The consequences of not managing software failure risks

Annie Combelles, COO DNV IT Global Services
ICST 2008, April 10
Agenda

- DNV & Risk Management
- Systems and Software challenges
- Existing tools: standards and techniques
- What the industrial community needs
New risk reality

- Companies today are operating in an increasingly more global, complex and demanding risk environment with “zero tolerance” for failure

- Climate change
- Increased demands for transparency and business sustainability
- Stricter regulatory requirements
- Increasing IT vulnerability
More than 140 years of managing risk

- Det Norske Veritas (DNV) was established in 1864 in Norway

Managing risk

Maritime
- Ship classification
- Certification of materials and components
- Assessments and solutions
- Fuel testing
- Training

Energy
- Risk management consulting
- Qualification and verification
- Offshore classification
- Laboratory services
- Training

Other prioritised industries
- Automotive
- Defence
- Finance
- Food and Beverage
- Health care
- ICT and telecom
- Public sector
- Transportation
Why bother with software?
- 800 passengers
- Completely dependent on programmable systems
- No longer ‘Fly by wire’ – based on electrical signals
- Over one million source lines on-board
Incident on an Airbus A319

 Shortly after a night departure from London Heathrow in October, a British Airways Airbus A319-131 on a flight to Budapest was approaching FL200 in clear weather when there was an audible ‘clunk’ and the flight deck became dark with a number of electrical systems and flight displays lost.

 Displays lost included the captain's and first officer's primary flight and navigation displays and one of 2 electronic centralized aircraft monitor (ECAM) displays, leaving only 1 ECAM; also lost were the autopilot, autothrust, intercom and most flight deck lighting, including that of overhead and pedestal panels.

 The captain took manual control and maintained the aircraft attitude by reference to the external night horizon, the standby horizon and those windows available. It was only to see it is probable that the electrically-driven standby horizon was not powered or lighted but might have remained useable for 5 minutes due to the gyro's inertia.

 The captain transmitted a MAYDAY on VHF but this was not received by ATC because the radio was no longer powered. The first officer carried out the ECAM actions and the primary flight instruments and most other systems were restored. The aircraft was in the degraded condition for a period of about 2 minutes.

 Communication with ATC was regained and the crew were allocated a holding pattern while they reviewed the status of the aircraft. A number of systems remained inoperative including the weather radar display and window heat. The aircraft was in the hold for some 40 minutes, during which the flight was continued to Budapest, where the first officer performed the landing. During the final approach, the flight crew noted a thrust reverser caution. After landing, all the remaining affected systems were successfully reset by a maintenance engineer and the aircraft continued in operation for 6 days with no further electrical failures.

 The incident was reported under the UK Mandate. Observers noted all and the aircraft was then taken out of service for investigation under AAB supervision. The aircraft was returned to service and has continued in operation without any further repeat of similar malfunctions. The AAB is investigating this with the cooperation of the manufacturer, operators and the operator. A further report will be published.

 The AAB Special Bulletin 52/2005 was issued on 25 November and can be downloaded from the AAB.


 Article Id: 162872
 Release date: 25 Nov 2005
Car Development

- Approximately 50-70 % of development cost is SW related
- SW based control of brakes, steering, anti-spin, air-bags etc.
Car industry: Is There a Problem?

- Software content of automobiles increasing
  - Volvo claims that 80% of the cost of one car is software related

- Substantial opportunity for failure
  - 257 recent recalls involved software (out of 7,000 analyzed)

- Industry continues to face price, quality, and cycle time pressure

- Development as well as manufacturing processes must be addressed

- Improvement is essential to survival
Failures in Automotive

The incident: a de-synchronization of the valvetronic motors may occur. If this happens, the engine could stall and the driver will not be able to restart the engine. Depending on the level of engine roughness, as well as traffic conditions and the driver’s reaction, this could lead to a crash.

The incident: some 30 car crashes over six months are alleged to have been caused by faulty cruise control. The allegation is that cars become locked into a certain speed and the cruise control does not disengage, resulting in cars crashing at full speed into motorway toll booths and other fixed objects.
Fjellstrand: 35 m FoilCat

The boat makes 50 knots...

The captain may have **20-30 screens** within reach, belonging to different SW based systems.
Failures at sea

The incident: the two GPS systems simultaneously changed “correct” position by 70 m. The software did not detect this failure, and consequently the DP system changed the rig position by 70 m.

The incident: probably it was a fault in the embedded software of the communication card which led the card to send out erroneous output signals. Same problem with all the cards of the same type, but the card failed infrequently so it was difficult to observe if and locate the fault.

The incident: the software should block requests for valve opening if the valve was 100 % open. The problem was that the valve was leaky and gradually closed by itself, so the status data in the software regarding valve opening did not reflect the reality. Therefore the software from time to time blocked requests for the opening of the valve, even though the valve was not 100 % open.
Why do projects fail to meet their target?

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting
- **Unmanaged risks**
  - Poor communication among customers, developers and users
  - Use of immature technology
  - Inability to handle the project’s complexity
  - Sloppy development practices
  - Stakeholders politics
  - Commercial pressures

According to Robert Charrette, IEEE Spectrum
Class notation: how to build trust

- Requirements for the total drilling plant
- International standard
- Trouble shooting failure investigation
- Third party Quality assurance
- World global organisation
- Traceability of documents
- Reduced downtime
- Certification of systems and components
- Technical expertise and commitment
- Survey in operation
- Increased level of safety
- Fewer question asked by authorities
- Commissioning and system integration testing
- Final product assessment (FAT) and Product Certification PC
- Survey during fabrication
- Approval of design
- DRILL Class
- OCTOBER 2006

OFFSHORE STANDARD
DNV-OS-E101

DRILLING PLANT

DETNORSKEVERITAS
HOW to verify a control and monitoring system?

- SW Code verification
- Analysis (e.g. FMEA/FMD)
- Plan approval

Other methods?
- Qualification / personnel certification?
- Advanced testing – simulation – HIL?
Standards & Guidelines

- DO-178B/ED-12B: Aeronautics (both civilian and military)
  - Applicable to s/w-based systems, provides guidance on applying FAA (and EUROCAE) standards
  - Defines five levels from A (highest) to E (lowest)

- ISO 61508

- ISO 17894 for programmable electronic systems in Marine applications (derived from 61508)
  - To address safety and dependability issues for On Board Marine Programmable Electronic Systems (PES)
  - Principle-Based (Mandatory)
  - Recommended Criteria

- EN-50128 (EN-50129): Railway industry standard

- ISO WD 26262: Automotive (WD = Working Draft)
  - Derived from 61508
  - Aims at simplifying and reducing costs.

- FDA 501(k): Food & Drug Administration Section 501(k), for medical devices, similar to FAA requirements

- Safety & Security Application Area
  - Defined by the FAA for the guidance of Safety & Security Process Assessment and Improvement

- Other Safety and Security Standards
  - MISRA C Rules (for the implementation of C s/w in Automotive
  - BS
The root standard for functional safety is the ISO IEC 61508:

- For Programmable Electronic Safety-Related Systems (PES)
- Defines functional safety
- Defines Integrity levels: the likelihood of a safety function being performed satisfactorily, from SIL 1 to SIL 4
- Risk-based approach
- Generic approach to be adapted by each industry or organisation

- **Part 1**: General requirements
- **Part 2**: Requirements for electrical/electronic/programmable electronic safety-related systems
- **Part 3**: Software requirements
- **Part 4**: Definitions and abbreviations
- **Part 5**: Examples of methods for the determination of safety integrity levels
- **Part 6**: Guidelines on the application of IEC 61508-2 and IEC 61508-3
- **Part 7**: Overview of techniques and measures
1. The PES shall be free from unacceptable risk of harm to persons or the environment.
2. In the event of failure, the PES shall remain in or revert to the least hazardous condition.
3. The PES shall provide functions which meet user needs.
4. Functions shall be appropriately allocated between users and PES.
5. The PES shall be tolerant of faults and input errors.
6. The PES shall maintain specified levels of accuracy, timeliness and resource utilization when used under specified operational and environmental conditions.
7. Unauthorized access to the PES shall be prevented.
8. The PES shall be acceptable to the user and support effective and efficient operation under specified conditions.
9. The operation of the PES shall be consistent and shall correspond to user expectations of the underlying process.
10. The interaction between the PES and the user shall be controllable by the user.
11. The PES shall support proper installation and maintenance, including repair and modification.
17894 Lifecycle principles

12. All PES life cycle activities shall be planned and structured in a systematic manner.

13. The required level of safety shall be realized by appropriate activities throughout the life cycle.

14. Human-centered activities shall be employed throughout the life cycle.

15. Verification and validation activities shall be employed throughout the life cycle.

16. All parties involved in life cycle activities shall have and use a quality management system.

17. Existing requirements for marine systems shall be taken into account throughout the life cycle.

18. Suitable documentation shall be produced to ensure that all PES life cycle activities can be performed effectively.

19. Persons who have responsibilities for any life cycle activities shall be competent to discharge those responsibilities.

20. The PES configuration shall be identified and controlled throughout the life cycle.
A Maturity Model for Software Product - 1

- A way to express the increase of confidence in RAM
  - How to determine if the system corresponds to a given specification and does not contain internal faults
  - Assess the « maturity » of interim & final products

- Product Areas are the main deliverables of the development phases
  - Context Analysis, User requirements, High level design, Detailed design, ....
  - Risk assessment is not a separate Product Area

- Maturity levels
  - 1: Initial
  - 2: Standardized (GG1>=1 & GG2>=1, GG3 = 0, GG4 = 0)
  - 3: Correct
  - 4: Consistent
  - 5: Verified

Based on work at Laboratory for Quality Software, Eindhoven University
Specific Goals hold for one Product Area only; they are implemented by one or more specific properties for Product Area
- Ex: User requirements
  - SG1: Complete
  - SG3: Correct
- SG2: Uniform
  - SG4: Consistent

Generic Goals
- Complete
- Uniform
- Correct (within elements)
- Consistent (between elements)

<table>
<thead>
<tr>
<th>GG1</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Some required elements are missing</td>
</tr>
<tr>
<td>1</td>
<td>All required elements are present</td>
</tr>
<tr>
<td>2</td>
<td>Semi-formal elements have been added</td>
</tr>
<tr>
<td>3</td>
<td>Formal elements have been added</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GG2</th>
<th>Uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No standardization</td>
</tr>
<tr>
<td>1</td>
<td>Within the product</td>
</tr>
<tr>
<td>2</td>
<td>Compliance to a company std</td>
</tr>
<tr>
<td>3</td>
<td>Compliance to an industry std</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GG3</th>
<th>Correct (within elements)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Faults are detected</td>
</tr>
<tr>
<td>1</td>
<td>Manual review/testing has not detected faults</td>
</tr>
<tr>
<td>2</td>
<td>Automated testing has not detected faults</td>
</tr>
<tr>
<td>3</td>
<td>Formal verification has not detected faults</td>
</tr>
</tbody>
</table>

Based on work at Laboratory for Quality Software, Eindhoven University
MPQA at DNV

A tool to measure a manufacturer’s ability to control product quality by using a rating assessment methodology.

A MPQA will quantitatively document the operation’s “capability level” measured against specific criteria.
### MPQA modalities

- **12 elements with sub-elements and questions**

1. Quality System Documentation
2. Human Resources
3. Internal support functions
4. Capability Review
5. Design and development
6. Purchasing and Subcontracting
7. Production Process Control
8. Distribution and Warehousing
9. After Sales Service
10. Quality Inspection and Testing
11. Document Control
12. Non-conformities and Corrective Actions

#### MPQA Summary Score Sheet

<table>
<thead>
<tr>
<th>Supplier of critical component</th>
<th>Contract No. and requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

|                                | Possible points | Points scored | % achievement |
|                                |-----------------|---------------|---------------|
| 1 Quality system documentation | 109             | 100           | 92            |
| 2 Human resources              | 84              | 80            | 95            |
| 3 Internal support functions   |                 |               |               |
| 4 Capability review            | 211             | 200           | 95            |
| 5 Design and development       |                 |               |               |
| 6 Purchasing and subcontracting| 73              | 60            | 82            |
| 7 Production process control   | 458             | 350           | 76            |
| 8 Distribution and warehousing | 152             | 130           | 86            |
| 9 After sales service          |                 |               |               |
| 10 Quality inspection and testing | 350       | 300           | 86            |
| 11 Document control            | 176             | 90            | 51            |
| 12 Nonconformities and corrective actions | 143       | 120           | 84            |
| **Total score**                | **1756**        | **1430**      | **81**        |

Acceptance criteria for level 4:
- Minimum average score: 84
- Minimum rating on any elements: 65

Clear all input
The Problem to solve
Failure Mechanisms

« Buying software is sadly similar to buying food from a street vendor! »

J. Voas

After all it is possible that the software:

- Has been tested at minimum and/or inadequately
- Was developed according to ad hoc processes
- Has high defect density
- Does not match the functionality in the marketing brochure
- Is used incorrectly

Test Quality Rating:

a measure of dependability?
What is applicable?

- Fault Tolerance
- Fault Prevention
- Initial defensive mechanism
- Fault Removal
- Protective mechanism
- Fault/Failure Forecasting
- Reliability Modelling
Techniques and Technologies

- Requirements development
  - Elicit needs
  - Identify interface requirements
  - Validate requirements

- Architecture & design
  - Develop alternative solutions
  - Develop the design

- Verification
  - Establish the environment & procedures

- Validation
  - Establish environment & procedures

Fault Removal
Fault Prevention
Protective mechanism
Initial defensive mechanism
Requirements challenges

- Specifications describe functional requirements in varying levels of detail. Many details are left to be defined in the system descriptions. Some are omitted which results in discussions and changes about functionality during later phases.

- Although operational modes are described, scenarios are generally not described in system descriptions, for S/W.

- Traceability mapping between detailed requirements within system description and FAT program is not explicitly made.

- Updated requirements are not systematically reintroduced in the documents.

- Lessons learned from previous projects are described to new suppliers, to avoid having the same surprises again. However applicable to what went wrong, and not what was successful which stays implicit.
Architecture challenges

- Lack of architects
- Interfaces between components insufficiently analyzed and documented
- COTS – Commercial Off The Shelf products and open source components
- Too often, no alternative scenarios considered
- Absence of architecture validation
- Very few “lessons learned”
- In case of suppliers/subcontractors, architecture and design produced not reviewed
Social & Human factors

- **Transfer of responsibilities**
  - From ‘mechanical manufacturers’ to control system suppliers
  - Safety functions - from mechanical devices to parameters
  - Competence – different professionals responsible

- **Systems are not completed at delivery**
  - Extensive implementation after installation

- **Change handling – Configuration Management (CM)**
  - Modifications – upgrading – new revisions during installation and operation
Verification and Validation

- Lack of structured strategies for verification
  - Too often, tests suites depend on engineers expertise
- Absence of traceability between requirements and tests
  - Coverage measurements of S/W requirements to FAT is generally not performed
- Documents are not part of verification
- Verifications are often squeezed because lack of time or cost constraints
- Regression testing is very limited and does not learn from previous mistakes
- The non-functional requirements for S/W (maintainability) are generally not tested or accepted in the FAT program

Fault Removal
Fault Prevention
Protective mechanism
Initial defensive mechanism
Software Reliability Engineering

Reliability Modelling

- SRE has not fully delivered its promise
  - > 100 models & publications

- Most reliability models ignore the development process

- Observations during testing not directly extensible for operational use

- Too few data to apply statistical predictions

- Some assumptions not realistic
  - Faults are independent
  - Each fault has the same chance to be detected
  - Corrections never introduce new faults
What Industries need: Recommended Practices

- **Owner**
  - Includes the RP as contract addition with the SI to ensure good practices are followed by SI
  - Controls that the RP are followed by SI OR mandates a third party to control that RP are followed by SI

- **System Integrator**
  - Follows the RP included in the contract to perform its tasks
  - Includes the RP in the contracts established with component suppliers
  - Controls that RP are followed by Subs (or mandates a 3rd party to do so)

- **Supplier**
  - Follows the RP included in the contract to perform its tasks

- **Operator**
  - Follows the RP included in the contract to perform its tasks

- **Verifier**
  - Controls if RP are followed by SI, SU,OP (depending of its mandate)
Conclusion

■ The challenges have still to be addressed
  - How good/safe is the system/software, quantitatively?

■ Some industries are faster movers than others but none is fully mature

■ It is not adequate to focus at one technique only

■ In order to build trust in systems and software, we have to
  - Take a risk oriented approach
  - Share knowledge and get best practices in place