



SPE-120584-PP

Safety and Reliability Incidents Caused by Software: How They Could Have Been Prevented

SPE Americas E&P Environmental and Safety Conference
23–25 March 2009 • San Antonio, Texas, USA

Don Shaler, Chief Technology Officer, Athens Group, Inc.

don.shaler@athensgroup.com

www.athensgroup.com

Software Kills!



A Safety Minute – 22 March 2009

Table 1: Number of Fatal Injuries among U.S. Oil and Gas Extraction Workers by Type of Injury Event, 2003-2007¹

Injury Event	Fatalities	% Total
Highway crash	151	28.7
Struck by object	109	20.7
Explosion	44	8.3
Caught or compressed in moving machinery or tools	41	7.8
Fall to lower level	36	6.8
Electric current	30	5.7
Fire	29	5.5
Aircraft crash	20	3.8
Other	66	12.5
Total	526	

¹Data for 2007 are preliminary.

Presentation Outline

- Examples of Software Related Incidents
 - Software you can “see”
 - Software you cannot “see”
- 3 Processes to Reduce Software Risk
 - Life Cycle Recognition
 - Configuration Management
 - FMECA



Software you can “see”



NE



Potentially Deadly Mishap

A driller was performing a test with a riser joint suspended 70 feet (21 meters) above the drill floor. Prior to leaving the drill cabin for a Job Risk Analysis meeting with the roughnecks, the driller selected “standby” mode on the drilling chair. While doing so, he inadvertently pressed the keypad button that activates Pipe Handling mode. In this mode, the drill control system sends a pressure monitoring command to the pipe elevator every 3 minutes. The driller stepped out onto the drill floor and three minutes later the pressure monitoring command was sent to the riser handling equipment which mistook it for an unlock command. The riser tool released the joint which fell through the well center and into the ocean. The joint fell perfectly through the slips and luckily, neither personal injury nor collateral equipment damage was experienced.

An FMECA of the equipment covering operational states and message flow could have prevented this incident.



Top Drive Out of Control

During the voyage to location, a technician was ‘tweaking’ the zone management parameters on a newbuild. A few minutes later the top drive started rotating by itself. The technician in his zeal to fix one thing had broken another – thereby introducing regression into the system. He was also unable to quickly recover to a previous known state as he wasn’t following software change control protocols. He and the team had to scramble to correct the issue. Fortunately there was no equipment damaged.

Following software change control and testing protocols would have prevented this.



Injured Rig Hands

Recently the elevators and bales of an older-model top drive reacted erratically to a rapid and erroneous user command. They swung around and injured two of the rig hands resulting in reportable LTIs. One finding of the investigation showed that a vendor had released a software patch to that model to prevent this erratic behavior, but somehow it had not been communicated or installed on that drilling unit. This example shows two things. First poor initial design and testing of the control software: the software interlock issue should have been discovered during its initial design and testing. Proper requirements gathering in addition to an FMECA would probably have identified the issue. Secondly it also demonstrated poor management of software as an asset on the MODU between the supplying vendor and the operator.

Software change control protocols would have avoided this incident.



Generator Trip

A vendor arrived onboard a rig having been officially requested to make changes to the rigs automation system. During their time onboard an unofficial request was made by a system operator regarding the numbering of main engine cooling system valves. The vendor either hadn't completely understood the request or had been distracted and unsuspectingly made the change to the wrong valve. Some time later a different operator attempted to give a close command to the valve in preparation for maintenance of the system. This resulted in the closing of the incorrect valve, ultimately causing a generator trip.

If formal control procedures had been adopted no unofficial change requests should have been carried out.



Control System Reset Kills 4

J-Lay pipe-laying operations on a large, offshore construction vessel were suddenly stopped when a control system failure occurred and hydraulic power was lost. During troubleshooting the two control units were restarted twice, unsuccessfully. A blinking red lamp on the PLC indicated that a memory reset was required even though a memory reset had NEVER been requested by control system diagnostics during equipment operations. As soon as the hydraulic power packs started, a loud bang was heard. A quadruple joint of pipe was released within the J-Lay tower and dropped approx. 1 meter to the welding deck below. A second quadruple joint of pipe in the pipe elevator was released (all clamps opened and the hydraulic safety stop swung away) and fell the full length of the tower, smashing through a crowded access platform to the deck below.

Eight personnel were injured - four fatally. All were located on the access platform and several were thrown overboard by the impact. The J-Lay tower initialization instruction was pre-loaded in PLC EPROM memory and the initialization included instructions to OPEN ALL CLAMPS.

This tragedy resulted from a poor understanding of control system software functionality and a lack of understanding of the operational state after a reset and restart is performed.



What Did We Learn?



What do the Authorities Say?

- **Functional safety of electrical/electronic/programmable electronic safety related systems (IEC 61508)**
 - To achieve software safety and reliability certain planning, design, analysis and verification activities must take place. The achievement should be measured throughout the life cycle based on a combination of product, process and competency. Specialized knowledge, skills and experience are required to obtain software safety and reliability.
- On July 5, 2005, at approximately 1200 hours, an unplanned riser disconnect occurred on the Ensco semi-submersible drilling unit 7500.
 - A description by the Operator of the station-keeping equipment and/or systems on board that support the rig classification and certification. Pre-qualifications of DP vessels should include in the Pre-hire verification a complete vessel Failure Mode & Effect Analysis (FMEA) to highlight known failure modes, complete and inclusive proving trials, and evidence of systematic yearly trials procedures based on the vessel's FMEA.

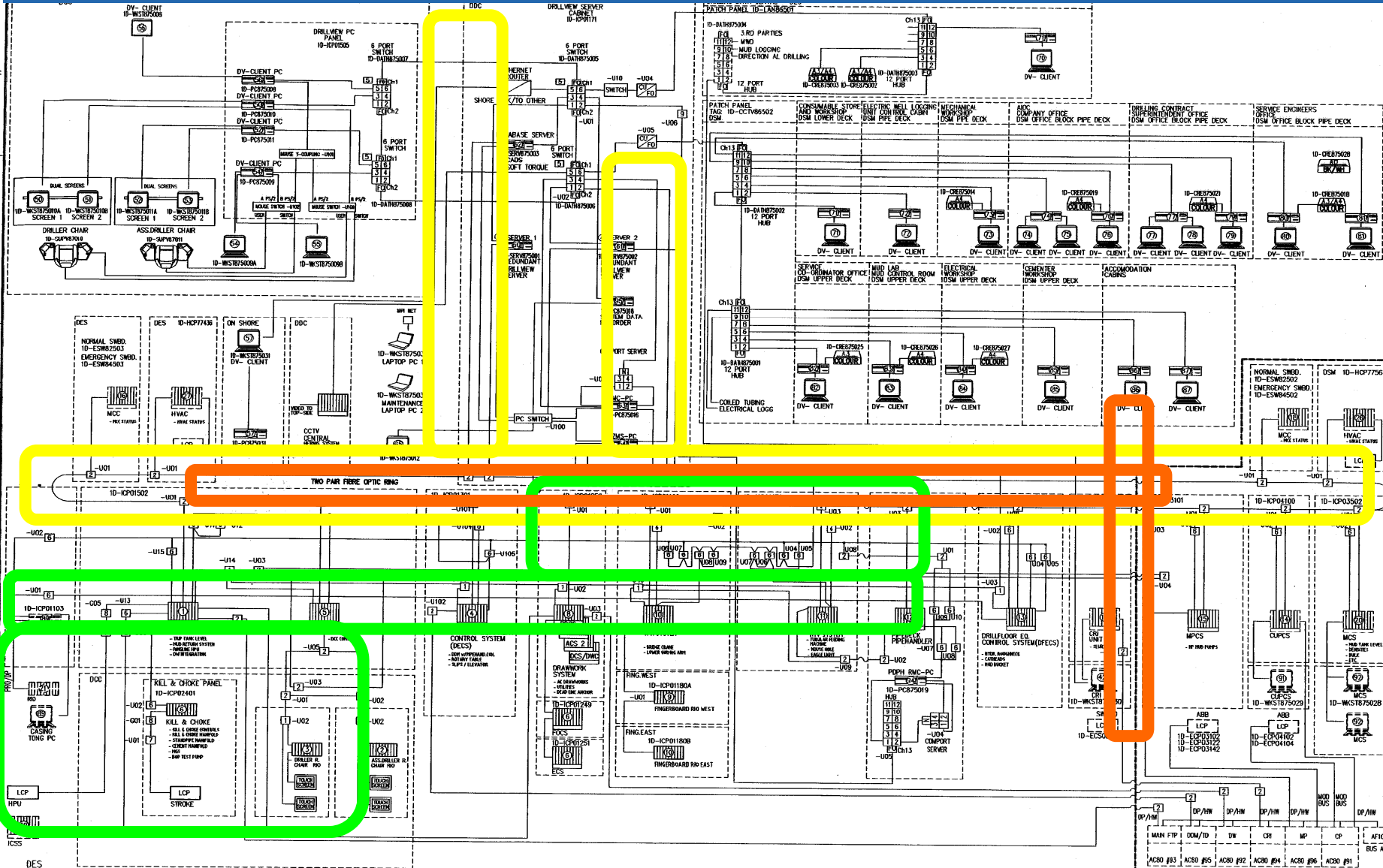


What does UK HSE Say?

- Process control systems are primarily implemented for economic reasons. However, those which are not considered safety related should still be designed, installed, operated and maintained so that their failure does not place a rate demand in the protective system which was not anticipated in its design. Part 1 of BS IEC 61508[43] provides guidance. The dangerous failure modes of the control system should be determined and taken into account in overall safety system specification. The control system should also be sufficiently independent of the safety systems.
- The control system may provide steady state or change of state (start-up, shutdown, batch) control functions. The latter may be implemented by automatic sequences or procedurally under manual control. Control systems should be implemented to provide stable control of the process under all expected normal and upset circumstances, including start-up and shutdown.
- The system should be designed to prevent or verify operator commands which might place a demand upon the protective system.
- The dangerous failure rate of the control system should be supported by operational experience of the system in a similar application, reliability analysis or reliability data from industry databases. The failure rate that may be claimed may not be less than 10^{-5} dangerous failures/hour.



Software you cannot "see"



Complexity is NOT your friend!



Your IT Network is Safe?

IT contractor indicted for sabotaging offshore rig management system, Company had refused to offer him a permanent job, feds say, March 18, 2009:

- **Mario Azar, 28 of Upland, Calif., was charged with illegally accessing and compromising a computer system used by Pacific Energy Resources Ltd. (PER) to monitor offshore platforms in California and Anchorage and to detect oil leaks. The indictment papers allege that Azar's actions affected the "integrity and availability" of the system and resulted in it becoming temporarily unavailable. Though no oil spill or environmental hazard occurred while the system was compromised, Azar's actions caused thousands of dollars in damage, the indictment said.**



Cyber criminals targeting energy and oil – 15 March 2009

- Based on an analysis of more than 240 billion requests for analysis by the company's corporate customers, that there was near 600% malware growth between like quarters in 2007 and 2008, and a 300% volume ratio increase from January 2008 through December 2008.
- A vertical industry analysis of malware growth found the energy and oil sector to rank in the top five targets in all threat categories. But energy and oil leads the pack by a long shot when it comes to one important category: encounters with unique new variants of data theft Trojans.
- With advances in the technology and sophistication of cyber attacks, malware delivered through the web can be remotely customized and configured once in place, based on the victim's identity.

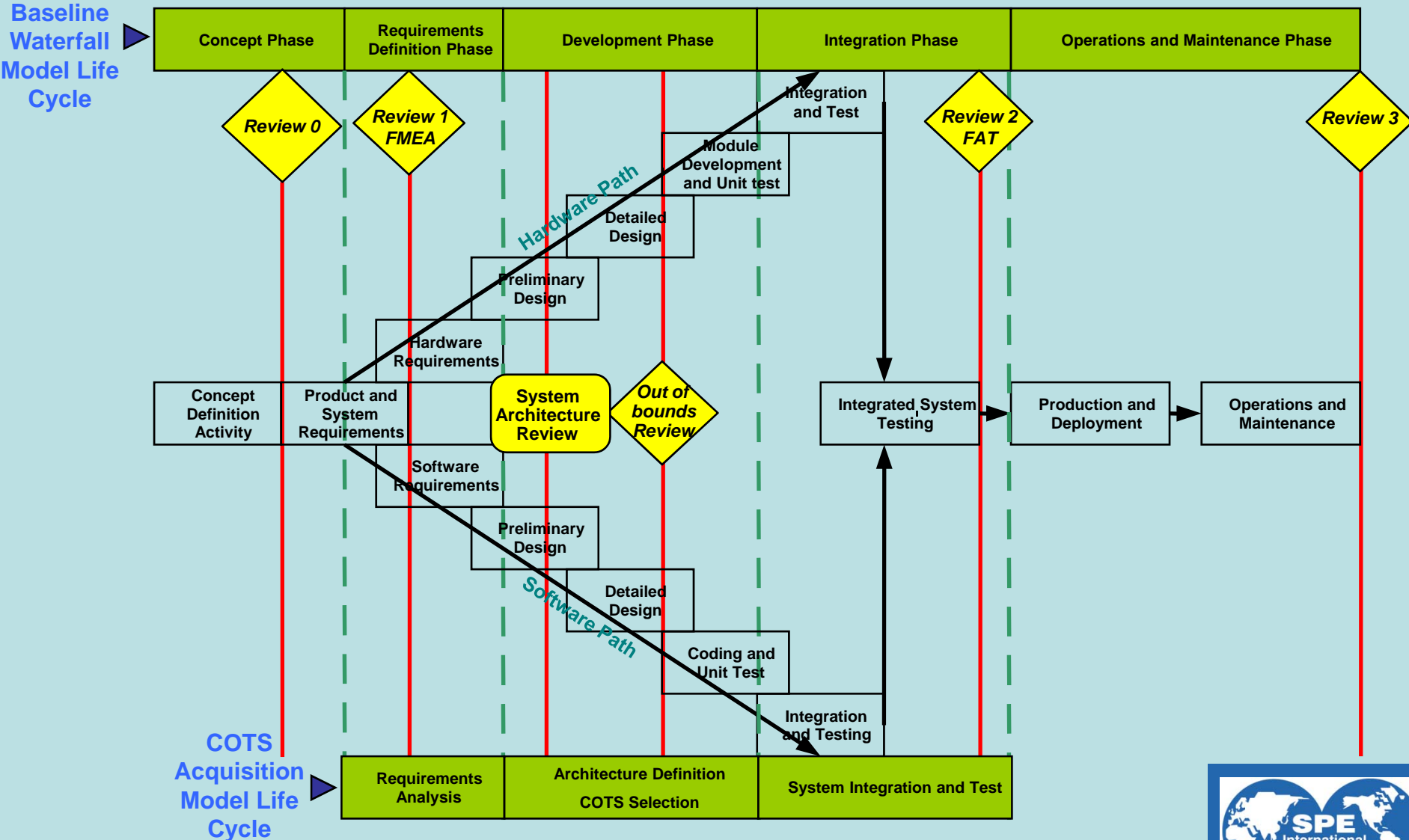


How can Software become Safer?

- Awareness of Development Life Cycle
- Software Configuration Management (SCM)
- Failure Mode and Effects Criticality Analysis (FMECA)



Life Cycle Model



Software Change Management

- Conduct a thorough assessment of software control procedures to identify inconsistencies and end-of-life exposure
- Inventory all software release levels and patches, and hardware parts, comparing them to applicable support-level agreements
- Maintain a central registry and verify each update against the previous version
- Understand all inconsistencies such as vendors not reporting system changes – and prioritize your risk
- Track all software changes



Failure Mode and Effects Criticality Analysis (FMECA)

Designed to identify potential failure modes for a system or process, to assess the risk associated with those failure modes, to rank the issues in terms of importance, and to identify and carry out corrective actions to address the most serious concerns.

In general, FMEA requires the identification of the following basic information:

- Item(s)
- Function(s)
- Failure(s)
- Effect(s) of failure
- Cause(s) of failure
- Actions to be taken in case of Failure
- Remediation Recommendations



FMECA Process Steps (1)

1. An offline analysis of the system identifies a list of components/functions whose failure/loss will be considered.
2. Experts are assembled who understand a) the design and support of the system and b) the operations the system is used to perform.
3. Then, for each component/function on the pre-assembled list:
 - a) the system experts familiar with that component score the likelihood of the failure occurring, and make clear to the operations experts the functional effect of that failure on the system
 - b) the operations experts consider what could/should be done to make the system safe for people and assets in the event of the failure; multiple operations scenarios may need to be considered, with a bias towards those more likely to involve safety-critical conditions
 - c) the operations experts consider what could/should be done to continue operations; multiple operations scenarios may need to be considered, with a bias towards those more likely to present fewer options for working around the lost function
 - d) the group scores the severity of the consequences
 - e) unless likelihood and/or severity are low enough that the item is not worth further attention, the group agrees on a recommendation for corrective action to be taken to reduce likelihood and/or severity, or to provide additional information if there are unanswered questions.
 - f) The assignment of the corrective action may be part of the recommendation, or be deferred. 18



FMECA Process Steps (2)

4. During the process, the group may decide to make additions and corrections to the pre-assembled list of components/functions.
5. Different failure modes are discussed and recorded only if they led to different safety and operational actions, or to significantly different occurrence likelihoods and severities.
6. Unless Risk Ranking is low enough that the item is not worth further attention, the group agrees on a recommendation for corrective action to be taken to reduce likelihood and/or severity, or to provide additional information if there are unanswered questions.
7. The assignment of the corrective action is usually part of the recommendation, or may have been deferred.
8. Failure causes were not explicitly discussed or recorded, although potential causes may have been explored during discussions of likelihood, severity, or mitigating actions. This is because the goal of this FMEA was to understand the frequency and impact of the failures, rather than the causes.
9. Results are compiled and distributed, and responsibility for tracking corrective actions to completion is assigned.



In Conclusion

You can make Software Safer

1. Awareness of Development Life Cycle
2. Software Configuration Management (SCM)
3. Failure Mode and Effects Criticality Analysis (FMECA)





Questions???

Don Shafer
Chief Technology Officer
Athens Group, Inc.
5608 Markcrest Drive, Suite 200
Austin, TX 78731
donshafer@ieee.org
www.athensgroup.com
512.345.0600 x117

**ATHENS
GROUP**
OIL & GAS

