Software Risk: Why must we keep learning from experience?

Don Shafer  
Chief Technology Officer  
Athens Group, Inc.  
4201 Southwest Freeway  
Houston, Tx 77027  
www.athensgroup.com
Presentation Outline

• “Ready” means “fit for purpose”
• Why do we care about software? – Cases of software failure.
• Where do we stand with O&G E&P software?
• Goals of Software Engineering
• Is there hope?
• Software Engineering Payback
• Questions
Why Software Failure

1. FAT, FMEA, FMECA of Software
2. In Production Software Maintenance
3. No Process
4. No Systems view

Computer science also differs from physics in that it is not actually a science. It does not study natural objects. Neither is it, as you might think, mathematics; although it does use mathematical reasoning pretty extensively. Rather, computer science is like engineering - it is all about getting something to do something, rather than just dealing with abstractions as in pre-Smith geology.

Richard Feynman, Feynman Lectures on Computation, 1996, pg xiii
This is “ready” software?

1) DP computer crashed. Determined to be calendar based program faults so it crashes once per month. A fix has been prepared but until we have an opportunity to install it we are just automatically rebooting periodically before they have a chance to crash. That keeps it on OUR schedule rather than as a worst possible moment experience.

2) DPO reports poor positioning during heading maneuver. Loss of position during 180 degree change was 25 meters. Frequent DGPS alarms. No bias, improper gain settings, frequent change of heading set point during turn, incorrect ocean current estimate.
There is a Process?

- Drive-off is caused by a fault in the thrusters and/or their control system and a fault in the position reference system. As for the drift-off incidents, the failure frequencies are derived from the DPVOA database, which consists of mainly DP Class 2 vessels. For the drive-off incident, these figures are representative also for the DP Class 3 vessels. The major difference between DP Class 2 and 3 is that only the latter is designed to withstand fire and flooding in any one compartment. These two incident types do not affect the drive-off frequency significantly due to the fact that drive-off is caused by logical failures in the thrusters, their control system and/or the position reference system. Hence, the DP Class 2 figures are used directly also for the DP Class 3 vessel.

- The frequency of fault in the thrusters and/or their control system can be obtained from data in Ref. /A-4/. The control system is here meant to include all relevant software and hardware. **The statistics show that the software failure is about four times as frequent as the hardware failure and slightly more frequent than the pure thruster failure.**

  - KONGSBERG OFFSHORE A.S.
    - DEMO 2000 - LIGHT WELL INTERVENTION
    - HSE EVALUATION
    - REPORT NO. 00-4002, REVISION NO. 00
IMCA Software Incidents as a Percentage of the Total

<table>
<thead>
<tr>
<th>Year</th>
<th>Total Software</th>
<th>Other Incidents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>1991</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1992</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1993</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1994</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1995</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1996</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1997</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1998</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>1999</td>
<td>20%</td>
<td>80%</td>
</tr>
<tr>
<td>2000</td>
<td>20%</td>
<td>80%</td>
</tr>
</tbody>
</table>

Average: 20% Software, 80% Other Incidents
Class 1: “Serious loss of position with serious actual or potential consequences.”
Where is the Systems Engineering?
Where do we stand with O&G E&P Software? (1)

- Many vendors, multiple networks, no systems/software engineering
- Incomplete, but incorrect, requirement sets
- No enforced standards that all purchased components, including embedded Programmable Logic Computers, can communicate through existing, defined interfaces
- Network topologies and protocols are incompatible
- Single points of failure in the software and network subsystems are NOT identified
- Early estimation of the network data throughputs and maximum loading scenarios is either lacking or inadequate
Where do we stand with O&G E&P Software? (2)

- No early identification of overly complex control system designs that lead to impossible to test operational domains
- No early estimates of the software development timeline and impacts on commissioning
- No system models or standard equipment definitions against which to build a library of FAT, commissioning and regression test scenarios
- No system or equipment models against which to rationally conduct software and network FMEAs
- No baseline for comprehensive make versus buy decisions and the next system’s contract
- No life cycle model for system upgrade or work-over
# Software Risk: Why must we keep learning from experience?

## Is there hope?

- **Semiconductor Fabrication Facility Model**

<table>
<thead>
<tr>
<th></th>
<th>Offshore Drilling Rig</th>
<th>Semiconductor Fab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software Fails – People Die</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Drills Wells</strong></td>
<td>Yes - $10 \times 10^3$ meters</td>
<td>Yes - $2 \times 10^{-7}$ meters</td>
</tr>
<tr>
<td><strong>Production Environment</strong></td>
<td>Flying mud, 12 hour shifts, steel-toed boots</td>
<td>Clean Room, Positive Pressure, bunny suits</td>
</tr>
<tr>
<td><strong>Cost of Software Failure</strong></td>
<td>$100,000s/day</td>
<td>$10,000,000s/day</td>
</tr>
<tr>
<td><strong>Automation Software</strong></td>
<td>Plug and Prey</td>
<td>Plug and Play</td>
</tr>
<tr>
<td><strong>Software Quality</strong></td>
<td>No effort to measure</td>
<td>3.4 errors/million</td>
</tr>
<tr>
<td><strong>Equipment Standards</strong></td>
<td>No, not POSC, WITS nor WITSML</td>
<td>SEMI Standards begun in early 1970s</td>
</tr>
<tr>
<td><strong>Software Process Model</strong></td>
<td>NONE</td>
<td>SEI/CMMI</td>
</tr>
<tr>
<td><strong>Years to Improve Quality</strong></td>
<td>UNKNOWN</td>
<td>15</td>
</tr>
</tbody>
</table>
Yes, It is a Cyclical Business!
Systems Engineering Life Cycle Model

Areas likely needing software engineering improvements and risk mitigation
Payback in Software Engineering

Amount of Risk

With Software Engineering

Dollar Value of the Project

Highest Project Planning Impacts
Highest Project Execution Impacts
Highest Production Impacts

Concept Exploration | System Exploration | Requirements | Design | Implement | Installation | Operations & Support | Maintenance | Retirement

Without Simulation

Without Software Engineering

With Software Engineering
Cost of Software Quality

IBM Cost Data

Boehm Cost Data
Cost of Platform Down Time

- **Average Daily Rate for 5th Generation Drilling Rigs:**
  - $154,022* per day, ~$77,000 per 12 hour shift
  - In startup, could be down 10 shifts per week at a cost of $770,000 per week

- **Production platforms pump 20,000 bbl per day**
  - Cost of Nymex Crude $41.25** per barrel
  - Production platform down for 21 days cost $17,325,000

- **Cost of doing a front end software and network Verification and Validation:**
  - < $100,000

* Updated at 1330hrs, 16 July 2004, [www.rigzone.com](http://www.rigzone.com)

** Updated at 1330hrs, 16 July 2004, [www.bloomberg.com](http://www.bloomberg.com)
Measure and Simulate the Control Software

- Distance (inches)
- Time (seconds)
- Speed (inches/second)
Collision Timing Results - 1st Pass
Collision Timing Results - 2\textsuperscript{nd} Pass
Software Engineering Goals Checklist

☑ Complete and correct set of requirements have been collected
☑ Requirements of all purchased components that include embedded Programmable Logic Computers (PLCs) can communicate with each other through existing, defined interfaces
☑ Network topologies and protocols are compatible
☑ Early identification of single points of failure in the software and network subsystems
☑ Early estimation of the network data throughputs and maximum loading scenarios
☑ Early identification of overly complex control system designs that lead to impossible to test operational domains
☑ Early estimates of the software development timeline and impacts on commissioning
☑ Development of a model system against which to build a library of FAT, commissioning and regression test scenarios
☑ Providing a model against which to rationally conduct software and network FMEAs
☑ Provide a baseline for comprehensive make versus buy decisions and the next system’s contract
Questions???
Is there software risk in the dynamic positioning regime? Cited by Kongsberg Offshore A.S. in HSE Evaluation Report No. 00-4002: “The statistics show that the software failure is about four times as frequent as the hardware failure and slightly more frequent than the pure thruster failure.” Based on IMCA data, the percentage of “Loss of Position Class 1” DP problems that were caused by software for a recent 5 year period was 33%. Can we mitigate the risk? FMEA, FMECA and good software engineering practices will go a long way toward reducing today’s DP software risks. This is not rocket science but the lack of good engineering practices. The Airbus 300 series and the latest Boeing 7x7 aircraft are completely fly by wire. The airbags in your automobile have autonomous processors with embedded software. Embedded medical devices contain processors run by software. We would never tolerate the number of software failures in these devices that occur on DP systems. Why don’t they fail at a 33% rate? This presentation will propose a set of software and hardware life cycle processes along with a mitigation model for identifying and eliminating software risk within DP systems. Several recent incidents will be analyzed to show how these processes would have mitigated the potential for failure. Attendees will be able to take away processes they can implement in their own organizations to reduce software failures.
Don Shafer is a co-founder, corporate director and Chief Technology Officer of Athens Group, Inc. Incorporated in June 1998, Athens Group is an employee-owned consulting firm, integrating technology strategy and software solutions. Since 2001, he has led Athens engineers in the analysis, verification, validation and simulation of drilling, safety and positioning systems for deep water oil platforms. Prior to that he worked on the development of seismic databases, down hole logging equipment design, SCADA system design and production control. His latest patents are on joint work done with Agilent Technologies in state-based machine control. He earned a BS degree from the USAF Academy and an MBA from the University of Denver. Shafer’s work experience includes positions held at Boeing and Los Alamos National Laboratories. He is currently an adjunct professor in graduate software engineering at Texas State University, a Senior Member of the IEEE and the Editor in Chief of the IEEE Computer Society Press. His latest 2004 presentations on oil industry software engineering are for the Society of Petroleum Engineers Leading-Edge Drilling Rigs Advanced Technology Workshop and the Marine Technology Society Dynamic Positioning Conference. With two other colleagues in 2002, he wrote Quality Software Project Management for Prentice-Hall now used in both industry and academia. He is a multiple book author for the IEEE Software Engineering Series.