U.S. Commercial Nuclear Power Plant Digital I&C System Operating Experience

Revision 0, June 13, 2008

Prepared for:

NEI Digital I&C and Human Factors Working Group

Prepared by:

Bruce Geddes, Southern Engineering Services
Thuy Nguyen, Electricite de France
David Blanchard, Applied Reliability Engineering
Ray Torok, Electric Power Research Institute

EPRI Project Manager
R. Torok

ELECTRIC POWER RESEARCH INSTITUTE
3420 Hillview Avenue, Palo Alto, California 94304-1338 ▪ PO Box 10412, Palo Alto, California 94303-0813 ▪ USA
800.313.3774 ▪ 650.855.2121 ▪ askepri@epri.com ▪ www.epri.com

NOTE

For further information about EPRI, call the EPRI Customer Assistance Center at 800.313.3774 or e-mail askepri@epri.com.

CONTENTS

1
EXECUTIVE SUMMARY

This white paper presents the results of a digital operating experience research project commissioned by EPRI and NEI. The project team found 322 reports in NRC and INPO databases (LER, EN, and INPO OE) of plant "digital events" that occurred between 1987 and 2007. The reports include, but were not limited to, plant transients and mishaps caused by digital system problems, problems discovered in surveillance tests, and problems discovered during commissioning of new equipment. Many reports described events in which digital equipment was adversely affected by failures in other components, or events where a digital system misbehaved, and an analog system would have had the same problem (e.g., events caused by incorrect setpoints). For the purposes of this study, any reported occurrence that involved or affected digital equipment was considered a "digital event."

Because the effort focused primarily on addressing Diversity and Defense-in-Depth (D3) issues, the data evaluation approach was tailored to search for the presence of common defects and isolate the events of greatest significance due to actual and potential Common Cause Failures (CCFs) that could disable or significantly degrade safety functions at the system level (e.g., reactor trip, core heat removal, containment integrity, etc.). In the context of digital system upgrades, the regulatory discussion has focused on the concern that software-based systems introduce a new vulnerability in the form of potential CCFs resulting from latent software faults. Therefore, it also was important in this investigation to separate actual and potential CCFs into those caused by software and those caused by other factors (e.g., hardware failures, incorrect setpoints, or operator errors).

Results

In assessing the results, we looked at software errors in the broader context of all the causes of potential and actual CCFs that have been reported, and discovered software to be a relatively minor contributor. Table 1-1 shows a breakdown of the 44 (out of 322) reports that involved actual or potential CCFs that either affected or could have affected system-level functions. "Potential CCF" refers to situations in which system function could have been lost or significantly degraded due to CCF of multiple trains, had a demand signal occurred. "Actual CCF" refers to situations in which such a demand occurred and there was a CCF at the system level.

We differentiate system-level CCFs from potential sub-system or even single-channel failures due to the nature of the conditions required to trigger the common defect. For example, a software defect that is triggered by a transmitter failure is common to multiple channels, but can

only result in a single channel failure because transmitter failures are considered single random hardware failures. In these cases, the overall system function remained unimpaired.

In safety systems, there were no actual CCFs. There were six potential CCFs, only one of which was caused by a software design error. The non-safety events were similarly dominated by non-software causes, with 7 out of 38 events involving software design problems. The greater numbers for the non-1E events are explained in part by the fact that there is more experience with digital systems in non-1E applications, but other factors are likely important, as discussed in Section 5.

Table 1-1
Reports of Actual or Potential System-Level CCFs

Effect	1E Systems		Non-1E Systems	
	Software	Non-Software	Software	Non-Software
Potential CCF	1	5	0	5
Actual CCF	0	0	7	26

It is important to note that 1E systems are much better protected from CCF by regulations and standards that require independence and rigor. Non-1E systems are more susceptible to CCF due to their functional complexity, their use of shared resources (e.g., power supplies), and their fundamental master/slave architectures that are connected to single control elements (e.g., feedwater regulating valves).

In 1E systems, there were more reports of software defects discovered via testing and surveillance activities than actual common cause failures due to software. The associated corrective actions such as software changes and lifecycle program changes were effective, resulting in no repeat events, thus demonstrating improved system reliability. Also, the reported software defects were typically discovered within one or two fuel cycles after implementation.

In 1E systems, there were no reports of events where a demand signal alone triggered or could have triggered a hardware or software defect to create a system level failure. This is not surprising in light of the rigorous development processes and periodic testing practices applied to 1E systems, with particular focus on ensuring that they respond correctly to demand signals.

None of the reports involved situations in which using diverse platforms would have been effective in protecting against potential or actual CCFs. This is consistent with the observation that the defects typically originated in requirements specification or application logic errors, neither of which would be remedied by platform diversity.

The operating experience suggests that current software development and testing practices as well as the design features used in critical plant systems have been effective in keeping software a minor contributor to CCF. Therefore, it is recommended that industry and NRC work together to modify current D3 guidance to endorse and credit methods that have proven effective in protecting against software CCFs in 1E systems.

2
BACKGROUND

NRC/NEI Activities on Digital I&C Issues

The United States Nuclear Regulatory Commission (USNRC) and the U.S. nuclear power industry, through the Nuclear Energy Institute (NEI), are working to develop common understandings and guidance for various technical and regulatory issues related to the use of digital instrumentation and control (I&C) equipment in critical nuclear plant applications. NRC has documented its approach in the *Project Plan – Digital Instrumentation and Control*, Revision 1, February 2008. The NRC plan describes a set of Task Working Groups (TWGs) that are addressing specific topic areas. TWG #2 is responsible for issues associated with Diversity and Defense-in-Depth (D3), which refers to the concern that the use of identical software elements or shared resources creates a vulnerability to common-cause failures (CCFs) of redundant trains of safety equipment or multiple plant systems.

In two different industry communications, one by the Advisory Committee on Reactor Safeguard (ACRS), Letter # 2252, dated May 2007, and one by NRC Commissioner Peter Lyons, Speech to the International Atomic Energy Agency (IAEA), dated June 2007, the message was clear: Industry and Staff should share digital operating experience (OE) to obtain insights regarding failure modes, and use those insights to improve the current regulatory guidance on defense-in-depth and diversity for digital upgrades. This white paper addresses this request.

Definitions

The key terms defined here are vital in understanding the subtle but important characteristics that differentiate the reported events.

Event – A reported plant, system or component defect or failure to meet the full range of functional and performance requirements for normal, abnormal and emergency operations, or any event that is on the NRC list. For the purposes of this study, the term “digital event” refers to any reported event that involved or affected digital equipment.

Defect – A deficiency in characteristic, documentation or procedure.

Common Defect – A defect that affects replicated elements or shared resources such that it could create a vulnerability to common-cause failure (CCF) of redundant trains of equipment or multiple plant systems – for example, a software fault that exists in all divisions of a redundant safety system. Safety and non-safety systems tend to have different types of CCF vulnerabilities:

Safety Systems (1E) – Separation and independence requirements limit common defects to identical or similar components in redundant trains, and requirements errors and omissions.

Non-Safety Systems (Non-1E) – Use of shared resources that could be CCF sources is common, for example, a shared communications network or a power supply that feeds multiple process controllers.

Software Event – An event involving design defects introduced in the software development process (not, for example, incorrect setpoints or flawed requirements, since these defects would also affect hardware or analog systems).

Failure – Degraded or terminated ability of a functional unit to perform a required function. For example, as it relates to software, a failure results when a demand is placed on a system and a latent software defect prevents the system from performing its intended function.

Potential CCF – A defect common to multiple redundancies that can result in an Actual CCF in the presence of concurrent triggers.

Actual CCF – A malfunction on demand that results in an incorrect response or loss of function across multiple channels, sub-systems or systems at the same time.

Trigger – A condition or specific set of inputs that activates a defect; in digital systems, this is nearly always an unanticipated, unexpected, or untested condition.

Common Defects and Triggers

The emphasis of this work on CCFs gives rise to discussion of the importance of "common defects," or defects that exist in multiple channels, sub-systems or systems. Software faults or incorrect system setpoints are examples of common defects. In regard to software, is important to note that for a software defect to cause a channel or system failure, it must encounter a trigger that provokes an incorrect or undesired response. Triggers are initiated by conditions that are independent of the software defect itself, such as a sensor input outside of its expected range.

In regard to D3 and potential software CCF, it is very important to understand that the existence of a common software defect in multiple redundancies does not by itself imply a credible potential for CCF. To generate a potential CCF, there must also be a realistic likelihood of concurrent triggers. Triggers are considered concurrent if the time interval between them is too short for repair or recovery measures to be taken. If, for example, the trigger is a sensor failure that affects only one channel, or an error in a maintenance procedure that is performed on one channel at a time (real examples from the OE), then the defect, though common to multiple channels, is not considered a potential source of CCF. These types of failures are constrained to one channel at a time, and are discovered and corrected before they can propagate any further. In addition, the reports show that discovery of these defects is typically occurring within one or two fuel cycles after initial installation. These considerations are particularly relevant for safety systems, where regulations require separation and independence between redundancies.

Another reason that common software faults do not necessarily result in potential CCFs involves the use of design features that eliminate or avoid potential triggers altogether. Since it is not generally possible to ensure that there are no residual software faults, designers and users of high-integrity digital systems typically incorporate features and characteristics ("defensive measures") that effectively eliminate potential triggers for various types of postulated faults.

For example, digital systems designed for critical applications often use simple cyclic application software and an operating system whose behaviour is not influenced by external conditions. This helps ensure that the software trajectory is confined to a well tested path, and residual faults in other parts of the code cannot cause failures. Data validation routines are often used to prevent software from "seeing" out of range sensor values that might trigger undetected faults, and cause failures or undesired behaviors.

Maintenance and operations procedures and practices can also implement defensive measures, such as avoiding activities that can initiate EMI failures during normal operations (e.g., nearby welding), verifying and installing approved hardware and software versions, and verifying setpoints and parameters are current and approved before they are entered. In practice, digital system designers and users manage many types of potential faults and improve system dependability using defensive measures to ensure detection, avoidance or tolerance of fault/trigger combinations, and thereby eliminate them as credible sources of CCF. Corrective actions for undesired behaviors of digital systems often involve implementation of defensive measures to avoid recurrence.

Nuclear plant design principles often limit the potential effects of software defects, such that they have minimal impact at the system level. The use of internal signal diversity by reactor protection systems is a good example. Several diverse sensor inputs can initiate a reactor trip, and for most accidents and anticipated occurrences there are at least two diverse inputs that can initiate a trip. This means that in many cases, a software fault that affects the processing of one signal, even if it disables that input in multiple redundancies, may not significantly impact the safety function of the system.

A database was prepared for capturing the characteristics of each event in terms of when, where, how, and why the event occurred, and what steps were taken to correct the deficiency that caused the event. The database also captures information such as the involved plant system type, its safety classification, whether or not the event was due to a common defect, and its actual or potential effect at the system, subsystem, or single channel level.

Ground Rules

It is of utmost importance that this research be objective and credible from the perspective of licensees and the NRC. Reporting of nuclear power plant events generally follows a rigorous process whereby event investigations, root cause or apparent cause analysis, and corrective action determinations are objectively established by qualified individuals or teams. When an event report is issued, depending on the nature of the event and the reporting requirements, it has usually passed inspection or review by an independent person or a team made up of senior,

experienced individuals. Therefore, this research was performed under the following ground rules:

- The OE reports accurately represent the facts and circumstances surrounding each event. This paper does not "rewrite history." The database prepared for this work collects and organizes information from the OE reports "as-is."
- Interpreting the OE reports for implications regarding appropriate defensive measures, potential diversity attributes, or other means for preventing or avoiding each event is encouraged, as long as such commentary is clearly noted and separated from the reported facts.
- Discussion and tabulation of causes and corrective actions includes all of the primary and contributing causes reported, and all of the primary and secondary corrective actions reported (typical reports list multiple causes and multiple corrective actions). This was done in an effort to ensure that all the relevant information was considered, and eliminate potentially subjective filtering that might mask important insights. For example, if "inadequate software design" is listed as a failure cause, one might legitimately question what other factors might have been involved, for example, incomplete requirements specification, inadequate V&V, or inadequate testing. Generally, bundling the stated causes and corrective actions provides a better, more complete understanding of the insights revealed by the OE.

OE Library

The primary data source is the INPO OE search engine, available by permission at www.inpo.org. All available document types were included in the search. The public ADAMS search engine was used to supplement the INPO search where needed, at www.nrc.gov. Keywords such as digital, computer, software, EHC, RPS, ESFAS, feedwater, control, protection, and others were used to search for OE "source documents."

The objective of the OE search was to build an event library from source documents, such as Licensee Event Reports (LER), Event Notifications (EN), or INPO Operating Experience (OE) reports for each reported event, so that an independent analysis of each event could be performed.

It is helpful to first recognize that 342 events in the database were previously identified by the NRC in a separate list, "Digital System Failures from 1987 through 2006". EPRI, NEI and Southern Engineering Services gratefully acknowledge this work, and recognize that it was provided by the NRC "for information only" in the mutual interest of learning from operating experience.

It was only possible to evaluate events for which source documents with detailed descriptions could be found. Source documents were found for 160 of the 342 events on the NRC list, but the remaining 182 could not be included in the event categorization studies. The current EPRI/NEI effort added 162 new events with source documents, for a total of 322 that were evaluated out of the original collection of 504 (Figure 2-1).

Figure 2-1
Sources of Digital Event Information

3
EVENT CHARACTERISTICS

Basic Event Characteristics

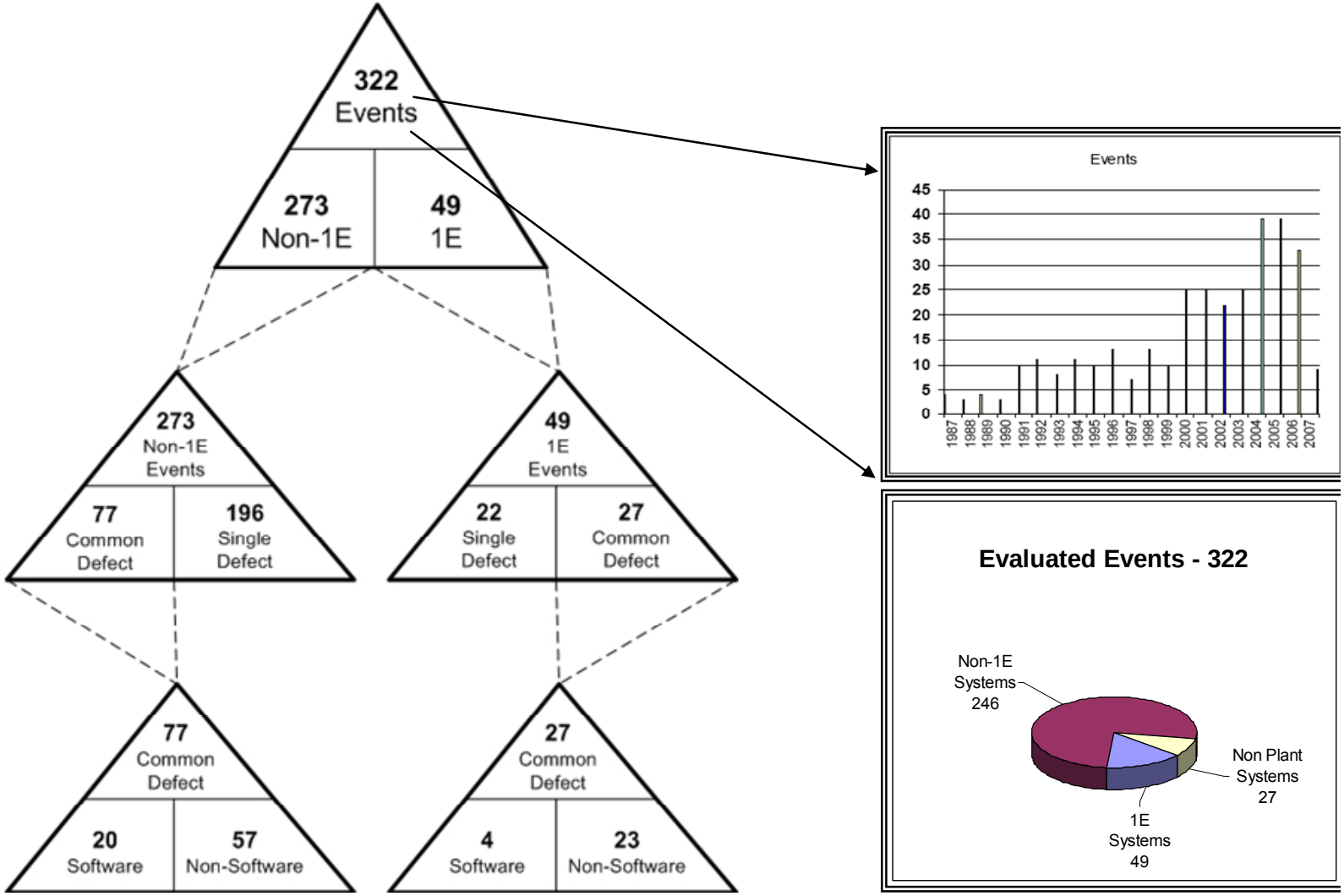
Figure 3-1 shows the breakdown of evaluated events in several ways. In the upper right corner, the spread of evaluated events by year shows peaks between 2000 and 2006, indicative of increased digital upgrade activity. Anecdotal evidence indicates that the number of First-of-a-Kind (FOAK) digital upgrades installed in the last three years has declined significantly, and the figure shows a declining trend in reported events in 2007. One likely explanation is that a large fraction of the reported events correspond to learning curve design errors and mistakes in FOAK upgrades that are discovered and corrected in the first or second fuel cycle after initial installation, and once corrected, are not recurring.

Second, in the bottom right corner of Figure 3-1 is an illustration of system classes associated with the event reports. Out of 322 evaluated events, 246 are related to Non-1E systems, such as plant computer, turbine electrohydraulic control (EHC), feedwater, etc.; 49 events are related to 1E systems, such as reactor protection (RPS), engineered safety features actuation (ESFAS), diesel load sequencer, post accident monitoring (PAM), etc.; and 27 events are identified where the affected system is not a typical plant indication or control system, such as emergency sirens.

Third, on the left side of Figure 3-1 is a basic illustration of the breakdown of events by various categories (see definition in Section 2). This "pyramid" figure breakdown is useful for navigating the various information sets that were developed while evaluating the event reports.

Finally, the pyramid construct in Figure 3-1 illustrates the primary objective of this research, which is to search for common defects and isolate the events of greatest significance.

Figure 3-1
Event Breakdown



4
1E EVENTS

1E Event Overview

Out of 322 gathered reports of events or defects involving digital systems, 49 were related to 1E systems, which include reactor protection, ESFAS, and auxiliary systems (Figure 4-1). Out of these 49 reports, 22 described the presence of a single defect (defect that exists in only one division). The remaining 27 reports described the presence of a common defect (exists in redundant channels or divisions).

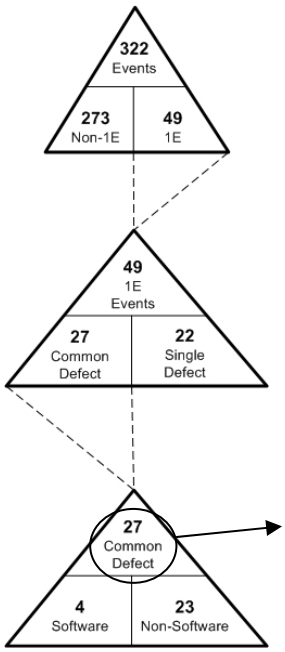


Figure 4-1
1E Events

Effect	1E Software Events			1E Non-Software Events		
	System	Subsystem	Channel	System	Subsystem	Channel
Single Failure	N/A	N/A	2	N/A	N/A	8
Spurious Actuation	---	---	1	3	---	2
Potential CCF	1	---	N/A	5	4	N/A
Actual CCF	---	---	N/A	---	1	N/A
	4			23		

Table 4-1
1E Events with Common Defects (Breakdown by Type and Extent of Impact on System Function)

Figure 4-1 is structured around the search for common defects, which is the primary objective of this work. Table 4-1 shows the breakdown of the 27 1E common defect events by type and extent of the effects of the defect. Note that the effects of a defect that is common to multiple and independent redundancies depend on the nature of the conditions, or triggers, that are required to force a failure.

The columns segregate the events based on the extent or potential extent of a failure.

- For events in the two "System" columns, the defect caused or could have caused significant degradation or loss of the system function.

- For events in the "Subsystem" columns, multiple channels were affected (e.g., one trip function), but the system had sufficient internal defense-in-depth and diversity that the defect could not cause a significant degradation of overall system function (other trip functions). An example of this would be an RPS system defect that partially impaired one trip function in all channels or divisions without affecting the other trip signals or the ability to trip the reactor.

- Events in the "Channel" columns are those for which concurrent triggers (and therefore CCFs) are highly unlikely or impossible; for example, situations in which the concurrent triggers would have to be multiple random hardware failures. Hence, in this column, CCFs boxes are marked "not applicable."

The rows segregate the events by type.

- "Single Failures" by definition do not have subsystem or system level effects (shared resource failures in non-1E systems are included in the CCF rows).
- "Spurious Actuations" involved channels, subsystems or systems taking action when no action was required.
- For events in the "Potential CCF" row, concurrent triggers are needed for a CCF and were judged to be at least somewhat likely, but did not actually occur.
- The "Actual CCF" row contains events where a demand signal occurred in the presence of defects that had been activated by concurrent triggers

Table 4-1 shows that of the 27 events involving common defects in 1E systems, none led to an actual CCF at the system level. **Only one actual CCF was observed, and it had minimal effect on the system function.** The 1E common defect data suggests that the inherent independence of 1E systems (which is required by regulations) results in triggering conditions that more often result in subsystem failures or single channel failures than **CCFs**. Table 4-1 shows that software was a minor contributor to 1E events.

In addition, there were no events in 1E systems where a demand signal by itself activated or triggered a hardware or software defect that caused a system-level failure. This is not surprising in light of the functional simplicity of the systems and the rigorous development processes and periodic testing practices applied to 1E systems, with particular focus on ensuring that they respond correctly to demand signals.

The majority (18 of 27) common defect events in 1E systems resulted in subsystem or channel effects, leaving the balance of the system unaffected, and available to perform its overall safety function by other means using functional diversity or signal diversity. There were no reports of common defects in 1E systems that affected more than one trip or actuation function.

Finally, 26 of the 27 reports of common defects in 1E systems were discovered during surveillance testing or while the system was in service (via spurious actuations, vendor reports, or single random hardware failures). The single event that occurred during a demand situation momentarily affected only one subsystem (trip function) of the RPS, leaving other subsystems fully available.

1E Common Defect Causes

Figure 4-2 illustrates all the "as-reported" primary and contributing causes of the 27 common defect events reported in 1E systems. Most of the reports indicate multiple causes, so the figure includes some double accounting. For example, the three events for which "inadequate software V&V" was listed as a cause, likely are also three of the four events for which "inadequate software design" was given as a cause. Most of the causes of these common defects are related to Lifecycle Management and Human Performance issues. Software design problems, which are the primary concern for D3 and software CCF purposes, played a minor role, appearing in only four of the twenty-seven reports (darker bar). Note that Figure 4-2 includes the subsystem and channel failures shown in Table 4-1.

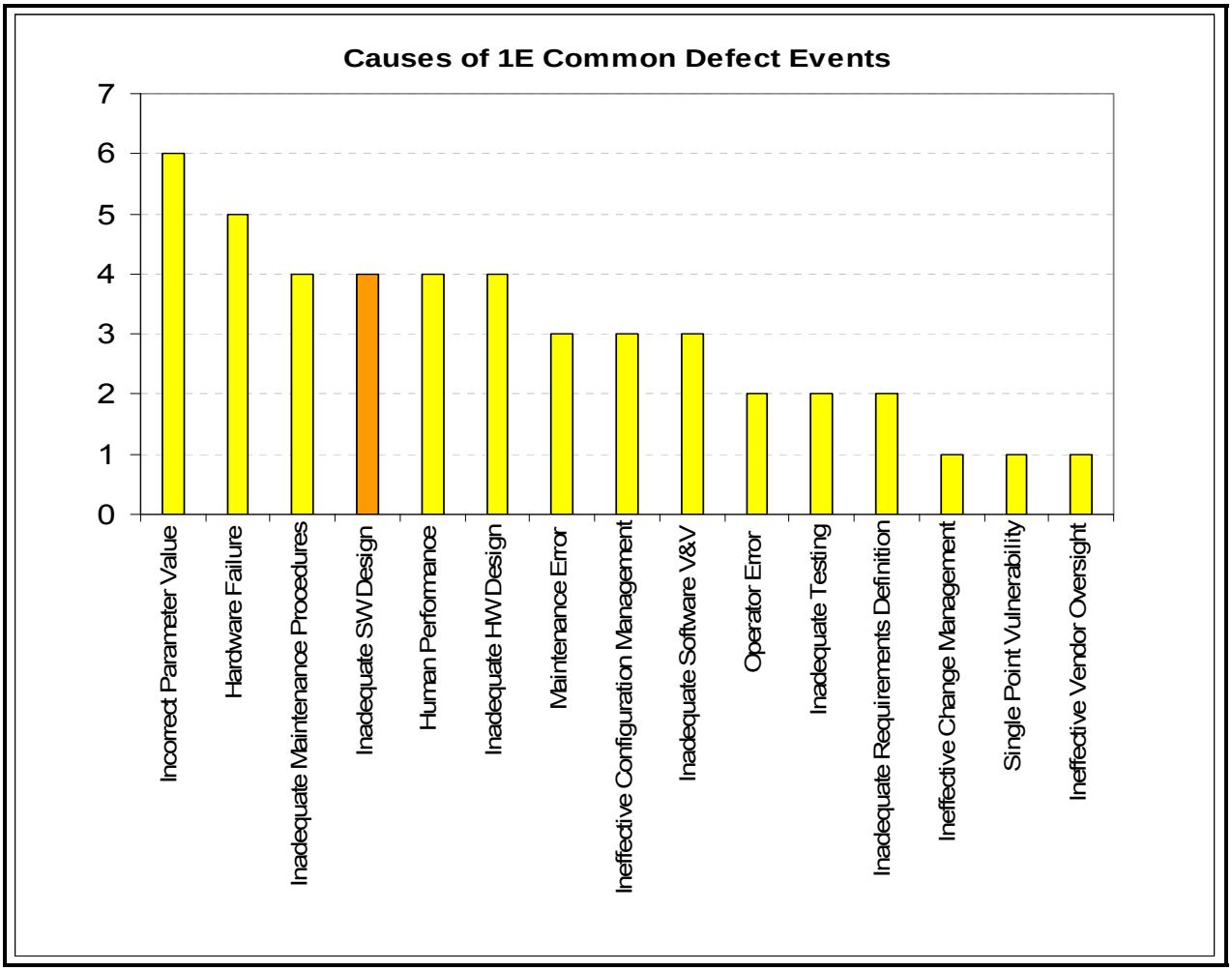


Figure 4-2
Causes of 1E Common Defect Events

1E Common Defect Corrective Actions

Figure 4-3 illustrates the corrective actions after discovery of the 27 common defects reported in 1E systems. The chart includes all the primary and secondary corrective actions, more than one for most events. While most of these actions are related to Lifecycle Management and Human Performance issues, it is interesting to note that Software Changes are performed in twice as many events as those where Software Design issues were reported as causes (Figure 4-2). This trend suggests that licensees are using the software to add features that protect against recurrence of non-software failures.

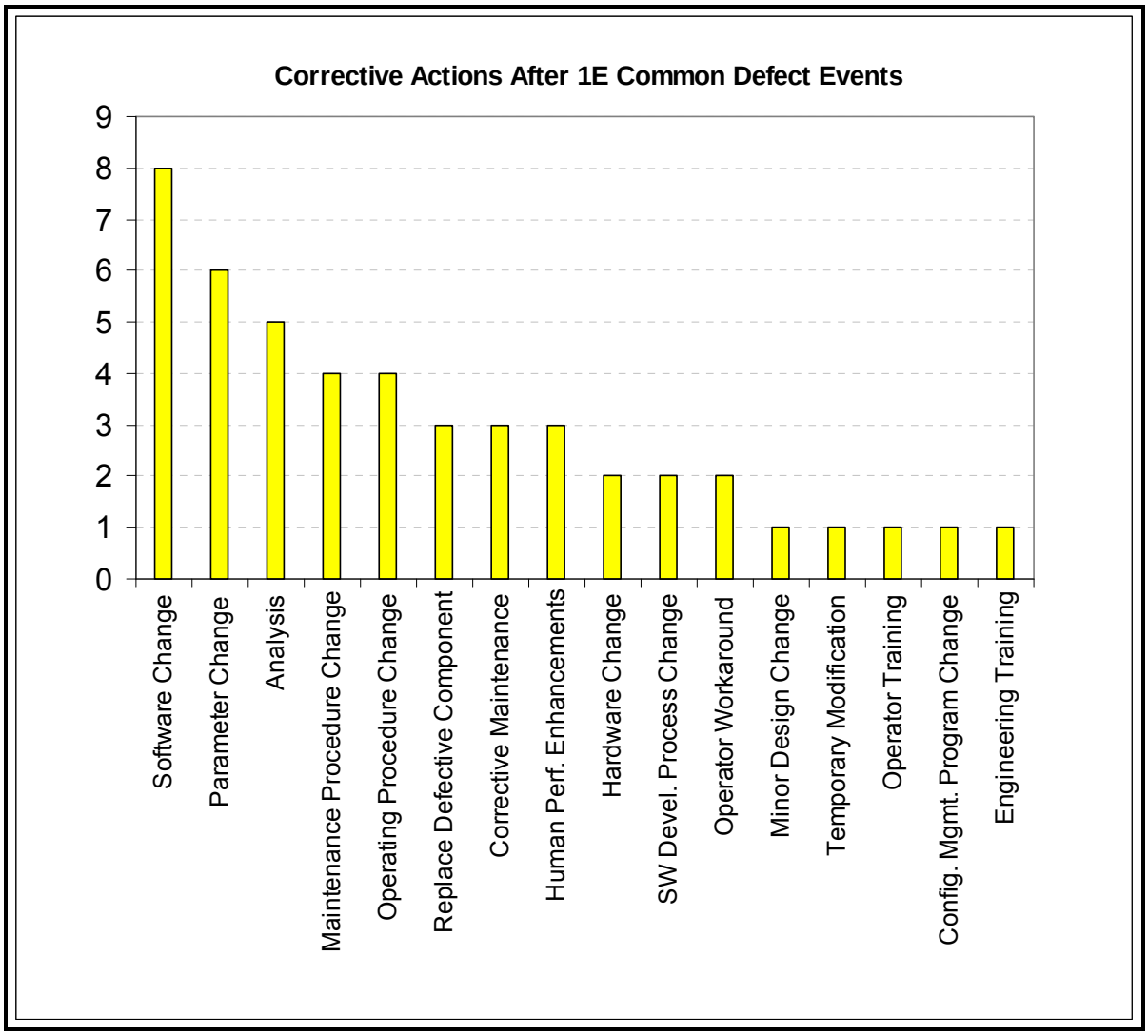


Figure 4-3
Corrective Actions after 1E Common Defect Events

Selected 1E Common Defect Events

1E Potential CCF Caused by Software Fault

Table 4-2 summarizes the characteristics of an Inoperable Sequencer event that occurred in 1994 (the Event number refers to the record number in the OE database compiled for this report). This **1E Software event** is the only one found that could have resulted in an Actual CCF of a 1E system under certain conditions. The root cause was determined to be Inadequate Software Design, coupled with Inadequate Software V&V.

This digital system was deployed with a selectable automatic self-test feature. It was discovered later during surveillance testing that 5 of 18 automatic self test routines running in each of 4 asynchronous sequencer channels had an error in the application logic that would have prevented an actual safety injection (SI) signal from passing through while in auto test mode. The licensee determined that this condition resulted in all 4 sequencers being inoperable some of the time, triggered by asynchronous yet overlapping automatic tests. These individual channel tests were set up on a set frequency, not on every processor scan cycle. The immediate corrective action was to take the system out of automatic test mode, then correct the self-test logic with a software change.

It is interesting to note that adding automated self-testing features to the relatively simple safety function logic led directly to the problem. It added complexity to the system functionality, with corresponding adverse effects on several aspects of system development that are important from a dependability perspective. For example, requirements become more complicated and more difficult to specify with high confidence that they are complete, correct, unambiguous, etc. It becomes more difficult to anticipate, specify and test all the potential abnormal and faulted conditions that the system might see (the Table 4-2 system was operating for a few years before the problem was discovered), and so on. This event reinforces the notion that safety systems in nuclear plants tend to be functionally simple (compare a signal to a setpoint and change a zero to a one if the setpoint is exceeded), and changes that increase complexity should be considered very carefully.

It is also important to note that in this case at least one important defensive measure had apparently not been applied in developing the application software. A cyclic software design, free of conditional statements (timed, periodic tests), independent of external conditions (test switch position) might have been used to ensure that multiple channels could not be disabled simultaneously.

Further analysis and tests by the licensee demonstrated that Operators would have recognized the condition during a LOCA and manually initiated safety injection in time to stay within acceptance criteria.

The resultant impact on Core Damage Frequency was determined to be relatively minor. Successful automatic SI injection does not require all four ECCS trains to be available, so even with the software fault affecting various combinations of the four sequencers, the estimated

availability of automatic SI injection was greater than ninety percent. Further, the defect had no imp act on ability of the operators to act as a backup and initiate the system manually.

Table 4-2
Example of a Potential System-Level CCF Event in a 1E System

Effect	1E Software Events		
	System	Subsystem	Channel
Single Failure	N/A	N/A	2
Spurious Actuation	---	---	1
Potential CCF	1	---	N/A
Actual CCF	---	---	N/A
	4		

(See Table 3-1)

Event #	10	Event Date:	Nov-94	System:	ESFAS

Inoperable Load Sequencer

	System	Subsystem	Channel		
Single Failure				Root Cause:	Inadequate Software Design
Spurious Actuation				Contributing Cause:	Inadequate Software V&V
Potential CCF	X			Contributing Cause:	---
Actual CCF				Corrective Action 1:	Software Change
				Corrective Action 2:	Software Development Process Change

Failure Mode:	Software logic defect in the application code on asynchronous channels can prevent valid safety injection signal from passing through some of the time when in automatic test mode.
Risk Significance:	Auto SI function available 90% of time. Manual actuation available as a backup (SGTR, Small & Med LOCA). Simulator verified manual action could take place in time for Large LOCA

1E Actual CCF at Subsystem Level (Non-Software)

Table 4-3 summarizes the characteristics of a **1E Non-Software event** that resulted in an inappropriate delay of an automatic reactor trip due to an Incorrect Parameter Value. In 1991, the licensee experienced a lightning strike that caused a generator trip and Reactor Power Cutback (RPCB). About 35 seconds later, a reactor trip occurred due to a lower than allowable calculated safety margin for departure from nucleate boiling ratio (DNBR). Post trip review determined that the reactor trip had been inappropriately delayed by 16 seconds. The Core Protection Calculator (CPC) designer had provided a feature in the Control Element Assembly Calculator (CEAC) that can detect a RPCB when Subgroups 4 or 5 drop, which will then cause the CPCs to ignore position information for these rods for 16 seconds in order to prevent an unnecessary reactor trip. A RPCB was successfully performed, then a subgroup in Regulating Group 4 slipped about 11 inches.

The CEAC assumed a second RPCB had occurred, thus inserting (or triggering) a second delay. This second delay was inappropriate in that the plant is designed for only one reactor power cutback following a generator trip. After the delay timed out, the CPCs sensed the subgroup deviation, applied an appropriate DNBR penalty, and correctly initiated a low DNBR trip. The system designers had assumed CEA slips would last more than 0.5 seconds and had set a CEA deviation calculation parameter at this value. The deviation parameter was reset to 0.0 seconds to prevent recurrence.

This event was an Actual CCF at the subsystem level of the RPS, yet it had very little impact on the system safety function. Other, diverse trip signals and the trip function itself were not affected and were still available throughout the event. In fact, the signal that was affected by the problem actually initiated the trip, albeit after an undesired delay.

This event calls attention to a number of important D3 insights. It helps illustrate that an application defect in a subsystem (the CPC in this case) does not necessarily propagate to the whole protection system (the RPS). Reactor protection systems contain significant built-in diversity in the form of different input signals that can initiate trips, such that a software fault in the processing of any one of the signals has limited impact on overall safety function. For most events, at least two diverse signals can initiate a trip in time to avoid exceeding design basis acceptance criteria.

This event also shows why platform diversity is not particularly effective. Using diverse platforms in redundant RPS divisions would have been of no value in this event, because the problem originated in a system design that allowed multiple RPCBs coupled with an incomplete understanding of specific details of rod drop phenomena, which was then reflected in the application software logic. A diverse platform running the same logic would have had the same problem, compounded by increased complexity in its design, operation and maintenance.

Similarly, none of the events reviewed in this work provided examples where adding platform diversity emerged as an appropriate corrective action to prevent recurrence of a problem.

Table 4-3
An Actual Subsystem-Level CCF Event in a 1E System

Effect	1E Non-Software Events		
	System	Subsystem	Channel
Single Failure	N/A	N/A	8
Spurious Actuation	3	---	2
Potential CCF	5	4	N/A
Actual CCF	---	1	N/A
		23	

Event #	222	Event Date:	Nov-91	System:	RPS/CPC

Inappropriate Delay of Automatic Reactor Trip

	System	Subsystem	Channel
Single Failure			
Spurious Actuation			
Potential CCF			
Actual CCF		X	

Root Cause:	Incorrect Parameter Value
Contributing Cause:	Inadequate Requirements Definition
Contributing Cause:	---
Corrective Action 1:	Parameter Change
Corrective Action 2:	---

Failure Mode:	Control element assembly (CEA) calculation software did not account for CEA slips less than 0.5 sec - sent time delay to CPC, delayed Rx trip by 16 seconds. Degraded (but did not disable) one of several trip signals. Defect originated as requirements error due to misunderstanding of actual system behavior (diverse backup would not help)	
Risk Significance:	CPC successfully performed its trip function, although in a delayed timeframe. Other RTS signals were still available to provide trip signals depending on transient (pressure, flux, etc.)	**RISK COLOR**

5
NON-1E EVENTS

Non-1E Event Overview

Out of 322 gathered reports of events or defects involving digital systems, 273 were related to Non-1E systems, which include plant computers, feedwater controls, electro-hydraulic control (EHC), and other systems (Figure 5-1). Out of these 273 reports, 196 described the presence of a single defect. The remaining 77 reports described the presence of a common defect in multiple redundancies.

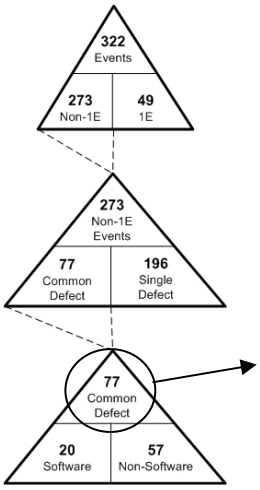


Figure 5-1
Non-1E Events

Effect	Software Events			Non Software Events		
	System	Subsystem	Channel	System	Subsystem	Channel
Single Failure	N/A	N/A	4	N/A	N/A	6
Spurious Actuation	1	---	1	7	1	1
Potential CCF	---	3	N/A	5	4	N/A
Actual CCF	7	4	N/A	26	7	N/A
	20			57		

Table 5-1
Non-1E Events with Common Defects (Breakdown by Type and Extent of Impact on System Function)

The non-1E events are reported separately from the 1E events. While the non-1E events often reveal important insights in regard to problematic issues for digital equipment, direct comparisons of 1E and non-1E data tend to be difficult and misleading from a D3 perspective. For example, 1E systems are subject to regulations that require redundant divisions to be separate and independent; while non-1E redundancies often share components (e.g. power supplies) that can cause redundancies to fail concurrently, so their CCF vulnerabilities are very different. These differences are discussed in greater detail below.

Figure 5-1 is constructed like Figure 4-1, around common defects, even though Non-1E systems are not designed with the independence that protect 1E systems from CCFs. Table 5-1 shows the

breakdown of the 77 non-1E common defect events by type and extent of the effects of the defect. Columns and rows are identical to those used in Table 4-1 for the 1E events. Table 5-1 shows that software was not a dominant contributor to non-1E events. Twenty of the seventy-seven events involved software problems. Seven of the thirty-eight potential or actual CCFs at the system level involved software. The numbers of non-1E events are significantly greater that those for 1E events, in part because there are more non-1E systems, and more operating experience, but also due to other factors, as discussed in the following sub-section.

1E vs. Non-1E Vulnerability to CCF

While insights regarding vulnerability to CCF causes can be obtained from the preceding figures and discussion points, it is difficult to directly compare the characteristics of 1E events to Non-1E events. First, there are many more installed digital Non-1E systems than 1E systems, thus increasing the number of events. However, the number of Non-1E events does not mean an equal proportion of 1E events should be expected as more 1E installations emerge.

A significant driver of Non-1E events is based on the nature of their designs. While reliability is a very strong design objective, complete independence between redundancies is often difficult to achieve, especially when the Balance of Plant control systems share resources (e.g., power supplies, backplanes, network segments, etc.) and are connected in the end to single control elements (e.g., feedwater regulating valves).

Table 5-2 lists some important differences between 1E systems and Non-1E systems. In addition to the larger installed Non-1E base, the design attributes of Non-1E systems can lead to events due to common defects more often than 1E systems.

From the D3 perspective, in every category, the differences between 1E and non-1E systems and practices are such that the 1E systems tend to be better protected against potential CCFs. This is why direct comparison of non-1E experience to 1E experience is problematic.

However, lessons learned from non-1E events are still valuable, and licensees are using the Operating Experience gained from Non-1E events to strengthen their digital system designs and associated engineering programs, as evidenced in Figure 5-3. Over 40 of the 77 Non-1E common defect events report some kind of program change, procedure change, or human performance enhancements and training as a corrective action.

Table 5-2
1E vs. Non-1E Design Characteristics

Design Attribute	1E Systems	Non-1E Systems
Redundancy	Independent Channels	Master/Slave
Shared Resources	Never	Almost Always
Signal Diversity	High	Low
Functional Diversity	High	Low
Formal SQA Methods	Always	Varies (Improving)
Functional Complexity	Low	High
System Interactions	Low	High

Non-1E Common Defect Causes

Referring to Figure 5-2, the three most reported causes of Non-1E Common Defect events are Incorrect Hardware Design, Incorrect Software Design, and Inadequate Testing. Hardware failure mechanisms include EMI induced disturbances, power supply transients and failures, signal drift, wiring and termination issues, and failed modules. Software failure mechanisms are described in more detail later in the document.

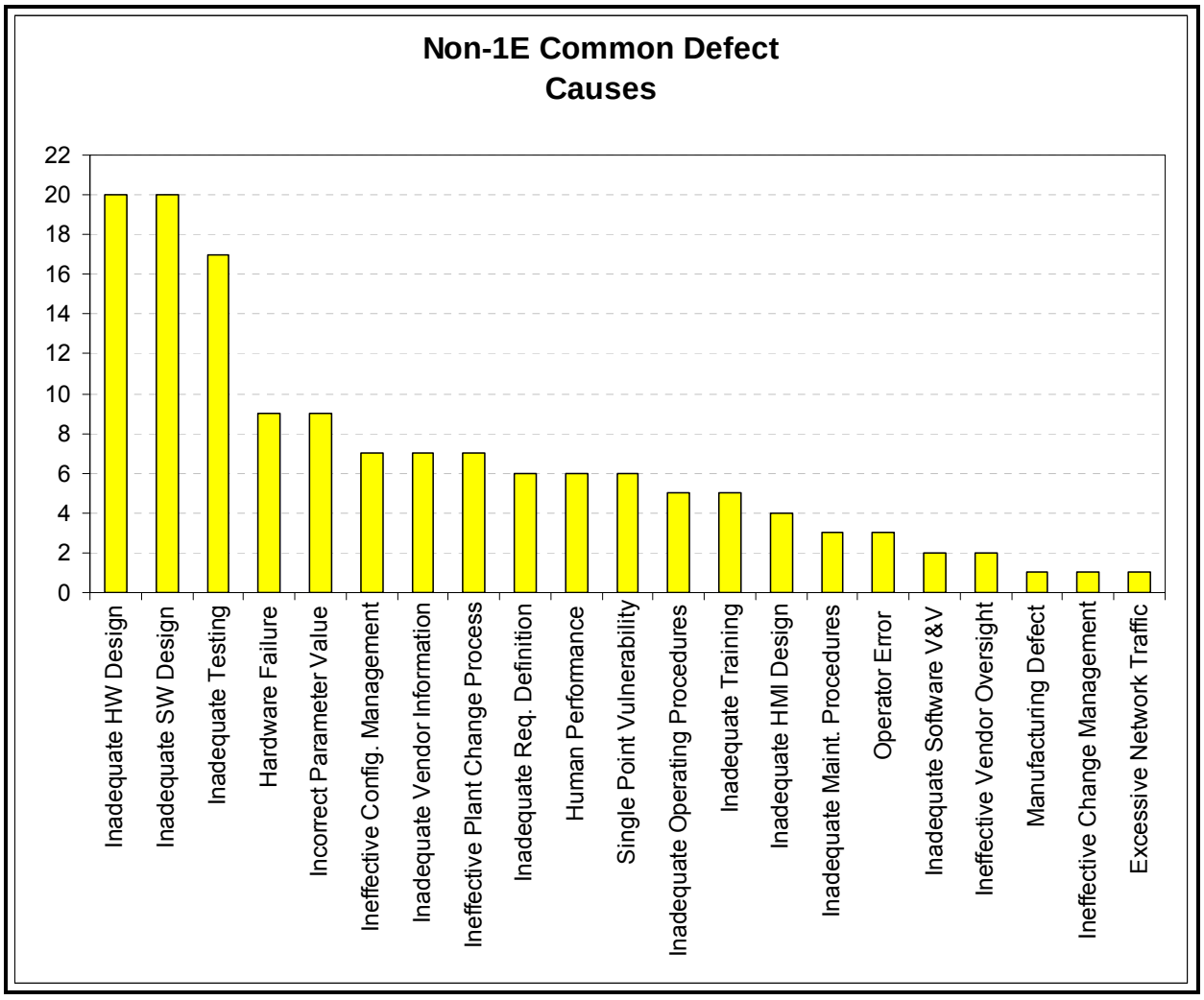


Figure 5-2
Non-1E Common Defect Causes

Non-1E Common Defect Corrective Actions

Figure 5-3 illustrates the corrective actions taken after discovery of common defects in Non-1E systems. Software Change appears in 25 event reports, while Inadequate Software Design appears in 20 events (Figure 5-2).

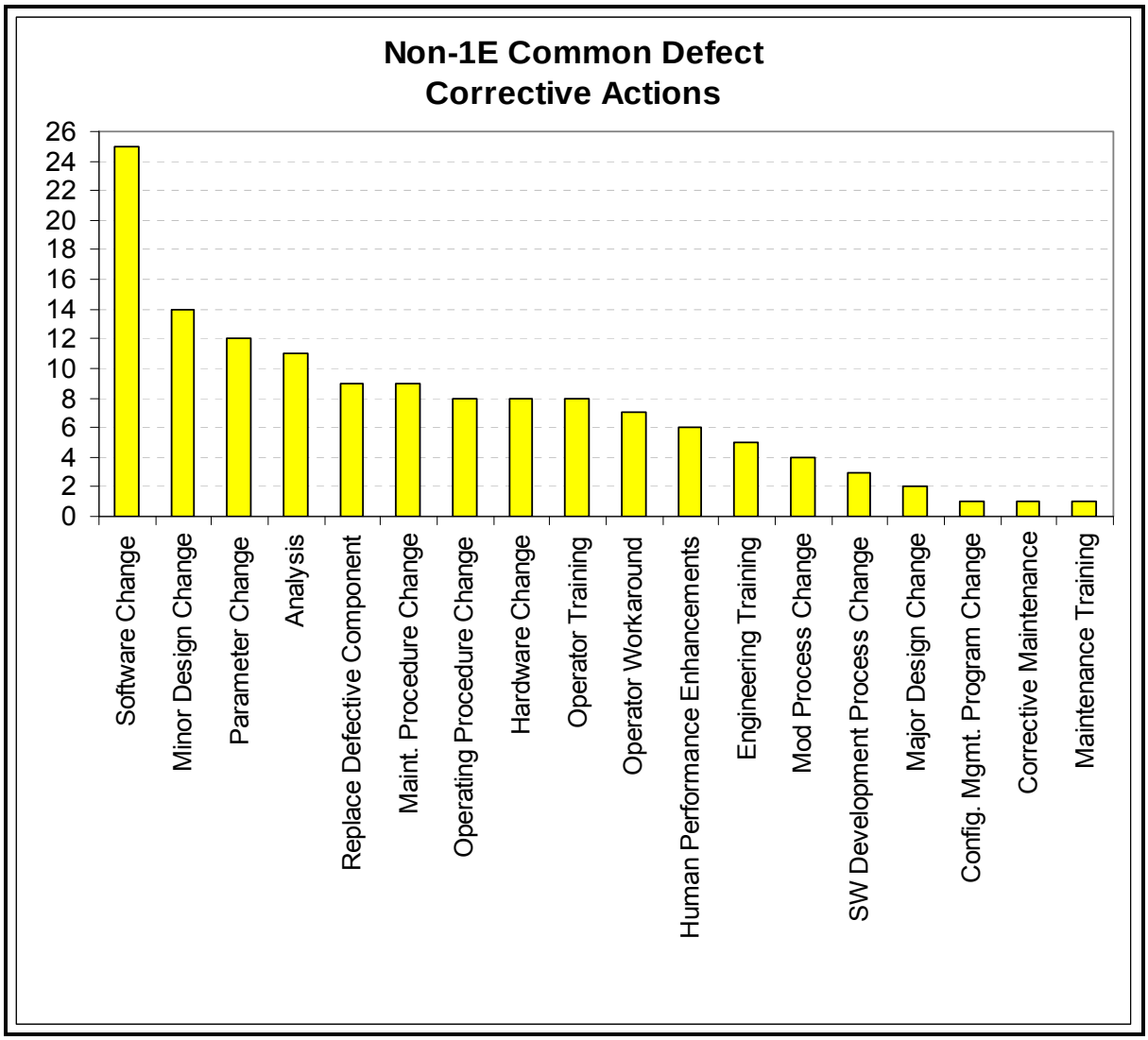


Figure 5-3
Non-1E Common Defect Corrective Actions

Non-1E Software Failure Modes

Figure 5-4 illustrates the failure mechanisms associated with the 20 software related events in Non-1E systems. The lighter bars in this figure are associated with software defects at the application level of the system. The darker bars are associated with defects in specific modules of the operating system software. The cross-hatched bar related to Buffer Overflows can be associated with the application software or the operating system. Further research is needed to make this determination.

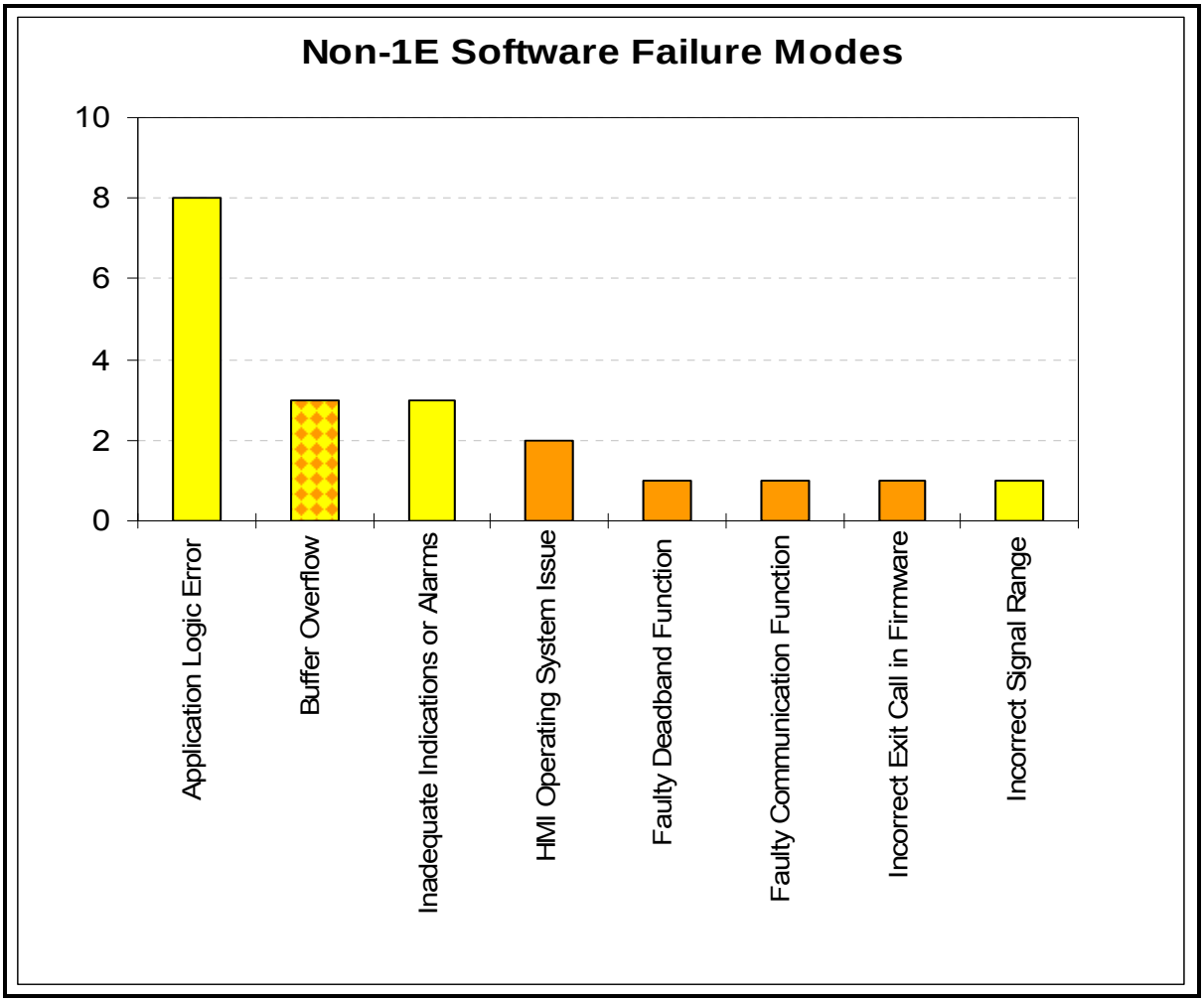


Figure 5-4
Non-1E Software Failure Mode

Selected Non-1E Common Defect Events

Table 5-3 summarizes the characteristics of a Loss of Annunciator System event that occurred in 2004. This Non-1E Software Event resulted from a buffer overflow problem triggered by two conditions: 1) when the duration between alarms exceeded 25 days; and 2) when a new alarm came in. When there were less than 25 days between alarms, the system remained fully

functional. After 25 days between alarms the system continued to scan points, but when one hit an alarm condition the overflow was triggered, and the system provided a "Loss of Annunciators" alarm.

This event illustrates a software design that did not anticipate the potential for a buffer overflow and the related triggering conditions. One of the software subroutines used Integer 4 math to access the lower 32 bits of a 64 bit real time clock. The corrective action was to hard code Integer 8 math into the calling program, and subsequent testing using time stamps up to 100 years showed no buffer overflow condition.

Given the availability of other instrumentation indicating plant conditions and the short time frame that the annunciators were out of service, this event had a relatively minor impact on risk.

Table 5-3
Actual System-Level CCF Event in a Non-1E System

	Non-1E Software Events		
Effect	System	Subsystem	Channel
Single Failure	N/A	N/A	4
Spurious Actuation	1	---	1
Potential CCF	---	3	N/A
Actual CCF	7	4	N/A
	20		

(See Table 4-1)

Event #	20	Event Date:	Dec-04	System:	Annunciator
Loss of Annunciator System					
	System	Subsystem	Channel	Root Cause:	Inadequate Software Design
Single Failure				Contributing Cause:	Inadequate Testing
Spurious Actuation				Contributing Cause:	---
Potential CCF				Corrective Action 1:	Software Change
Actual CCF	X			Corrective Action 2:	Software Development Process Change
Failure Mode:	Buffer overflow affected primary and backup annunciator system computers. Error in subroutine causes "Integer 4" math overflow under certain conditions.				
Risk Significance:	Annunciator system out of service 1 hour and 25 minutes.				RISK COLOR

This event also illustrates the effective use of a defensive measure whereby the system provided a “Loss of Annunciators” alarm that was independent of the buffer overflow design problem and the associated triggering conditions.

6
FINDINGS

The following list summarizes the insights and recommendations that were developed during this project based on evaluations of 322 nuclear plant operating experience reports describing digital system events between 1987 and 2007.

Observations

1. There were no Actual CCF events in 1E systems that disabled a safety function at the system level (Table 4-1).

2. The majority (18 of 27) common defect events in 1E systems resulted in subsystem or channel effects, leaving the balance of the system unaffected, and available to perform its overall safety function by other means using functional or signal diversity (Table 4-1).

3. Potential CCF events in 1E and Non-1E systems were dominated by non-software issues (Tables 4-1 and 5-1).

    - Of six system-level Potential CCFs in 1E systems, one involved a software design defect.

    - Lifecycle management and human performance issues were more prevalent, e.g., incorrect setpoints and parameters (Table 4-1).

4. Non-software issues made up the majority of both 1E and non-1E digital system events (Figures 4-2 and 5-2).

5. Software changes were often implemented as corrective actions for non-software problems (Figures 4-3 and 5-3).

6. There were several events that confirmed the effectiveness of signal and functional diversity in protecting against CCF (Table 4-3 for example).

7. None of the reports involved situations in which using diverse platforms would have been effective in protecting against potential or actual CCFs.

8. There were many events where defensive measures were deployed to prevent recurrence, and there were no repeat occurrences (Figures 4-3 and 5-3).

9. None of the evaluated Potential CCF events among 1E systems were safety significant (Tables 4-2 and 4-3 for example).

10. The event frequency over the 20 year span illustrated in Figure 3-1 generally follows the frequency of installations. Most of the reported defects (Figures 4-2 and 5-2) were discovered via testing, vendor reports or single random hardware failures within 1 to 2 fuel cycles after the digital systems were put into service.

Insights

1. Software has been no more problematic than other CCF contributors. Current methods have been effective in keeping software a minor contributor to Potential CCFs in 1E systems (Table 4-1). Important contributors to protection against CCF include:

    - Design and process features and characteristics that preclude, avoid or limit CCFs (independence, defensive measures, and inherent signal & functional diversity attributes, etc.)

    - Rigorous application of software codes and standards

2. In the data evaluated, comparison of the frequencies of software-related and non-software-related problems in 1E systems suggests that for 1E systems, protection against software failures and CCFs is already adequate at a "reasonable assurance" level.

Recommendations

1. Additional OE investigations should be performed, including risk significance analysis to clarify and/or reinforce the conclusions of this study.

    - Per Figure 2-1, source documents were found for 160 of the 342 events on the NRC List. A future joint task should be considered for NRC and Industry to obtain and review source documents for the remaining 182 events.

    - Source documents should be obtained from other countries and industries and evaluated for similar characteristics (confirm U.S. results).

2. Diversity and Defense-in-Depth guidance should be modified to endorse and credit methods that have proven effective in protecting against software CCFs in 1E systems.

3. Industry should continue to focus on prevention of hardware failures, which are usually the triggering condition for an event. At the same time, industry should continue applying lessons learned from the OE to improve software and human performance defensive measures.