700 I/O relay system is not. An end user described to one of the authors certain problems they experienced with their 700 I/O relay system. One circuit consisted of a pneumatic sensor that went to an electric switch, that went to a PLC, that went to a satellite (honest!), that went to a DCS, that went to the relay panel, and worked its way out again. (This was obviously not your typical relay system!) The engineer said that between sensor and final element there were 57 signal handoffs! The author asked, “What’s the likelihood that you think that’s going to work when you need it?” The response was, “Zero. I know it won’t work.” Upon further questioning the engineer even admitted, “Everyone in this facility knows don’t go *near* that panel! The mere act of opening panel doors has caused nuisance shutdowns!” This was due to old wiring and worn out insulation and connections that would cause open and short circuits with the slightest vibration. Similar stories have been told at other locations.
- **Manual changes to wiring and drawings:** Any time logic changes are required with a hardwired system, wiring must be physically changed and drawings must be manually updated. Keeping drawings up to date requires strict discipline and enforcement of procedures. If you have a relay panel that is more than a decade old, try the following simple test. Obtain the engineering logic drawings, go out to the panel, and check to see if the two match. You may be in for a bit of a surprise!

9
Initial System Evaluation


A failure once in a million years?! Common sense leads me to think you've neglected some factors.
Gruhn

"There's always an easy solution to every human problem, neat, plausible... and wrong." ~ H.L. Menken

Things Are Not as Obvious as They May Seem

There are an almost endless number of decisions to make in the design of a safety instrumented system. Sensors could be discrete switches or analog transmitters. Their configuration could be single, dual, or triplicated. There are more technology and configuration choices when it comes to logic solvers. Final elements can also be different technologies with multiple possible configurations. Which technology, which configuration, and what proof test intervals should you choose for your next system, and why?

The problem is that things are not as intuitively obvious as they may seem. Dual is not always better than simplex, and triple is not always better than dual. Diagnostic coverage refers to the percentage of failures that can be detected automatically by the system. Would a nonredundant system with 99% diagnostic coverage be better than a triplicated system with 80% coverage? Common cause is a single stressor or failure that impacts a redundant configuration. How much of an impact on the performance of a triplicated system do you think a 5% common cause factor would have? How much of an impact on system performance do you think lowering the manual proof test coverage from 100% to 90% would have? How much of an impact on system performance do you think leaving a function in bypass for 1 day, or 1 week, would have? How do you even derive answers for any of these questions?

Intuition may be fine for some things, but not others. Jet aircraft are not built by gut feel. Bridges are not built by trial and error, at least not anymore. Nuclear power plants are not built by intuition. If you were to ask the chief engineer of the Boeing 777 why they used a particular size engine, how comfortable would you feel if their response was, “Well, we weren’t sure... but that’s what our vendor recommended.”

You’ll learn the answers to the above questions—and more—throughout the rest of this chapter.

Why Systems Should Be Analyzed *Before* They’re Built

Ideally, would you rather perform a process hazards analysis on a plant *before* you build it, or *afterwards*? The obvious answer is before, but not everyone who is asked this question realizes the real reason *why*. It’s *cheaper* to redesign the plant on paper. The alternative would be to rebuild the plant after the fact. The same applies to safety instrumented systems.

Deciding which system and design is appropriate for a given application is not always a simple matter. It's therefore important to be able to analyze systems in a *quantitative* manner. While quantitative analyses may be imprecise and have uncertainties, they are nevertheless valuable for the following reasons:

- They provide an early indication of a system's potential to meet the design requirements.
- They enable one to determine the weak link in the system (and fix it, if necessary).
- They allow an "apples to apples" comparison between different offerings.

Caveats

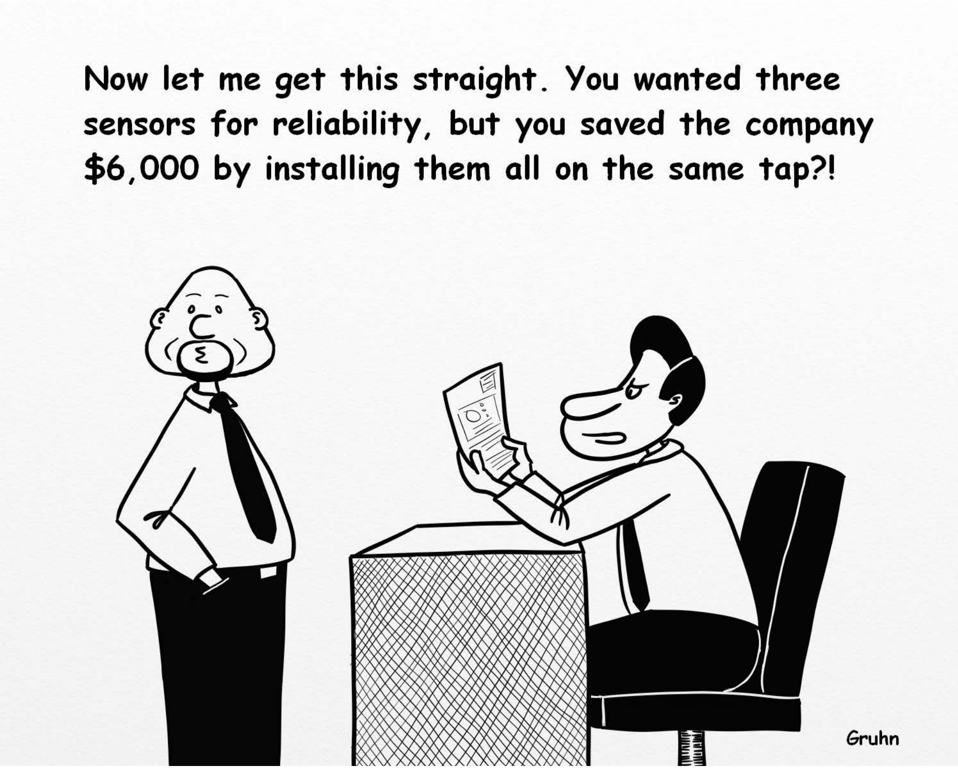
"There are lies, there are damn lies, and then there's statistics." ~ M. Twain

Simple models may be solved by hand using a calculator. As more factors are accounted for, however, manual methods become rather unwieldy. It's possible to develop spreadsheets or other computer programs to automate the process. A major drawback of some models is often not what they include, but what they do *not* include. One can model a triplicated system according to one vendor's optimistic assumptions, and then model it with a more realistic set of assumptions, and change the answer by multiple orders of magnitude! It's generally not the *accuracy* of the modeling technique that matters (all models are wrong, some are just less wrong than others), it's the *assumptions* that go into it. Computers are known for their speed, not their intelligence.

"Computer models can predict performance with great speed and precision, yet they can also be completely wrong!" ~ Unknown

When the Boeing 777 was being designed, there were two different factions regarding the modeling and testing of the engines. One group felt their computer models were so good that testing a real engine was unnecessary. Another group felt that actual testing was essential. The latter group eventually won the argument. Two engines on one side of a 747 were removed and a new single engine intended for the 777, which was twice as powerful as the earlier engines, was installed in their place. The first time the plane took off the new engine flamed out. That particular flaw was not revealed in the com-

10 Field Devices


Now let me get this straight. You wanted three sensors for reliability, but you saved the company $6,000 by installing them all on the same tap?!
Gruhn

"History repeats itself because no one was listening the first time." ~ Anonymous

Where the Real Action Happens!

Sensors and final trip devices are the safety function components that are directly connected to the process. Sensors must be designed to measure select parameters of potentially high pressure and temperature process streams of varying chemical compositions. Additionally, they must deal with corrosion, plugging, and vibration. Trip valves often have to quickly shut off process feeds, which are erosive, fouling, corrosive, or worse. Furthermore, both sensors and trip devices are often installed in adverse environments (e.g., hot, cold, humid, dusty, and marine). They also have to contend with being installed in hazardous area locations and can be subject to corrosive atmospheres from minor plant releases.

Safety instrumented system (SIS) logic solvers are typically installed in cabinets within climate-controlled equipment rooms fed from regulated and filtered power supplies. The final design ends up being fairly standard (e.g., 4–20 mA signals wired in and on/off 24 VDC signals wired out). Modern SIS logic solvers assessed to IEC 61508 [1] and installed/maintained in accordance with the safety manual are typically trouble free.

In contrast, the installation and specification of field devices can vary significantly according to the different process applications. In the authors' experience, it isn't possible to adequately cover all of these requirements in the field device safety manual, nor is it appropriate to always apply "cookbook" solutions. What seems like a standard selection isn't always the case on closer inspection. All too often, field devices adversely affect plant performance and, at worst, severely compromise safety. It isn't good enough for the safety functions to just operate on demand. Because start-ups and shutdowns can be the most dangerous periods for a plant, spurious trips should be minimized. Poorly operating safety functions also run the risk of being "unofficially" bypassed.

Selecting the most appropriate field devices requires an understanding of the process, piping and vessel design, metallurgy, and more. This chapter will explore the requirements for properly selecting field devices and reducing systematic faults. The aim is to provide workable designs that help the process operate smoothly and not just meet a risk reduction target. Selecting IEC 61508 [1] assessed field instruments isn't of much use if the measurement and process shutoff techniques don't match the application.

Timing of Field Device Specification

As discussed in other chapters, timing is everything. Field devices are connected to process equipment and piping. This means that late changes to instrument and final element devices can have serious implications for the plant layout design. It isn't like changing some wiring in a logic solver cabinet or making some program updates. In many instances, there will be resistance from the project for late changes to instrumentation and this may result in sub-optimal "fixes." Therefore, it's important for functional safety engineers to analyze the safety instrumented function (SIF) requirements early enough to fit in with the plant and piping layout design.

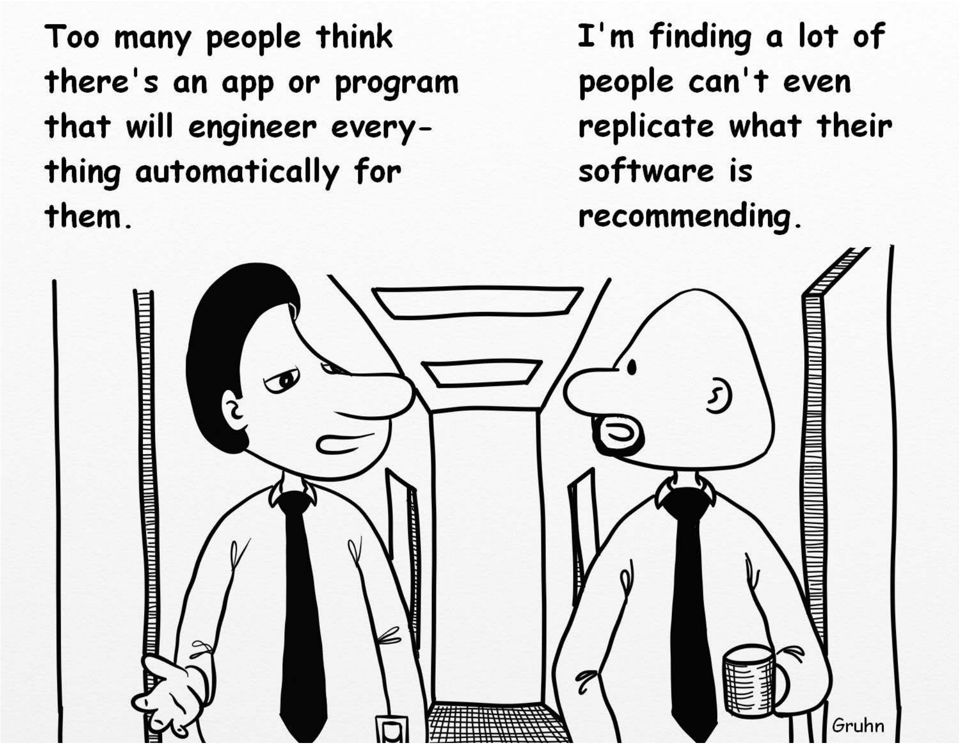
For example, a level instrument needs to be matched to the vessel design. It's important to consider the modern scale of process operations. One of the authors has been involved with column designs that are 250 feet high and 26 feet in diameter. If the redundancy needs to be increased from 1oo1 to 2oo3 due to the safety integrity level (SIL) verification requirements, additional connections are normally required on the vessel. Vessel designs can't be readily changed past a certain project milestone, and it may be necessary to use common process connections via an external level bridle/chamber. This approach may cause difficulties with the common cause error assessment in a plugging/dirty process application.

Another example may involve the speed of response specification for an on/off valve. Placement of large valve actuators can often present problems for the piping and platform access design. Some actuator types are more compact (e.g., piston). The operating speed for piston operated rotary valves can be limited by the torque limits on the actuator stem. In some cases, linearly operating globe valves are needed. However, globe valve actuators can be larger. This extra space, due to the process safety time requirements, needs to be considered at the right stage of the design.

Reliability and Systematic Capability

Previous chapters have reviewed reliability data, systematic errors, common cause faults, and initial system evaluation. IEC 61508 [1] and ISA/IEC 61511 [2, 3] also provide guidance about evaluating systematic capability. A key objective of the SIL verification activity is to confirm that a target risk reduction factor (RRF) is achieved. Readers may be familiar with SIL certificates provided by many vendors. A discussion about the validity, or the lack thereof, of the reliability data contained in these certificates could fill an entire

11
Engineering a System


Too many people think there's an app or program that will engineer everything automatically for them.
I'm finding a lot of people can't even replicate what their software is recommending.
Gruhn

"Successful engineering is all about understanding how things break or fail." ~ Henry Petroski

We Have a Safety Requirements Specification, What's Next?

The safety requirements specification (SRS) is the key document used to define the safety system [1, 2]. However, the SRS isn't meant to provide the detailed engineering and design requirements of the SIS. This chapter will consider how to tie together the fundamental hardware elements of a safety function (i.e., field measurements, logic system, and field final elements).

The goal of many engineers and companies is to develop a "cookbook" for system design. The more senior engineers, with their experience and knowledge, will document rules, standard design templates, and work instructions to be followed by the more junior engineers. One must be cautious, however, for one set of rules or procedures simply can't apply to all systems and organizations.

Some aspects of system design are impossible to quantify in a reliability model. However, they can have a profound impact on system performance and integrity (e.g., for minimizing systematic faults). This chapter will explore the principles behind the "cookbook." The aim is to help readers adapt these guidelines for their applications.

The physical design of an SIS may appear to be the same as the basic process control system (BPCS). Field instrumentation, cabling, wiring, and the cabinets housing the "smarts" look the same. Also, the SIS is often installed in the same equipment room as the BPCS. However, safety system performance requirements lead to some different designs than those of basic control functions (e.g., to facilitate proof testing, improved detection of latent faults, and additional redundancy).

The passing of information between the SIS and other systems, such as the BPCS, is an important aspect of the design. This interface may be implemented with hardwired signals between system inputs and outputs (I/Os), serial communications, or a "native" network connection. Deciding on the most appropriate configuration should be considered early in the design phase. Late changes frequently increase the risk of introducing faults.

Project Schedule versus Out of Sequence Design

In the authors' experience, most problems arise from late design decisions and changes (i.e., out of sequence design) rather than from fundamental technical difficulties. The SIS is often interfaced with many other systems, which may be on a different/early procurement cycle (e.g., mechanical packages). The safety system logic solver is also connected to the field instrumentation in a variety of configurations. This chapter will consider the many I/O connection design decisions that can affect the setup of the safety system. Therefore, the common theme in this chapter will be about establishing the right design criteria at the appropriate time on the project schedule.

Architecture Drawing

Although there are exceptions, most safety functions in the process industry are fairly straightforward. In a typical safety function, when a measurement goes beyond a high or low limit, something needs to close, open, or stop. With a modern SIS, this just means wiring up a field instrument to an input module and connecting the output module to the final element, as shown in Figure 11-1.

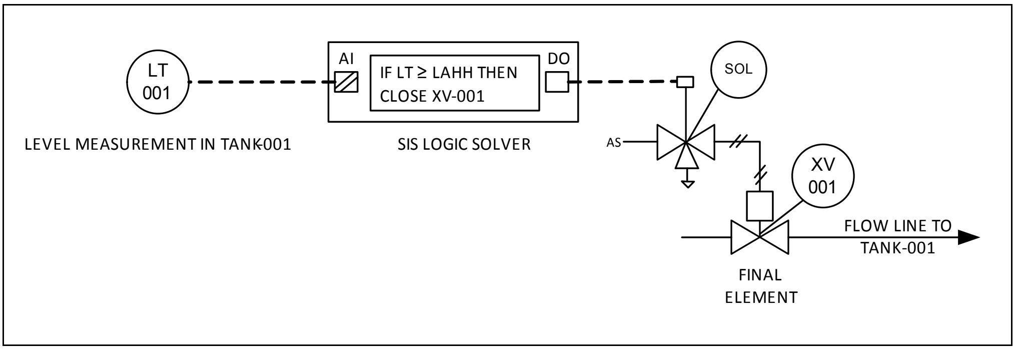
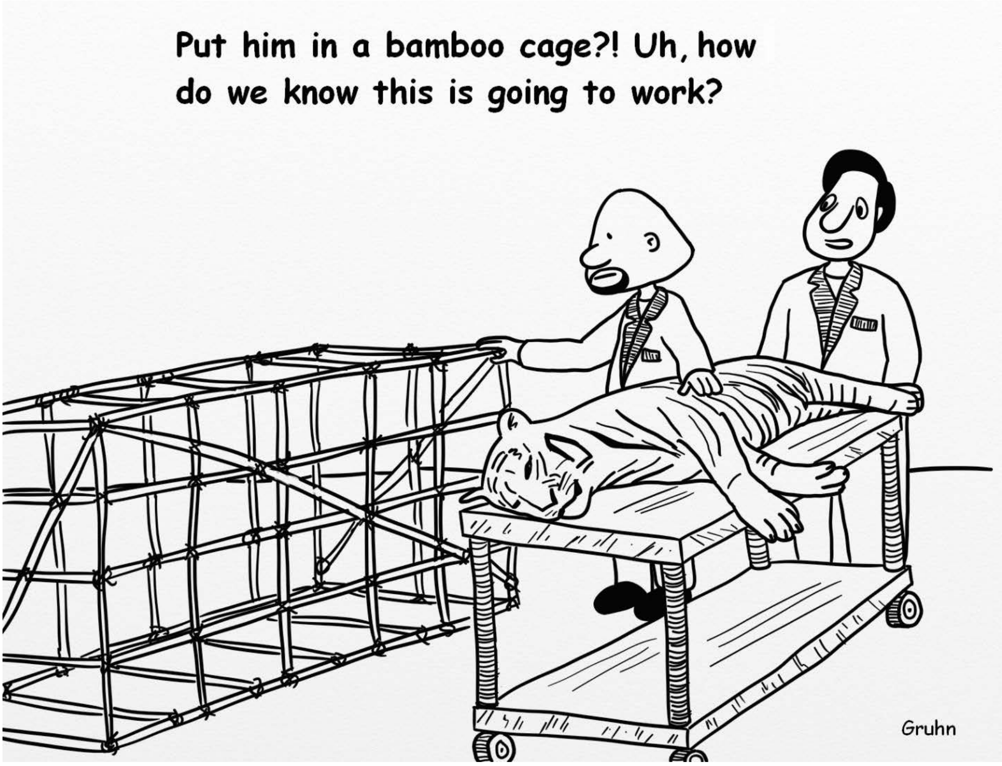


Figure 11-1. Safety Instrumented Function Example

However, there are often complications in how the SIS may need to be interfaced to other systems, such as:

- Basic process control system (BPCS)
- Machinery monitoring system (MMS) (i.e., vibration protection)
- Motor control center (MCC)
- Fire and gas system (FGS)
- Emergency shutdown panel (ESD)
- Heating ventilation and air conditioning system (HVAC)
- Other safety systems included on mechanical packages, such as fired heaters and compressors

12
Software


Put him in a bamboo cage?! Uh, how do we know this is going to work?
Gruhn

"There are two methods in software design. One is to make the program so simple, there are obviously no errors. The other is to make it so complicated, there are no obvious errors." ~ Tony Hoare

We Have Set Up All the Hardware, the Rest Is Just Software!

At its most basic level, a programmable safety instrumented system (SIS) consists of input modules, a logic solver, and output modules. The logic solver executes a program that compares field measurements to trip settings and then controls the final elements (e.g., turn on or off). This chapter will review the steps to translate the safety requirements specification (SRS) functional requirements into a program, or application software.

In the past, relay panels were often the only means to implement shutdown logic. The wiring between the relays was the equivalent of today's SIS programming. In fact, the term *ladder logic programming language* derives from the original relay implementation. Spare capacity was a crucial factor in the design of the relay panel. Additional logic meant more relays and therefore extra real estate. With the advent of programmable systems, the "apparent" pressure to completely determine the shutdown logic at an early stage decreased. It's far easier to have spare capacity in the logic solver and not affect the real estate. Unfortunately, this capability has sometimes led designers to downgrade the significance of the software development schedule. For instance, the system may be ordered with just an input/output (I/O) list but the application software's development is left to much later in the project.

Developing quality application software is important as it can affect the system's overall performance. The specification, design and design review, installation, and testing of application software isn't always given the priority and emphasis required. This can result in improper operation, project delays, and increased costs.

The application program isn't the only software that should be considered. The SIS operating system (OS) determines how the application software is executed. It's important to understand that the program execution cycle can affect the application software behavior in strange ways. Software-based SISs have simplified some of the design processes, but there are many other complexities (e.g., more sophisticated trip sequences, operational trip bypasses based on plant operating mode, and much more). These requirements need to be recognized and assessed early in the project cycle.

Sometimes engineering can get too wrapped up with overly clever software designs. However, the need for an operable system is paramount. The software should enhance an operator's ability to control the process plant and not place undue impediments in his/her way. Often, a simple discussion with the team about how the plant needs to be operated can prevent many problems.

A Systematic Approach to Software Development

Software doesn't degrade and fail the way mechanical components do. Software is either designed correctly, or it's not. System engineering considers safety failures as an emergent property [1]. It's usually some unforeseen combination of events that makes software-based systems "fail" (e.g., the software specification didn't consider that an operator would try to restart a pump before the shutdown sequence completed).

Many have tried to quantify the software's performance. This has developed into a highly contentious issue. There are dozens of techniques that yield very different answers. Thus, the safety system standards committees have abandoned the idea of quantifying the software's performance. Instead, users are urged to follow a methodical life cycle detailing requirements, development, and testing. Good specification and development practices should lead to good software development. Quantifying what many may realize are poor practices does little to improve anything.

As mentioned in Chapter 6, the cost of rectifying errors due to the software requirements specification is much more significant than for coding mistakes. Furthermore, errors of omission in this specification can be the most problematic. Chapter 11 emphasized the need to identify all the required interfaces to the SIS at an early stage of the project. This has a great bearing on the robustness of the application software structure. Last-minute, hasty changes to the application software can often introduce hidden errors.

Testing for these "hidden" errors can be difficult. One of the authors recalls many tumultuous design reviews when process computers were being introduced into the industry. One project objective was to automate many local operator functions and to move the operators away from direct local supervision next to reactors (i.e., it was a very hazardous job). The question arose about how to completely test the software considering all the input combinations (e.g., 50 digital inputs [DIs] equates to 2^{50} possible input combinations). It quickly became apparent that a systematic software design approach was required to make the testing manageable and more focused.

Software Life Cycle

Many techniques, models, and methods have been used to develop application software. These techniques normally follow a particular sequence, or set of steps, to ensure success. The intent is to get it right the first time, rather than being forced to do it over.

13

System Testing

The boss thinks a FAT is optional. He just wants to simulate everything in software. Something tells me this project is not going to turn out well.
Gruhn

"The best laid plans of mice and men often go astray."
~ Adapted from Robert Burns' "To a Mouse"

It Wasn't Supposed to Work That Way?!

Under normal circumstances, the safety requirements specification (SRS) is the starting point for the design of the safety instrumented system (SIS). This leads to the preparation of field device data sheets, the SIS logic solver specification, various procurement packages, the programming scope, and the installation package. Once the engineering and design activities have been completed, the various components of the SIS are installed and then readied for commissioning. At this point, the system should have gone through a number of test cycles to verify that the actual SIS behavior matches the design intent.

This chapter covers how to test the design and installation up to the handover to the plant operations and maintenance groups. Testing is a complex series of quality control (QC) activities. All too often, its planning is undertaken as an afterthought when the design has been completed. A proper testing plan/framework should be laid out at the early stages of the safety life cycle, once the SRS has been finalized. Errors and omissions can then be corrected during design development with a more effective process. The earlier faults are discovered, the more options we have to effectively correct the design and/or installation. There's an increased risk of "Band-Aid fixes" being applied to the SIS when errors are discovered later (e.g., at commissioning).

In many process plants the SIS can be large, complex, and distributed over a number of units. Large mechanical packages, such as compressors, are often installed in modules or skid assemblies with their own SIS. Furthermore, there may be various groups responsible for different sections of the SIS. The complete SIS may only come together for the first time at site. This creates a challenge for the validation of the SIS. As the various sections and components are designed, fabricated, and installed, the test plan needs to consider all the interfaces.

An important part of the testing plan is aligning the design intent (i.e., the SRS) with expectations of the various groups (e.g., operations and maintenance). The test plan should be integrated with the design reviews, which are performed throughout the engineering cycle. People are better able to give an informed approval of the design when shown how a system works during an acceptance test. This avoids unpleasant surprises at the prestart-up checks when operations decide that the system is not meeting their requirements.

Testing Philosophy and Concepts

The need for testing throughout the whole safety life cycle is clearly stated in ISA/IEC 61511 [1, 2]. However, a discussion about verifying control and safety systems can result in disagreements about how long testing should take, and even what it actually achieves. Many project execution folks view this as something that just takes too much time. In the authors' experience, functional safety engineers understand the critical value of testing, but may not be able to properly communicate this need as objectives that are relevant to project management. Therefore, it's helpful to recognize that testing an SIS involves three distinct activities:

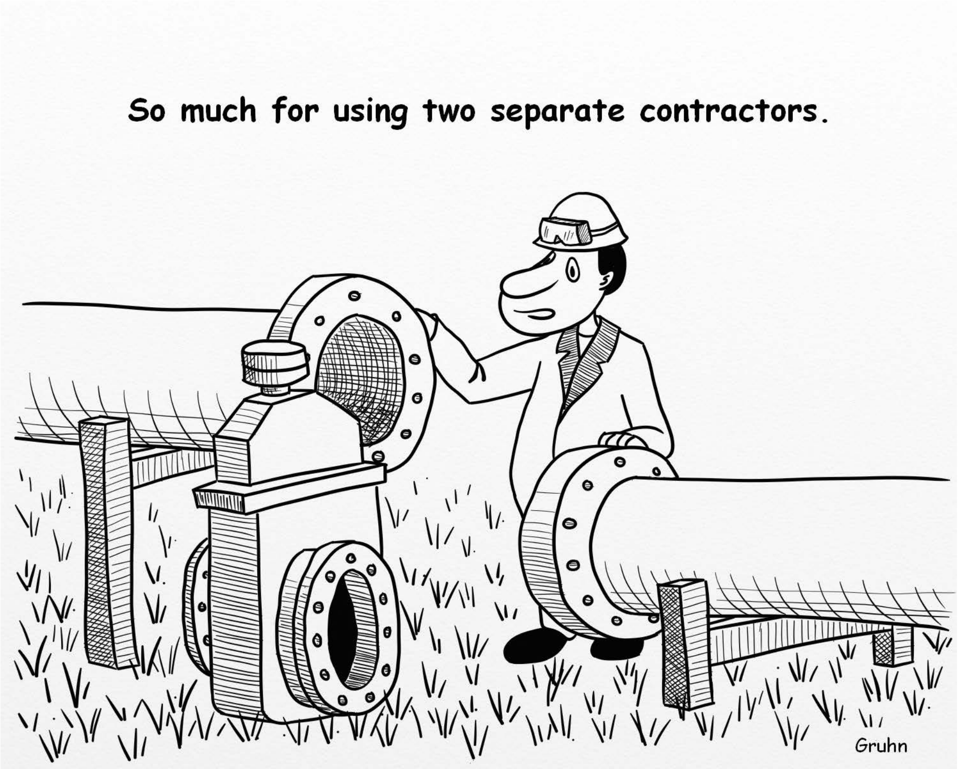
- Checking that the system behaves as per the SRS
- Diagnosing faults, errors, and unexpected behaviors
- Rectifying nonconformance

What may be seen from the outside as an overly long testing program may actually point to a system with too many faults. Testing a virtually error-free system doesn't consume the schedule and cause delays. In most cases, errors and omissions in the SIS, together with unexpected behaviors, are the main sources of concern.

An important reason for a robust testing plan is to discover and rectify faults well before getting to commissioning. Non-conformances in the SIS become increasingly problematic as the project moves into the module fabrication, installation, commissioning, and start-up phases. The costs and schedule delays rise exponentially since now there is a need to make physical, rather than documentation, changes. There are also fewer engineering resources to come up with solutions. The bulk of the design effort has been wound up by this stage of the project. Furthermore, it's a lot harder to make significant changes to the overall system architecture when the SIS has been already installed. This increases the risk of suboptimal "fixes." Some examples of unexpected behaviors are:

- Logic solver overloaded due to input/output (I/O) added after the integrated acceptance testing
- Ineffective grounding design
- New maintenance override strategy requiring additional switches wired from the operator console to the SIS

14
Installing a System



"A desk is a dangerous place from which to view the world." ~ John Le Caré

The safety requirements specification (SRS), as described in ISA/IEC 61511 [1, 2], is an important basis for the selection, engineering, and design of safety instrumented functions (SIFs). Once the various components are specified, procured, fabricated, and assembled, this chapter considers the next step, which is installing field devices, SIS logic solvers, I/O cabinets, power supplies, cabling, wiring, process connections, and more.

An SIS must contend with the fabrication and construction sequence together with the problems associated with transporting the system to the site. The final "home" for the SIS logic solver is usually multiple cabinets within a clean, climate-controlled equipment room fed from well-regulated and filtered power supplies. However, the "journey" to get there can be difficult (e.g., there are other construction activities in the equipment room while the SIS cabinets are being installed). The SIS logic solver also needs to be protected from electromagnetic interference (EMI), both direct and via the wiring from the field instrumentation to ensure reliable measurements and control of final elements.

Software has taken over many functions previously implemented in hardware. It's often treated as a "black box" and independent of the installation. However, software installation has become more complex with the increasing modularization of plants. The application software may not be available as one completely tested package. Various parts may be required for different plant process units at different times.

Knowing *when* to install the field instrumentation in the construction schedule can often be more important than *how*. The piping and structural steel fabrication environment isn't for the fainthearted. Field devices are installed in a variety of configurations to match different process conditions. Incorrectly installing measurement devices can adversely affect the accuracy and response time of the measurement. The performance of trip valves needs to also consider how they are installed in the pipe work. In the authors' experience, it isn't possible to adequately cover all these applications in the safety manual. All too often poorly installed field devices adversely affect plant performance and at worse, severely compromise safety. Also, it isn't good enough for the safety functions to operate on demand. Spurious trips should be minimized; a reliable plant is a safe plant. Poorly operating safety functions run the risk of being "unofficially" bypassed.

Selecting the most appropriate field device installation detail requires an understanding of the process, piping and vessel design, metallurgy, electrical hazardous area design requirements, and more. This chapter will explore the requirements for properly installing field devices and connecting them to the SIS logic solver. The aim is to provide workable designs that help the process operate smoothly and not just meet a risk reduction target. A properly installed safety system will greatly reduce the risk of introducing systematic faults.

The Installation and a Bit of Philosophy

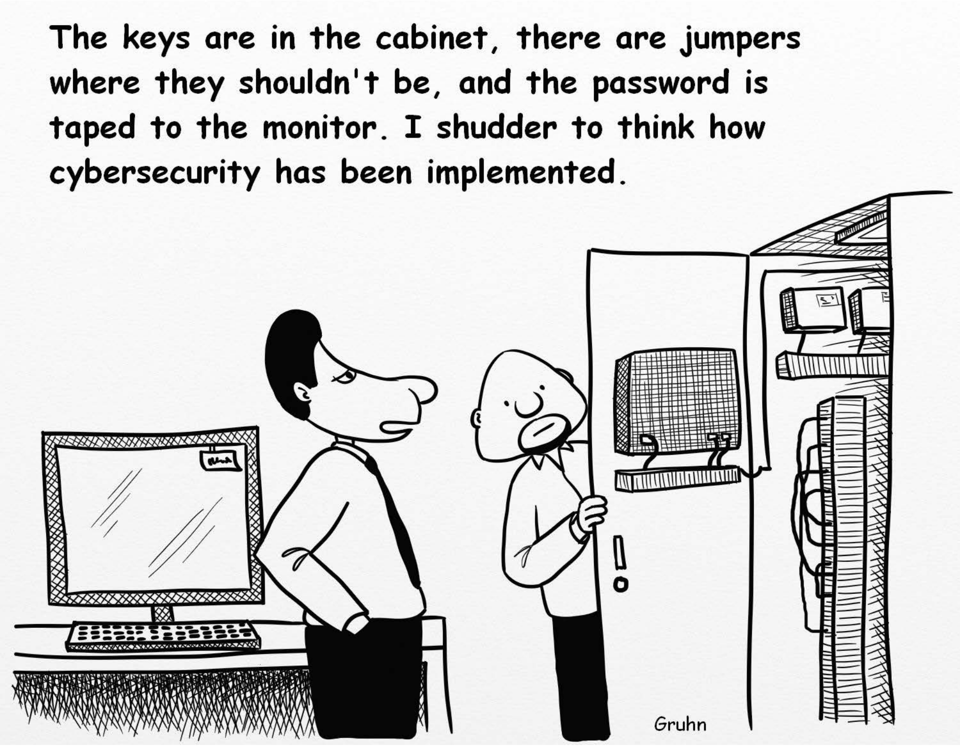
It's important for functional safety engineers to be aware of the steps involved in designing a multidisciplinary installation package for a typical process plant. The piping designers, with support from the process engineers, lay out the main equipment on a plot plan (e.g., vessels, reactors, tanks, sumps, compressors, pumps, and much more). The plot plan eventually leads to a three-dimensional (3-D) computer model, which also details all the interconnecting pipe work, structures, instrumentation, and electrical equipment. The process engineers undertake many studies that determine the equipment and piping layout (hydraulic, relief and blow down, etc.). The final design is a result of a very complex and time-consuming series of tasks.

The installation can affect the reliability of the safety function. All too often the instrument installation isn't considered early enough to get the best design. The 3-D model is often already at an advanced stage when the instrumentation folk require some changes. The following sections provide some suggested installation designs, which improve instrument reliability. However, these designs do affect piping and equipment layout. It's important that instrumentation installation requirements are identified at the right time. Trying to make major changes to a 36-inch piping layout for flowmeter straight-run requirements late in the project is going to be a difficult task. However, in the authors' experience, it isn't that difficult to get the right design if the piping group knows about instrument requirements earlier in the project.

A typical design process for a safety system involves functional safety engineering, instrumentation engineering and design, and automation engineering (i.e., for the basic process control system [BPCS] and SIS logic solver). The output of the design is an installation package that is provided to a construction group. Usually, functional safety engineers don't directly undertake the detailed design and rely on others to complete the installation package. They provide the SRS to the instrumentation and automation groups and they get involved in testing (e.g., factory acceptance test [FAT]).

15

Cybersecurity


The keys are in the cabinet, there are jumpers where they shouldn't be, and the password is taped to the monitor. I shudder to think how cybersecurity has been implemented.
Gruhn

Industrial control system cybersecurity is a growing concern. Power grids have been brought down, and process facilities have been damaged through cyber events. The Presidential Executive Order 13636, "Improving Critical Infrastructure Cybersecurity," released in 2013, acknowledges the importance of this issue.

The focus of this book is safety instrumented systems (SISs), but the latest generation of SIS standards state that cybersecurity must also be considered and evaluated within these systems. The International Society of Automation (ISA) and the International Electrotechnical Commission (IEC) are developing a family of 13 documents (standards and technical reports) on cybersecurity. About half of the documents had been published as of early 2018. The National Institute of Standards and Technology (NIST) have written guidelines on this subject [6]. ISA and others have written books on this subject. The most recent edition of the ISA-84 technical report on cybersecurity is more than 100 pages long [1]. ISA and others have training classes and qualification programs on this topic. The four separate courses from ISA (focusing on different portions of the cybersecurity life cycle) total *11 days* of classroom training. Therefore, this is obviously *not* a simple or trivial topic. A single chapter in a book like this cannot hope to cover all the details necessary to qualify anyone to do such work. All this chapter can do is summarize basic concepts and steps to be followed. Practitioners will need to refer to the references for the remaining details.

Similarities and Differences Between Functional Safety and Cybersecurity

The cyber concerns of informational technology (IT) systems tend to be different than for operational technology (OT) systems. The order of concerns for most IT applications is confidentiality, integrity, and availability (CIA). The order of concerns for most OT applications is just the reverse: availability, integrity, and confidentiality (AIC). Occasional outages of an IT system can be tolerated (and are often scheduled). The same cannot be said of a process control system. The technology life cycle of most IT components is less than 5 years, but often control systems must operate for 20 years or more. Antivirus software is common in IT applications, yet many control systems simply do not have the communication bandwidth or computing power to even support it. IT practitioners tend to be very aware of cyber issues; the same cannot be said for the majority of control professionals.

There are similarities between the functional safety standards and their life-cycle approach and the cybersecurity standards and their life-cycle approach; yet there are differences as well. For example, target safety integrity levels can be quantified, and the performance of hardware can be quantitatively evaluated. The determination and evaluation of security levels is purely qualitative at this time.

Access control to documentation concerning functional safety and cybersecurity differs. Many different disciplines require access to process and functional safety documentation. Cybersecurity documentation consists of zone and conduit drawings and other documents related to countermeasure design, verification, and validation activities. Such documents would provide obvious and significant assistance to any potential malicious attacker. The distribution and access to such confidential information should therefore be controlled.

Open Systems Are Vulnerable

Control and safety instrumented systems are usually microprocessor-based systems programmed using computers running Windows operating systems. They are also monitored and operated using computers running Windows operating systems. Control and safety systems often reside on the same process control network. That process control network often has other computers and monitors connected to it and will typically interface with the enterprise network. The enterprise network will often interface with the Internet.

What if a malicious actor or malicious code were able to enter and compromise the control system? This could result in a loss of both control and alarm layers. Values in both layers could be manipulated if they were to reside in the same system. An even worse scenario would be if the malicious code were also to compromise the safety instrumented system. In this scenario, an attack could result in the loss of three layers of protection based on a single initiating event or attack. Such attacks *have* happened.

Process safety standards now require cyber vulnerability and risk assessments. The second edition of IEC 61511-1 was released in the summer of 2016, and was adopted by ISA in 2018. In clause 8 on process hazard and risk assessment, clause 8.2.4 states, "A security risk assessment shall be carried out to identify the security vulnerabilities of the SIS." There are six bullet points outlining the documentation and assessments that need to be carried out. In clause 11 on SIS design and engineering, clause 11.2.12 states, "The design of the SIS shall be such that it provides the necessary resilience against the identified security risks." That's as far as the new standard goes, but it does provide further guidance by pointing readers to an ISA-84 technical report [1] and the ISA/IEC 62443-2-1:2010 standard [4], which covers *how* to perform cyber vulnerability and risk assessments.

Basic Concepts of ISA/IEC 62443 Standards

The ISA/IEC 62443 series [4] consists of 13 documents, mainly standards and some technical reports [4]. These documents are grouped into four categories, as follows.

- **General** – Concepts and models, glossary of terms and abbreviations, security conformance metrics, and security life cycle and use cases
- **Policies and procedures** – Management system, implementation guidance for the management system, patch management, and requirements for solution suppliers
- **System** – Security technologies, security risk assessment and system design, system security requirements, and security levels
- **Component** – Product development requirements, technical security requirements for components

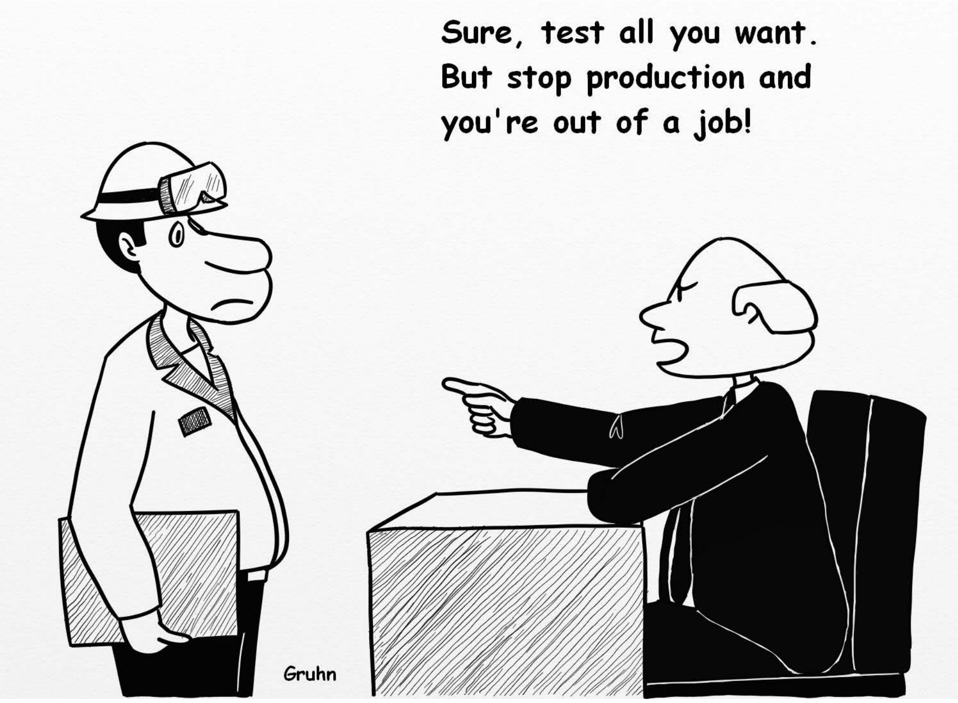
There are several **fundamental concepts** with the cyber standards: security life cycles, zones and conduits, and security levels.

Security life cycles are composed of three smaller, interconnected life cycles, each devoted to a specific phase and focused on a specific class of stakeholders: product development, integration and commissioning, and operation and maintenance.

A large, interconnected system can be decomposed into a series of **zones**, connected by **conduits**, with each zone having a specific set of security requirements based on a detailed risk assessment. Zones and conduits can be likened to an airport. There are different zones in an airport. The ticketing and baggage claim area are zones open to anyone. Inside the airport is a zone only accessible to those with boarding passes and identification. The conduit between the two is the security checkpoint. Outside on the tarmac is a zone only accessible to airport employees. The conduit between the two areas is generally a locked door that requires two-factor authentication (e.g., a badge and a password/PIN). Many process facilities have a similar set of zones (e.g., business, process control, and safety) with conduits connecting them. It should not be possible to corrupt information in the safety zone from outside.

16

Operations and Maintenance


Sure, test all you want. But stop production and you're out of a job!
Gruhn

"Constant attention by a good nurse may be just as important as a major operation by a surgeon." ~ Dag Hammarskjold

Pressing the Start Button

This chapter covers the operating phase of the safety instrumented system (SIS), beginning with commissioning. Start-up and commissioning is the transition period where the SIS is validated for long-term operation. Field instrumentation is connected to a "live" process for the first time. Therefore, there are important design assumptions that should be confirmed before the system can be declared fully operational. Just pressing the start button and hoping that the SIS behavior matches the design intent isn't a recommended approach. Furthermore, the setup of the SIS sometimes needs to be temporarily adjusted during this phase. The plant won't be operating under normal design conditions until sometime after start-up.

As described in previous chapters, much of the engineering effort prior to mechanical completion is concerned with installing the system. The resulting documentation may not be best suited for the operations and maintenance groups. It's helpful to have these groups review the engineering package (e.g., the safety requirements specification [SRS]) before the design has been completed. This should result in a better understanding of the design intent (e.g., the rationale for safety function bypass limitations). On the other hand, operations and maintenance personnel can provide invaluable insights on how equipment works in practice (e.g., the manual functions required to start up a large compressor).

It's important that appropriate proof test documentation is in place by the time the plant is brought online. Checking the operation of the SIS on a running process is a different proposition than an offline factory acceptance test (FAT). Typically, commissioning groups would work out start-up plans, including detailed test procedures. However, even commissioning procedures may not be entirely suitable for ongoing maintenance. The level of engineering resources available during the design and commissioning phases is much greater than is typically present for the longer operating phase of the SIS. These operational aspects can sometimes be overlooked during the project phase, and the maintenance group may struggle to come up with an effective test program in the available time.

Training is crucial for ensuring that the SIS can be effectively operated and maintained. A competency program should already be in place once the system has been handed over by engineering. Setting up a competency and appointment program isn't a trivial exercise. It can cover activities ranging from system performance monitoring and program modifications to field instrument repair.

Without an ongoing audit process, it's difficult to ensure that the SIS will continue to operate over the long term according to the design intent. The life of a process plant can be in excess of 30 years. Equipment will wear out and need to be replaced. The design intent may be continually evolving over the life of the plant. The process itself may be optimized and units may even be decommissioned. What was working properly yesterday may not be effective tomorrow. Also, changes in personnel and loss of experience are inevitable. The SIS management systems should cover these aspects.

Note: Operating facilities can be organized in a variety of ways. In this chapter, the terms operations *and* maintenance *are used in a generic manner. It's expected that each group will have appropriate engineering resources. However, in many end-user organizations, engineering and management functions may reside in different groups (e.g., technical support). Sometimes, maintenance and operations functions are undertaken by one multidisciplinary group. Therefore, the reader should validate how the various activities and responsibilities described below are best executed for his/her organization.*

Documentation for the Operations versus the Project Phase

The scope, size, and setup of a project to install an SIS can vary considerably. The design and commissioning of the system can be part of a large lump sum turnkey contract. A running plant, complete with trained operations and maintenance personnel, is handed over to the end user. Other large-scale contracts may finish at mechanical completion, as described in Chapter 13, where the end-user team has to complete the site acceptance testing (SAT). This team would then plan and execute the start-up and commissioning phases. The team would also establish the operating and maintenance procedures. Many smaller SIS projects may be implemented by the end-user engineering group or even by the maintenance group. The scope may be a modification or addition to an existing SIS on a running plant.

The timing of when the operations and maintenance groups are engaged in the project will also vary. The plant may be a completely new facility where the workforce may not be fully onboard until late in the project. At the start of the project, the end-user staff are often from the capital engineering group and may not be running the plant when commissioned.

17 Management of Change

Back track and check all you want,
BUT GET THAT UNIT RUNNING NOW!!
Gruhn

"It's a bad plan that admits of no modification." ~ Publilius Syrus

When and Why Did We Change That?!

Building, starting up, and running process plants doesn't usually follow a straight line from concept to full on-spec production. There are many twists and turns that eventually lead to a design, which may also be modified during testing. The construction group may then have to "tweak" the design—what seems workable on paper may not be readily constructible. Even after the plant is built and running, change doesn't suddenly go away. The plant may not perform exactly as anticipated, or feedstock specifications may have changed. As the operations group gains more experience, they will identify debottlenecking opportunities to improve safety, efficiency, and profitability. This often entails changes to main equipment and instrumentation. Part of these improvements may also require updates to operational and maintenance procedures. Proof testing and maintenance are crucial for optimal plant performance. The maintenance group may have to contend with some poorly performing devices or impractical testing requirements. They will also want to improve (i.e., change) the design.

The shutdown and demolition of an entire plant is most likely an intensive but well-executed exercise. However, the decommissioning of individual pieces of equipment or safety functions does not always get the attention it needs. This change activity can be more complex than people expect, therefore, it gets its own "mention."

Safety and environmental directives can often come from the head office or regulators. These may involve unexpected changes for the plant operations team. However, it is important to properly manage change regardless of the source, even if it is a regulatory requirement. IEC 61511 [1] recognizes the need to manage change throughout the safety life cycle. This chapter will focus on what factors make for an effective management of change (MOC) process, with due emphasis on the safety instrumented system (SIS).

Managing Changes

Trevor Kletz has written many papers and books on what can go wrong in the process industry. Any student of process safety should read at least one of his books [2]. Trevor Kletz provides many "what went wrong" case studies and provides in-depth analysis of these failures. In addition to much other valuable advice, he stresses the need for a properly functioning modification procedure (i.e., management of change). This forms the foundation for reviewing the proposed changes against the original design basis/intent. Most importantly, MOC facilitates scrutiny by other personnel outside the design team. It is always better to have a fresh set of eyes to uncover missed faults.

At this stage, one may be tempted to make a case for management of change by citing an example from Trevor Kletz's works. However, the authors believe it's best to read through the original works and absorb the wisdom therein. Furthermore, the authors contend that most people do recognize that implementing MOC is the proper course of action. The problem is knowing how to best implement a proper MOC. It's important to understand what is and isn't helpful for setting up this process. Having a well-documented and well-understood design intent (i.e., a basis of design) is critical to ensuring the MOC process is effective. The following sections will explore what is meant by "well documented."

IEC 61511 Modification Safety Life-Cycle Approach

Clause 17 of the standard deals with the proper implementation of modifications to the SIS. The key objectives, as stated in the standard, are:

- To ensure that modifications to any SIS are properly planned, reviewed, approved, and documented prior to making the change
- To ensure that the required safety integrity of the SIS is maintained despite any changes made to the SIS

In meeting these objectives, it's important to determine if there is an impact to functional safety as a consequence of the proposed modification. This analysis needs to consider changes in other layers of protection, hazard-initiating causes, and operational procedures that can potentially affect the safety functions. These may include:

- Control functions increasing the demand frequency on the protection layers, including relief valves
- Process equipment such as pumps creating different pressure and flow regimes
- Operating conditions affecting factors, such as process safety time and reaction profiles
- Increased relief loading on flare systems
- Manning levels in certain areas
- Operator actions in response to alarms and other cues

18

SIS Design Checklist

Introduction

The use of a checklist will not, in and of itself, lead to safer systems, just as performing a process hazards analysis and not following any of the recommendation will not lead to a safer facility. A checklist that merely consists of a clause-by-clause listing from one particular standard would not really be adding to the body of knowledge. The following checklist is an attempt to list as many procedures and practices as possible (from standards, books, and accumulated knowledge, much of it learned "the hard way") in the hope that, by following a systematic review of the overall design process, nothing will fall through the inevitable cracks within and between organizations and be neglected.

This checklist is composed of various sections, each corresponding to different portions of the safety life cycle as described in various standards. The numbering of the checklist sections does not correspond to the numbering of any of the standards. Different sections of the checklist are intended for different groups involved with the overall system design, ranging from the user, engineering firm, vendor, system integrator, and consultant. Exactly who has what responsibility may vary from project to project and application to application. The checklist, therefore, does not dictate who has what responsibilities—it only summarizes items in the various life-cycle steps.

The checklist might not be used in the same way for every circumstance. For example, a project may involve building a new facility, adding a few safety functions to an existing system, or maintaining an established system. The "Management Requirements" checklist would be used differently for a new project versus for maintaining an existing facility.

The following checklist should in no way be considered final or complete. As you review and use it, you are encouraged to add to it (for the benefit of others who may use it in the future). In fact, each section starts with a new page, leaving ample space for your additions and suggestions.

The safety life cycle is not always just a once-through process. It is important to review these checklists at the appropriate phase of the life cycle. As an example, the safety integrity level (SIL) verification calculation (i.e., for PFD_{avg}) should be done while there is time to make any required adjustments to the design (i.e., during the design cycle). Finding out for the first time that the safety function does not meet the integrity requirements during precommissioning is going to severely limit your options to remedy the design.

Often, it is very beneficial to have review meetings, using these checklists, to make sure that people are aware of the scope, responsibilities, and progress. The exact number, timing, and scope of these review meetings will vary considerably depending on the project, plant modification, or maintenance activity. Only some of the checklists below may be relevant to the review and/or there may be additional elements that need to be considered. The important objective is that there is some sort of review process being undertaken.

This checklist covers the following steps:

1. Management requirements
2. Process hazards analysis
3. Safety requirements specification (SRS)
4. Conceptual SIS design
5. Detailed SIS design
6. Instrument air
7. Power and grounding
8. Field devices
9. Operator interface
10. Maintenance/engineering interface
11. Communications
12. Hardware specification
13. Hardware manufacture
14. Embedded (vendor) software
15. Software coding/programming
16. Factory acceptance test (FAT)
17. Installation and commissioning
18. Validation
19. Operations and maintenance
20. Testing

21. Management of change (MOC)
22. Decommissioning
23. Documentation

Why bother with a checklist at all? Please go back and look at Figure 2-1. The majority of the accidents involving control and safety systems were due to *incorrect and incomplete specifications* (functional and integrity). One can easily see that the majority of issues covered in the figure focus on *user* activities. Industry standards, as well as this checklist, attempt to cover *all* of the areas shown in Figure 2-1 and not just focus on any one particular area. After just glancing through all the detailed sections of this checklist, ask yourself whether your organization has done and verified all the items listed, and then decide whether this detailed list is really worthwhile.

Section 1: Management Requirements

Item #	Item	Circle a Choice			Comments
1.1	Has someone been formally appointed as accountable for the management of the SIS?	Y	N	N/A	
1.2	Have persons or departments responsible for carrying out the phases of the life cycle been identified?	Y	N	N/A	
1.3	Have persons or departments responsible for carrying out the phases of the life cycle been informed of their responsibilities?	Y	N	N/A	
1.4	Are persons competent to perform the tasks assigned to them?	Y	N	N/A	
1.5	Is personnel competency documented in terms of knowledge, experience, and training?	Y	N	N/A	
1.6	Is there a procedure in place to manage competence of all those involved in the SIS life cycle?	Y	N	N/A	
1.7	Has the functional safety management system of suppliers been checked?	Y	N	N/A	
1.8	Is a safety plan in place that defines the required activities?	Y	N	N/A	
1.9	Are procedures in place to ensure prompt and satisfactory resolution of recommendations?	Y	N	N/A	
1.10	Are procedures in place to audit compliance with the requirements?	Y	N	N/A	
1.11	Has a gap analysis been carried out to ensure compliance of existing SISs?	Y	N	N/A	
1.12	Has a plan been put in place to close the deficiencies of existing SISs identified by the gap analysis?	Y	N	N/A	

19

Case Study

That's an interesting and creative solution. Perhaps next time try following the instructions.
Gruhn

"If you wait until there is another case study in your industry, you will be too late!" ~ Seth Godin

What's in a Case Study?

This chapter will present a case study illustrating how to specify, design, install, commission, and maintain a safety instrumented system (SIS). International standard IEC 61511 [1, 2] provides guidance for implementing the safety system life-cycle phases. Armed with this knowledge, many may consider that safety systems engineers should be able to tackle any project. However, as discussed previously in this book, the design of an SIS doesn't always fit into a nice tidy package. It's important for the reader to understand that IEC 61511 isn't, and shouldn't be, a "cookbook" with a recipe for every occasion.

There is always a dilemma in setting up a case study. There is a need to strike a balance between providing realistic examples versus making sure the principles for implementing a safety system are clearly demonstrated. The safety system described below will only contain a few safety instrumented functions (SIFs) and won't in any way be a complete setup for a typical process plant. However, the chosen safety functions, based on level measurement, won't always have a "clean" solution. It's important to recognize what are the more important elements of a design. As an example, determining which factors affect systematic reliability has a greater bearing on safety than calculating the SIF PFD_{avg} to the third decimal place.

In this authors' experience, most functional safety engineers understand the critical importance of the safety requirements specification (SRS) for an SIS. As described in Chapter 6, key inputs to the SRS are the process hazard analysis (PHA)/ hazard and operability study (HAZOP), which are used to identify credible hazards, and the safety integrity level (SIL) allocation/determination review for the required SIFs. However, it's also important for functional safety engineers to be involved in other studies and design reviews at the right time to make a difference (e.g., piping and instrumentation diagram [P&ID] reviews, process desktop safety reviews). This case study will illustrate what other elements can influence the design of the safety system (e.g., the process control strategy and start-up/shutdown procedures).

Important note: It isn't possible for a case study to be the "answer" that can be applied to any installation. The reader should consider this case study as only illustrating the IEC 61511 safety life-cycle phases for training purposes. The aim of this chapter is to arm readers with the means to ask the right questions and to come up with their own approach that works for their safety system applications.

Furthermore, the various plant flowsheets, block diagrams, and designs are simplifications. They aren't intended to be actual working, or workable, plants. This author has significantly modified a blend of real-life examples in order to more clearly illustrate important aspects of the safety life cycle.

Making Sense of the Safety Life Cycle and Some Philosophy

Figure 2-6 from Chapter 2 summarizes the safety life cycle (SLC), as derived from IEC 61511. In the authors' experience, the most important phases are the hazard and risk analysis and the allocation of safety layers (i.e., phase 1 and 2, respectively). The normative part 1 of the standard doesn't provide detailed guidance about these phases. However, the informative part 3 of IEC 61511 [3] does refer to various Center for Chemical Process Safety (CCPS) and American Institute of Chemical Engineers (AIChE) documents, including hazard evaluation procedures [4] and layer of protection analysis [5]. Part 3 of the standard also provides additional information for several SIF allocation and SIL determination techniques.

These various reference documents provide invaluable information about hazard identification and the allocation of safety functions. However, they don't show how hazard identification should be integrated within project execution. As mentioned previously in this book, timing is everything. This case study will show how a safety function requirement can change from SIL 3 to SIL 1 by "simply" considering the design earlier in the project. Here, it's useful to reconsider some important design steps affecting functional safety from Chapter 17 to provide the context for this case study. These are shown in approximate chronological order below:

1. Process flow diagrams (PFDs) are issued for design during the middle of the front-end engineering design (FEED) phase.
2. Initial heat and material balance study is issued early in the FEED phase.
3. Relief and blow-down study is issued for information (IFI).
4. P&IDs are issued for review (IFR) for the P&ID review sessions.
5. P&IDs are issued for hazard analysis (IFH) following the P&ID reviews.
6. PHA/HAZOP is completed using IFH P&IDs.

Annex A

Things to Consider When Selecting an SIS Logic Solver

Just as each project is different, the most appropriate safety instrumented system (SIS) logic solver for each project may differ as well.

Vendors will often promote what they believe to be their unique differentiators and why they believe their system to be the best. Unfortunately, what a vendor perceives as an important feature may not always have an associated real benefit in the mind of the user. Every system can't be "best." Ford, Chevy, and Chrysler don't all tell people they make the "best" car. Besides, you don't need what's "best" (e.g., are you driving a Rolls-Royce?). All you really need is *what's suitable* for your application at *a price you're willing to pay, from someone you trust* (e.g., a used Ford sedan from a nearby dealer where a friend of yours is a salesman).

Most requests for quotations (RFQs) for safety logic solvers in the process industry call for a logic solver certified for use in SIL 3. While there may be a relatively small percentage of functions that require SIL 3 (including field devices, not just the logic solver), if you have just one SIL 3 function, that may be reason enough to specify an SIL 3–rated logic solver (as the logic solver is common to all the loops). Specifying an SIL 3–rated logic solver is often viewed as a conservative and safe choice, even if you don't have any SIL 3 requirements. However, over-specifying may mean overspending, so perhaps it may make sense to consider potentially lower-cost hardware that only meets SIL 2 requirements.

As with most things in life, when there are few choices available, selecting between them is relatively simple. However, when the number of available selections is large, and they all differ from each other in a myriad of ways, choosing between them can seem overwhelming. Rather than decide in such cases, some fall back on the old saying “nobody got fired for buying IBM.” In the process industry, this is often referred to as “the herd mentality” (you can’t be faulted for doing what the rest of the herd is doing). But what if in reality a group of cavemen are purposely driving the herd over a cliff? What if it’s really the lemming mentality? If your peers wear purple spiked hair, are you going to also?

If an SIL 3–rated logic solver is desired, there are approximately a dozen different manufacturers to choose from and four basic configurations (i.e., simplex, 1oo2D, triplicated, and quad). How can you choose between them all?

This annex offers a scoring system based on evaluating over three dozen different criteria. Each factor is initially scaled from -2 and +2. However, different criteria should have different weighting factors based on different application requirements. Certain criteria may *not* be important (size may not matter for a land-based installation, fault tolerance may not be important for a batch operation, etc.) and should be rated as 0 so as not to falsely skew the overall rating based on something that doesn’t really matter. Certain criteria may be *very* important (fault tolerance when uptime is critical, small size for an offshore application, very high speed for turbomachinery, good local support in a remote area, etc.) and could have a multiplication or weighting factor applied (e.g., a rating scale of -4 to +4, or -6 to +6 to show their greater significance). Numbers can then be totaled for an overall score, as well as viewed in a table providing an easy visual ranking (i.e., system X has more positive scores than system Y).

Company Issues

Knowledgeable Staff

Companies don’t deal with companies; people deal with people. Many users have downsized to the point where they must rely on outside expertise and support. Two types of knowledge should be considered—the vendor’s knowledge of their products as well as their knowledge and understanding of your industry, applications, and best practices. For example, can they help you perform actions such as developing specifications and providing the best practices for implementing overrides? Are their people involved in standards development? Do their people have safety certificates or certifications?

-2	-1	0	+1	+2

Relationship with Vendor

If you currently have equipment from one vendor, how well have they been supporting you? If you're installing a new system in another part of your plant, do you really want a completely different system than the one you may already have, be used to, and have spare parts for? If you buy a second system from another vendor, you'll have to send people to more training classes, stock more spare parts, and so on. The devil you know may be better than the devil you don't. You don't change spouses just because a prettier version walks by (no matter how much you may want to); the cost of "change" may be too great. Or, might you be replacing all your systems plant-wide? Is your current system aging and becoming difficult to support? Does your current vendor have a migration path to a newer system? Sometimes, the cost of *not* changing may be too great.

-2	-1	0	+1	+2

Support

No matter how superior a product may be, there will always be at least one problem at some point. Downtime costs a lot of money in most industries. Can the vendor provide the timely support you need? This may be remote via the phone and/or Internet. Do they have service people that can travel to your location? Do they have people—their own or partners—that are local and can provide service even quicker? Do they offer support agreements or contracts?

-2	-1	0	+1	+2

Integration by Third Parties

Some vendors will not allow others to integrate their hardware. This could be because they wish to limit their liability; their product may be too difficult for others to integrate, or they may simply wish to keep their staff of specialists employed. If that vendor has no presence in your geographical area, you may not be comfortable with the remote relationship, or it may not meet contrac-

Index