Proactive minimization of systematic failures in safety instrumented systems

Peter Clarke PhD CFSE

xSeriCon Limited

December 2012 – revision 0.3

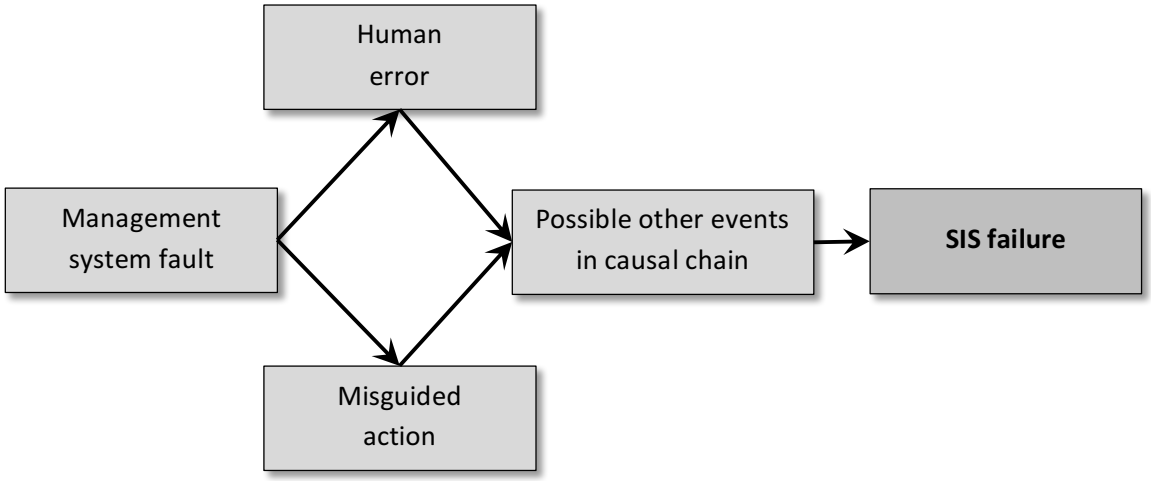
ABSTRACT

Safety instrumented systems (SIS) control major risks in process facilities, and are subject to two types of failure: random and systematic. Random failures, usually in hardware elements operated within their manufacturers' guidelines, are well understood and can be modelled and allowed for in SIS design. Conversely, systematic failures, which result from human error in areas like specification, design, fabrication, programming, operation and maintenance, are typically not well understood and are difficult to measure and control. As hardware improves and system complexity increases, systematic failures are becoming relatively more significant. This paper explores the meaning, origins and existing control measures for systematic failures; develops a strategy for the measurement of systematic failures; considers why existing error control strategies may be inadequate; and develops ideas for a proactive approach to SIS management to reduce the occurrence of errors before they happen. Methods are suggested for optimizing the use of the management tools mandated in the functional safety standards IEC 61508 and IEC 61511, such as the Safety Requirements Specification.

INTRODUCTION

Automated shutdown systems are becoming more and more important in protecting life, property and the environment. These systems are often referred to as Safety Instrumented Systems (SIS), and their design and implementation is addressed by a suite of international standards grouped around an umbrella standard, IEC 61508. Failure of these systems is always a possibility; the underlying causes of failure can be grouped into two categories: *random* and *systematic*. The first of these, *random* failures, is now well understood, and can routinely be modelled and analysed mathematically using techniques including FMEDA[7,8] and Markov models.[9] Broadly speaking, random failures are physical hardware failures caused by environmental stressors within the nominal operating limits of the equipment. Systematic failures, on the other hand, are much harder to define, model or control. As a generalisation, they could be described as failures arising from faults in a management system (hence, the term *systematic*). Such faults often lead to human errors which, eventually, may manifest themselves as functional failures. Figure 1 shows a simple depiction of the causal chain of events leading to a systematic failure in a SIS.

Figure 1. The origins of systematic failures



Since we need to develop reliable SIS, and ensure their ongoing reliability, control of both random and systematic failures is vital. Nowadays, the focus is shifting from control of random failures towards systematic failures. There are number of reasons why this is so. Firstly, continual improvements in hardware reliability and diagnostics are leading to reduced random failure rates. Increasing system complexity is also creating more opportunities for systematic failure.[3] Another important factor is that ever-increasing project size and tighter cost controls mean that projects nowadays involve many different entities—owners, licensors, major and minor contractors, suppliers and consultants. This creates opportunities for misunderstandings and miscommunication, which are another contributor to systematic failures. Therefore, our efforts and techniques to control systematic failures require more attention than ever. The continuing occurrence [4] of tragic losses and disasters in our industry—Buncefield, Texas City, Fukushima to name a few high profile cases—shows that the battle is nowhere near won.

The first objective of this paper is to deepen understanding within the industry of the meaning and importance of systematic failures. Secondly, causes of systematic failures are explored, along with current understanding and best practice for their control. Finally, we will argue that current systematic failure control methodologies are predominantly reactive in approach, and a proactive strategy will be outlined and justified.

DEFINING AND RECOGNISING SYSTEMATIC FAILURES

Put simply, systematic failures are caused by human error. In order to understand how to control them, it is important to recognise that the observed failure is merely the final event in a long chain of events. The sequence of events will include one or more specific errors made by particular individuals; however, it is also likely to include a number of less tangible human factors such as excessive workload, stress, or confusing information. Control of systematic failures needs to take into account not only specific, definable errors, but also these intangible factors that create the conditions under which errors are more likely to occur. This theme is addressed in later sections of this paper.

We should also emphasize that the final error in the chain of events, typically made by a process operator or maintenance technician, is not usually the most important or significant event in the chain. For example, a study of incidents in a one-year period in the nuclear industry showed that 92% resulted from systematic failures, but only 8% of these were initiated by the operator.[2] According to Moray, "even if automation were to replace operators entirely, it would not remove the opportunity for human error: rather, the locus of human error would change. An increased burden would be placed on the designer, who would have to foresee all possible failure modes of the system."[5] Our instinctive need to blame a named individual for an incident often leads us to focus on operator error, but this is not likely to lead to significant improvements in overall safety performance, for reasons that are well understood (see [1], p193).

Proactive management of risk is now generally the favoured approach, in preference to reactive management. Whilst techniques for proactively understanding and controlling random failures are now well developed and widely adopted, the situation is not so clear for systematic failures. Equipment design errors are well addressed by IEC 61508 parts 2 and 3, but the remainder of the functional safety lifecycle is less well developed in terms of proactive systematic failure avoidance. (Systematic failure control features of the lifecycle-based standards are listed in Table 4 later in this paper.)

Risk-based safety management requires us to assess the magnitude of risks; in order to do this, we must have a way of modelling or measuring them. This is necessary because the performance of any systematic failure avoidance strategy needs to be determined. Modelling random failures is now routine, but there is no method available for modelling systematic failures. This is partly because there are too many different failure modes to model—in other words, whilst a piece of equipment can fail randomly in only a small number of ways, there are huge numbers of possible errors that could lead to systematic failure. For example, even a simple chunk of software code could conceal any combination of hundreds of different errors, each of which may lead to failure under the right circumstances.

Since the modelling of systematic failure rates is not feasible, we must depend on other measurements instead. For a specific safety instrumented function (SIF), the observed total failure rate F_{tot} in a given operating environment is the sum of the random failure rate F_{rnd} and the observed systematic failure rate F_{sys}. In a SIL-capable system, F_{rnd} should be low, and it is unlikely that enough random failure events will occur to enable meaningful measurement of F_{rnd}; hence, a theoretical value can be used, which is obtainable either from related statistical data—if the equipment has many thousands of hours of operating experience in similar applications across the industry—or from calculations based on component failure data, using the FMEDA approach.

Thus, we have a method for assessing F_{sys}, based on the difference between expected (F_{rnd}) and observed (F_{tot}) total failure rates. A ratio of $F_{tot} : F_{rnd}$ significantly greater than 1 indicates that an unacceptable number of systematic failures is occurring. A suitable target for the ratio could be <1.1, with 1.3 set as a threshold for urgent investigation.

This approach to systematic failure analysis is, however, fraught with difficulty. Firstly, it is intrinsically reactive, and yields an indicator of failure rates that lags behind the actual commission of the underlying errors—possibly years behind. Secondly, it is critically dependent on accurate reporting and logging of SIS failures. This challenging area is explored further in a later section of

this paper. Thirdly, it requires meaningful values of F_{rnd}. Studies [11] have shown that the values from FMEDA are highly dependable, but this may not be true of values obtained by laboratory methods such as rapid cycle testing for low demand elements, which tend to be highly optimistic as they do not represent real operating conditions.[10] Statistical random failure rates are also dogged by the problems of distinguishing between random and systematic failures, and ensuring that the operating conditions under which data was collected closely match the conditions in the plant under consideration. Relevant conditions include environmental and ambient conditions, aggressive or harsh environment, vibration, corrosive substances, quality of maintenance, and even spurious trip and demand rate (frequent trips impose stresses on equipment that can lead to more frequent random failures).

In short, measurement of the ratio $F_{tot} : F_{rnd}$, which is relatively straightforward, can be used as a key performance indicator (KPI) to determine whether deeper investigation of systematic failure causes is worthwhile. If it is worthwhile, the methods in the following section may be considered.

TOWARDS ACCURATE FAILURE RATE MEASUREMENT

The discussion above demonstrates that, if we wish to measure our existing F_{sys} and our progress in reducing it, we are critically dependent on having accurate failure statistics for our SIS. This means that we need to know about every incident involving the SIS, including:

- successful trips of SIFs
- unsuccessful attempted trips of SIFs (i.e. cases where the final elements did not fully operate on demand)
- discovery of a fault by diagnostics
- discovery of a fault by inspection or testing
- discovery of a fault or error by any other means (e.g. somebody notices that an override has been left activated after completion of maintenance)

Additional data is required for each type of incident, so that it can be categorized as a random or systematic failure and the underlying cause can be understood. All possible events that reveal a fault are shown in Table 2 below, along with examples of the data collection requirements. Each type of event contributes to one or more of the rate values of interest—either a true demand rate, or a failure rate.

Table 1 summarises the failure rate quantities used in our discussion.

Table 1. Definition of demand rates and failure rate quantities

R_{dem}	Observed rate of demand on the SIF due to real process upsets
$F_{rnd,the}$	Theoretical rate of dangerous random failures, obtained from industry statistical data or by FMEDA

$F_{rnd,obs}$	Observed rate of dangerous random failures. If this is significantly higher than $F_{rnd,the}$, it may indicate the presence of systematic errors in design (e.g. equipment located in excessively harsh environment) or maintenance (equipment not replaced at the end of its useful life, covers not properly re-seated, etc).
F_{sys}	Observed rate of discovery of systematic failures that destroy capability of the SIF to mitigate the risk
$F_{sys,NC}$	Observed rate of discovery of noncritical systematic failures that cause upset or lost production, but do not prevent the SIF from mitigating the risk

Table 2. Types of SIS-related incidents, and their data collection requirements

Type of event	Cause of event	Specific data requirement	Which rate value(s) this event contributes to
Trip due to a real process upset	A real process upset	PV trend	R_{dem}
Trip due to hardware or software failure, operator action, maintenance, equipment damage, sabotage, or loss of utility	Any cause other than trip by diagnostics	PV trend, ambient conditions at location of field equipment; interview data if caused by manual trip or maintenance	$F_{rnd,obs}$ for random failures $F_{sys,NC}$ for systematic failures and human actions
Automatic trip by diagnostics	Diagnostics detected random hardware failure	Diagnostic event log, confirm hardware faulty	F_{rnd}
	Diagnostics detected systematic hardware or software failure	Diagnostic event log, confirm error leading to failure	$F_{sys,NC}$
	Misdiagnosis, unnecessary trip	Diagnostic event log, confirm hardware not faulty	$F_{sys,NC}$
Failure to trip on any of the demand causes above	A real process upset	PV trend	R_{dem} and: $F_{rnd,obs}$ for random failures F_{sys} for systematic failures
	Any spurious demand or diagnostics	As above	$F_{rnd,obs}$ for random failures F_{sys} for systematic failures
Discovery of a fault during maintenance, routine operation, or audit/review	Dangerous hardware fault (SIF inoperable as found)	Maintenance record	$F_{rnd,obs}$ for random failures* F_{sys} for systematic failures*

	No-effect hardware fault (SIF operable as found)	Maintenance record	None (does not affect SIS performance)

*If the fault should have been discovered during previous inspection, that represents a critical systematic failure, so add another F_{sys}.

Additional data that should be collected for each incident is shown in Table 3, along with the purpose of collecting the data. It may be possible to collect much of this under existing procedures such as testing and near miss reporting.

Table 3. Additional data collection requirements following SIS-related incidents

Data requirement	Purpose of collecting data
The identity of hardware involved in the failure (make, model, tag number, serial number)	To assess $F_{rnd,obs}$ vs. $F_{rnd,the}$ for each individual model of hardware. To discover whether any particular tag(s) are exceptionally prone to failure (may indicate hardware unsuitable for environmental conditions in that location).
The identity and version number of any software involved	To confirm correct version of software in use. Review testing and checking done on that version of software.
Date and time of the incident	To calculate time to restore SIF to normal state; compare with expected mean time to restore (MTTR) assumed during SIL verification.
Date and time the report was logged, and name of the person making the report	To enable follow-up questions to be asked if necessary. However, anonymous reporting could be considered if it will help to increase reporting rate and objectivity.
State of the process at the time of the incident (e.g. normal operation, startup, shutdown in progress, under online/offline maintenance)	To aid assessment of the cause of the event. To confirm assumptions made about the demand frequency during different phases of plant operation.
State of the SIF after the incident (normal, tripped, failed)	To calculate the actual availability of the SIF, for comparison with the predicted availability.
Result of investigation into the cause of the incident, and any follow-up carried out	To confirm cause of the event. To demonstrate proper follow-up of incidents relating to safety, environment, quality and profitability.
Restoration work carried out to bring the SIF back into an operable state	To confirm that proper, permanent repairs are made.
Date and time SIF is reset to normal state	To calculate time to restore (see above).

Since such a huge amount of varied data is required, how can we best ensure it is adequately collected? Specialised software is available to support this [6], but it requires reporting of incidents and manual data input. In principle, some of the data needed can be collected automatically by the SIS and DCS, and logged in a plant historian. Software needs to be developed to capture the critical elements of this data and store it in a form that readily lends itself to failure rate analysis.

Other data, however, can only be captured by humans, and depends on conscientious reporting and information gathering practices. Similar to general safety management systems, this relies on data collection policies that

- are easy to use for operators and maintenance personnel,
- make it easy to collect the right kind of data to be useful,
- have a genuine no-blame culture associated with any corresponding incident followup,
- encourage all personnel to report all relevant incidents, even their own errors, and
- are widely known, understood and respected throughout the organisation.

UNDERSTANDING THE CAUSES OF HUMAN ERROR

The main factors leading to human error—such as such as lack of training, fatigue, stress, poor HMI design, and boredom—are well recognised. Also, we have already mentioned that most failures originate with errors made during the design phase, not with operators. Despite this, not enough effort has been made to implement working practices for design engineers that address the typical causes of errors just mentioned, and this is an area with great potential for improvement.

Lessons learned from previous accidents have also revealed other, less well understood issues, which are discussed in the following paragraphs.

Drift into failure

There is often a gulf between those who write procedures, and those who have to follow them. For all of us, as we do our work—especially if it is repetitive and relatively invariant—our working practices naturally evolve into a rhythm that is comfortable and usually efficient, but typically not fully procedure-compliant. Often, this evolution is driven by commercial pressures—the requirement to do more work with less resources—or by complacency: "Nothing has gone wrong yet, so we can relax a bit." Checking becomes less meticulous; temporary fixes become permanently forgotten; emergency workarounds become unofficial standard practice. These incremental changes have been described as "borrowing from safety," in the sense that we are trading safety for expediency.

Thus, little by little, failure becomes more likely: a "drift into failure."[1, pp17ff]. Unless someone boldly presses "Reset" and insists that everything is done by the book once more, the accumulated changes will eventually create the right conditions for a critical failure. This critical failure might not even be the result of an individual error; it is more often the fruit of numerous decisions to flex the rules, resulting in the organization inching towards the boundary of safe operation. Some researchers have shown this cultural drift to be at the heart of major disasters such as the loss of the Challenger Space Shuttle. This kind of situation generates design weaknesses that are impossible to identify, so it is imperative to prevent them, proactively, by developing workable procedures, sticking to them, and meticulously applying management of change (MoC), even for procedural changes. Auditing is also essential to ensure that procedures are being followed successfully.

One particular problem with the "drift into failure" is that it affects incident reporting. When a procedural deviation is no longer an incident because it has become standard practice, it will not be reported. Thus, one opportunity to intercept the chain of events leading to catastrophe is lost.

Decision-making in context

"It seemed like a good idea at the time." Many errors, especially those made by conscientious, hardworking individuals, were actually responses to particular situations or stressors that existed at the time. What turned out to be an error in retrospect may have originally seemed like the best course of action, or perhaps the lesser of two evils. It may not have seemed like an error, or may not technically have even been an error at all. In other words, we often perceive errors only in hindsight. This has led some philosophers to the rather extreme view that there is no such thing as an error in absolute terms, since every act that we describe as an error is only recognised as a result of our perception of its outcomes. (It's a bit like the question of whether a falling tree makes a noise when nobody is there to hear it.)

The practical upshot is that we cannot expect to understand why someone performs an act that turns out to be wrong, if we only look at it through the spectacles of hindsight. That would only lead us to make statements beginning with "He should have …" or "If only she had …". Instead, we must understand the context in which our engineers are operating. The correct question to ask is, "Given the situation prevailing at the time, and the information and procedural flow *that he/she would normally experience*, was the act reasonable?" The answer will frequently be *Yes*, and this shows that the problem lies in the context, not in the engineer's response. The need to resolve the context to avoid such opportunities for faulty acts is another argument for a proactive approach.

An interesting example is described by Dekker ([1], chapter 5). In 1995, the cruise liner *Royal Majesty* ran aground off the New England coast as a result of a fault on the GPS. The system had defaulted to a fail safe mode which was not able to estimate the ship's position correctly. A warning flag was displayed on the GPS readout, but this was ignored as it was in tiny letters and, anyway, was often raised in normal operation. Other warnings such as radio messages and missed marker buoys were rationalized away by the ship's crew on the basis that they "did not apply" or "were to be expected." In other words, the crew chose a trajectory that, in hindsight, turned out to be wrong, but was actually reasonable in their real-time context. Remedial measures should focus primarily on fixing the context and culture, not on blaming the crew.

One practical implication of this finding is that we need to examine the conditions that make decision makers believe a particular course of action is right or wrong. Right trajectories through the decision tree need to be reinforced and encouraged, and wrong ones discouraged, by manipulating working conditions and information flows. Dekker ([1], p201) summarises the situation nicely by quoting from Claus Jensen's investigation into the *Challenger* space shuttle disaster: "…there is no point in expecting individual engineers or managers to be moral heroes; far better to put all of one's efforts into reinforcing safety procedures and creating structures and processes conducive to ethical behavior."

EXISTING SYSTEMATIC FAILURE STRATEGIES

The functional safety standards based on IEC 61508 stipulate a range of measures that are designed to reduce systematic failures in one way or another. These measures are built around the framework of the "functional safety lifecycle," which is an overall structure for managing SIS from cradle to grave. As Table 4 shows, the measures are mostly reactive in nature, although one may hope that identification of errors already committed would lead to proactive prevention of those errors in the next project. There is scope for more directly proactive measures, as outlined in the following section.

Despite the widespread acceptance of the general concept of functional safety management espoused in the standards, there is still little understanding of the importance of these measures. Too often, they are regarded as tick-box tasks or simply omitted. In particular, the power of the Safety Requirements Specification (SRS) as a focal point of reference for all SIS-related design work is not generally appreciated. Engineering contractors should raise awareness of the SRS's function, and reshape their working procedures around it.

Table 4. Systematic failure measures prescribed in functional safety standards

Measure	What it entails	How it controls errors	Proactive or reactive?
Functional safety planning and management	Developing procedures and resource allocation for compliance with standards	Creates the environment for all the other measures listed below to take place naturally	Proactive
Competency assurance	Ensures only competent persons perform safety lifecycle tasks	Reduce errors that arise from lack of knowledge or experience	Proactive
Safety Requirements Specification (SRS)	Developing a detailed description of the performance and safety requirements of the SIS	Provides a single point of reference for all subsequent lifecycle activities, including validation. Reduces miscommunications. Removes ambiguities and challenges 'default' design assumptions.	Proactive
Verification	For each lifecycle phase, confirm outputs meet requirements of that phase	Trap errors of omission. If detailed enough, may also trap specific errors in calculations, document versions, etc.	Reactive
Software "V" model	Perform a series of inspections and tests on software during the development process	Detect errors in software design and coding	Reactive
Factory Acceptance Test	Inspection and testing of critical hardware and software before installation	Detect discrepancies between specification and as-manufactured hardware and software	Reactive

Measure	What it entails	How it controls errors	Proactive or reactive?
Validation	Detailed testing and documentary review to confirm SIS, as commissioned, complies with the Safety Requirements Specification	Detect errors or omissions in construction, commissioning, documentation, and procedures required during operational phase	Reactive
Functional safety assessment	Review of documents and equipment to confirm tolerable risk targets are achieved	Detect management-level deficiencies in execution of safety lifecycle	Reactive
Cyber security audit	Confirm adequacy of measures taken to prevent abuse of networked safety systems	Trap failures to protect networked systems	Reactive
General audit	Confirm compliance with procedures	Find specific deviations from procedure	Reactive
		Identify needs for procedural change or training	Proactive

PROACTIVE STRATEGIES FOR SYSTEMATIC FAILURE REDUCTION

The following sections suggest additional ways in which working processes and tools can be designed to proactively reduce errors that can lead to systematic failure.

BUILDING A MANAGEMENT OF CHANGE PROCESS THAT WORKS

One of the key steps in the MoC process is to determine the impact of any change. This is, in fact, an impossible feat of reverse engineering. In order to know the effect of changing parameter *A*, it must be possible to know every decision that used the value of *A* as an input variable, either directly or indirectly. Even if all these decisions were perfectly documented throughout the entire lifetime of the project since inception (a highly unlikely state of affairs), it would be unfeasible to search all the records for every application of *A*, and then to confirm whether there is any logical link between those applications and the MoC request sitting on the desk. This is a bit like trying to build an entire causality tree starting from the twigs.

Instead of trying to build the tree in the reverse direction, a better approach would be to build it in the forward direction, proactively, during the analysis and design phase of the project. One possible approach to constructing the required edifice of information is the following two-step method:

1. Every time a parameter is used to make a design decision, construct an *Input-Output Pair* (IOP). The input is the parameter name (not its value), and the output is the description of the decision (not its result). A reference to the document recording the decision is also advisable.

2. Build a single, searchable register of the IOPs. This should be treated as a controlled document, and may need updating after each design change.

Here are some imaginary examples of how the IOPs might be constructed:

> ***Case 1.*** During a HAZOP study, an LPG-containing vessel is under consideration. This has a pressure control PC-201, which may fail and allow the pressure to fall, leading to a drop in temperature. The downstream piping material of construction is Low Temperature Carbon Steel. The HAZOP team accepts this as sufficient reason to assume that the piping will not suffer brittle failure in this event. The corresponding IOP is: Input = downstream piping material; Output = susceptibility to brittle failure in event of PC-201 failure. A separate IOP would be constructed for each affected item: the piping itself, plus any valves, check valves or restriction orifices.
>
> ***Case 2.*** During a sizing analysis for a pressure relief valve (PRV), a case of gas blowby from an upstream vessel is considered. The engineer's calculations show that the gas blowby case is not the maximum flow case for this PRV, as the flow is limited by the maximum upstream pressure and temperature, and the cross-sectional area of the valve through which the blowby would occur. Three IOPs would be constructed: their inputs are the maximum upstream pressure and temperature, and the valve cross-section, respectively; they have a common output of the PRV sizing.

These two examples illustrate the necessity of recording IOPs from *negative* as well as *positive* results. In Case 1, the HAZOP team found that the material of construction was adequate, and no safeguard was required—a negative result. In Case 2, the gas blowby case was found not to be the determining factor in the PRV sizing—another negative result. Meticulously recording all IOPs, not just positive ones (which are, in some ways, more obvious) is vital, because it is the negative cases that are easier to overlook during MoC studies. Indeed, in some cases they may not be recorded in design documentation at all, and therefore impossible to search or trace.

Once the IOPs have been assembled into a register, it is evident that they provide a powerful tool for MOC analysis. Suppose the LPG vessel in Case 1 above requires urgent replacement of a downstream check valve, but only a normal carbon steel check valve is available onsite. Is this acceptable? A quick search of the IOP register would reveal the risk of brittle fracture in the event of PV-201 failure. So, the MoC decision might allow the change as a temporary measure, with the operators being alerted to stop flow through the check valve in the event of PV-201 malfunction, and a strict time limit imposed for replacing the check valve with one made of the original material.

As a plant ages, the original engineering personnel are replaced by new staff who have no knowledge of how the plant was originally intended. On finding a piece of equipment they do not understand, they may make potentially dangerous assumptions about its purpose, and whether it is still needed. One benefit of the IOP register is that it would remove the uncertainty about whether it is safe to modify or remove a piece of potentially obsolete equipment.

Clearly, the IOP register would have to be quite detailed and thorough to be effective. Developing the IOPs during HAZOP studies and other engineering activities might seem arduous at first. It need not add a significant additional burden, however, if aided by well-designed software.

OTHER PROACTIVE STRATEGIES FOR SYSTEMATIC FAILURE REDUCTION

A successful proactive approach to error-free design engineering should address the following three aspects: creating an environment and tools that foster "right first time" working; preventing drift into failure; and taking advantage of early warning signs of failure. These are explored in the following paragraphs.

Right first time

The worlds of total quality management and Six Sigma have already shown companies how to proceduralize their work and eliminate variance. As we have seen, this is not enough to prevent human error, especially for non-routine tasks (like much design engineering work). The workplace, environment and tools need to be humanized, instead of forcing humans to work in a fundamentally nonhuman, mechanistic way. Engineers need sufficient control over their work practices and environment to enable each individual to work effectively and correctly. For some, this may mean a return to peaceful enclosed offices instead of cubicles in a shared, noisy and distraction-ridden openplan engineering office. Workloads and stress levels should be adjusted appropriately. It is common for engineers in Asia to routinely work 10-12 hour days, sometimes 6 days per week, in order to meet overzealous cost reduction targets: this is a dangerous way to work, as it may end up costing more money and causing increased risk in the long run.

Software tools should be designed to work in a more humanlike way. For example, HAZOP recording software typically requires the team to fill in elaborate tables, making extensive cross-references to drawings by manually typing tag numbers. There is no direct cognitive connection between the study inputs—such as P&IDs—and the outputs, the worksheet and recommendations list, and nothing to encourage correct, complete working or prevent errors. Instead, HAZOP recording software should take a visual form, allowing the team to work directly on the P&ID, visualise the elements that interact in the HAZOP scenarios (such as failures, consequences and safeguards) and capture their findings in a way that directly connects the team's thoughts to the P&ID's spatial representation of the process. Similar improvements could be made in all other computer-aided work processes. Even traditional structures such as text documents and spreadsheets need to be redesigned to make it easier to work right and harder to work wrong. Tools need to be provided and promoted to enable easier, more efficient review: 'tracked changes' functions available in some software are a start, but are of limited value in their present form.

Preventing drift into failure

As we have seen, two factors are strong contributors to the tendency to drift into failure: poor management of change—already addressed in an earlier section—and problems with procedures. Training should be provided on the importance of procedures, as part of a "right first time" culture, and backed up with effective auditing. If procedures are not workable in practice—in other words, if they are not achieving their intended purpose—they should be revised, preferably by the people who have to follow them. The revision process should be facile and transparent, while being subject to impact analysis just like any other change that may have a safety impact. Workarounds should

not be tolerated as a normal practice. Also, the purpose of procedures should be made known, since this may make them more likely to be accepted and followed.

Other factors that may also need to be re-evaluated include motivation, multi-tasking and boredom. A bored or overstretched engineer is certainly not likely to be error-free.[3] Managers need the freedom and resources to create a space that provides optimal psychological reward, tailored for each employee.

Early warning signs of failure

Near miss systems, already widely used to address incidents with direct safety impact, should be extended to handle situations that could lead to harm further down the causal chain. Such situations should include not only procedural violations, but also unworkable procedures, inadequate review and checking processes, and inappropriate working conditions. The handling of these types of incident could usefully be integrated with systems already in place for ISO 9000 deviations, with a potential concomitant benefit for quality as well as safety (which, in the area of design engineering, are arguably one and the same).

Incident reporting should be strongly encouraged and incentivised, for example through bonus structures and personal targets; true no-blame and, if necessary, anonymous reporting cultures should be fostered. Why don't we ever hear our organizations report: "Congratulations! Our incident reports are up 50% year on year!" Instead of tying bonuses to hitting low accident rate targets—as some organizations absurdly do—we should tie them to incident reporting targets, and high ones at that.

CONCLUSIONS

We are faced with growing commercial pressures, continued fragmentation of safety lifecycle tasks among large numbers of stakeholders, and the ever-increasing complexity of the risks controlled by SIS. Given these challenges, further reduction of major incidents due to systematic failures is virtually impossible unless a new approach is taken to reduce the underlying causes of failures. The proactive strategies described in this paper may go some way towards addressing this needed paradigm shift. Some of the proposed measures, such as advances in utility software design, can be propagated to benefit the whole industry at relatively little cost. Others, such as improved management of change and more rational procedure design, will have spin-off benefits such as cost reduction and improved quality. At the very least, it is vital for everyone involved in SIS management to recognise that operator error is rarely, if ever, the true cause of an accident. System level solutions need to be found, and this needs a bottom-up redesign of the way we, as humans, do our work.

REFERENCES

[1] Dekker, S.W.A. Ten questions about human error: A new view of human factors and system safety. Mahwah, USA: Lawrence Erlbaum, 2005.

[2] Stanton, N. (ed.) Human factors in nuclear safety. Taylor & Francis, 1996, chapter 1.

[3] Redmill, F.; Rajan, J. Human factors in safety-critical systems. Butterworth-Heinemann, 1997.

[4] Anon. The 100 largest losses 1972-2011: Large property damage losses in the hydrocarbon industry, 22nd edition. New York, USA: Marsh Ltd, 2012.

[5] Moray, N., in Noyes, J.; Bransby, M. (eds). People in control: human factors in control room design. London: IEE, 2001, page 102.

[6] exida.com LLC, SILStat [software]. http://www.exida.com/index.php/software/SILStat/ (accessed 30 September 2012)

[7] Grebe, J.C.; Goble, W.M. FMEDA—Accurate product failure metrics. Sellersville, USA: exida, 2007 [online]. http://www.exida.com/articles/FMEDA%20Development.pdf (accessed 21 December 2012)

[8] Goble, W.M. Control systems safety evaluation and reliability, 3rd edition. Research Triangle Park, USA: ISA, 2010, chapter 5.

[9] Ibid, Appendix E.

[10] O'Brien, C. Improved modeling of mechanical factors through adoption of use factors. Sellersville, USA: exida, 2007 [online]. http://www.exida.com/articles/Mechanical%20Failure%20Rate%20-%20R1.pdf (accessed 21 December 2012)

[11] Goble, W.M.; Bukowski, J.V. Development of a mechanical component failure database. Proceedings of the Annual Reliability & Maintainability Symposium (RAMS), IEEE, 2007.