ISO/IEC JTC 1/SC 22/OWGV N 0041

Paul Caseley, "Dependable software dependent systems?" presentation at Meeting #2

Date	2007-04-04
Contributed by	Paul Caseley
Original file name	Vulnerabilities release version and reviewed v2 release comments incorporated.pdf
Notes	

Dependable software dependent systems?

Paul Caseley

Dstl Fellow

15 September 2006

prcaseley@dstl.gov.uk

+44 1684 77 1476

Caveat

Some of the slides contain my views on MOD policy, some slides are my summary of policy, but are NOT the MOD policy

Overview

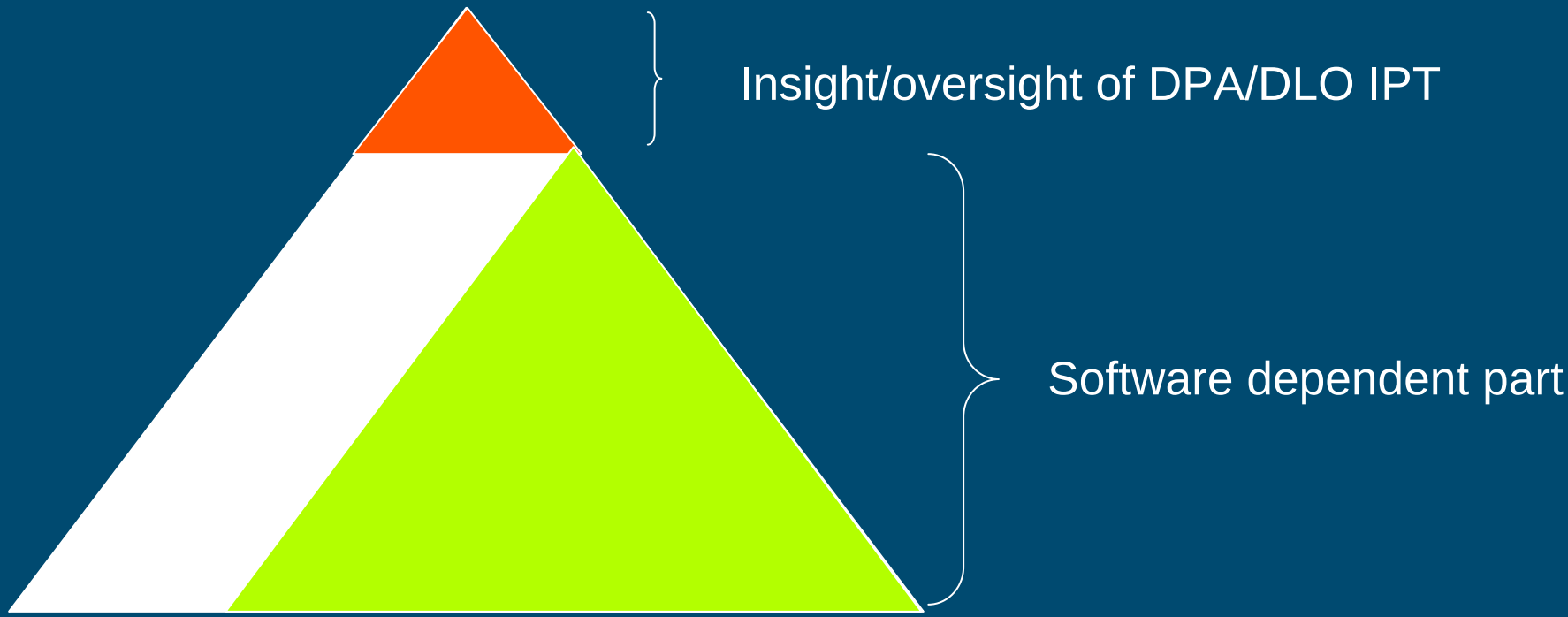
- Software the problem
- Safety – an example
- Evidenced based software safety (new Def Stan)

The Software Problem

The General Problem - Its expensive

- US senate report March 2004 by the General Accounting Office on software-intensive weapon acquisitions found:
  - For 2003 DoD estimated that they spent 40% of R&D budget on software ($21bn) of which $8bn was rework because of quality-related issues (waste)
  - "*to ensure DoD has the knowledge it needs to oversee software-intensive acquisitions, we recommend that acquirers require software contractors to collect and report metrics*"
  - Software issues were influential in the decision to cancel the Comanche programme.
- Similar problems exist in UK Government projects

A view on a typical MOD System



Safety problems

Current Requirement (Safety)

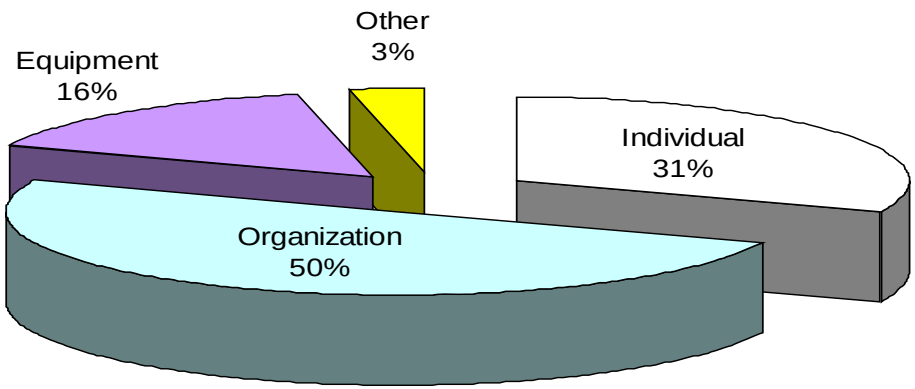
- MOD's approach to safety is set in the Secretary of State's policy statement on Safety, Health and Environment protection. This policy is reviewed regularly and can be summarised as follows:
- MOD are self regulating but MOD should be at least as good as the civil domain

3 Significant Military Safety Research Challenges

- Certification of AI for autonomous vehicles
- Certification of Network Enabled Computing
- Improvement in military aircraft safety by 2010

Where is the improvement going to come from

- Improvements in safety are going to have to come from all areas, e.g:
  - Organisation: Improvement in management and evaluation systems, e.g. safety cases
  - Individual error: Controlling and understanding complexity to reduce human error
  - Equipment dependability improvement, better system modelling



US accident analysis research on causes – based on accident reports

Michael Holloway, Chris Johnson, Aug 2005

The evidence based approach to software safety

Overview Evidence based approach

- The Def Stan 00-56
- Explaining the Evidence as a measure against risk
- Evidence based approach to software safety
  - Using the Legal Analogy

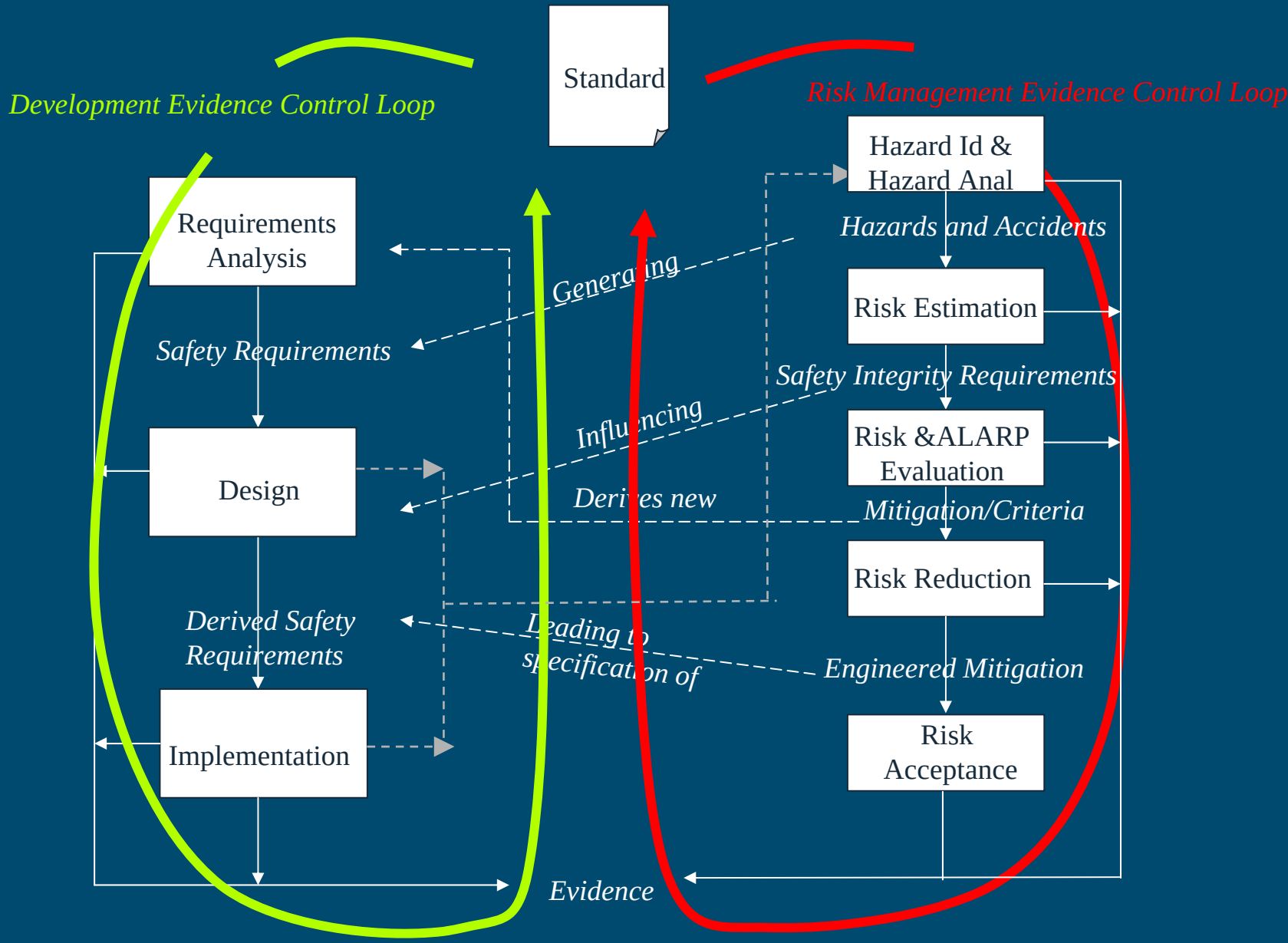
Software

- The Def Stan 00-55 approach and its problems
  - "the Standard describes the requirements for procedures and technical practices for the development of Safety Related Software"

- The Def Stan 00-56 system approach
  - "the Standard describes the requirements for procedures and technical practices for the development of Safety Related Software"

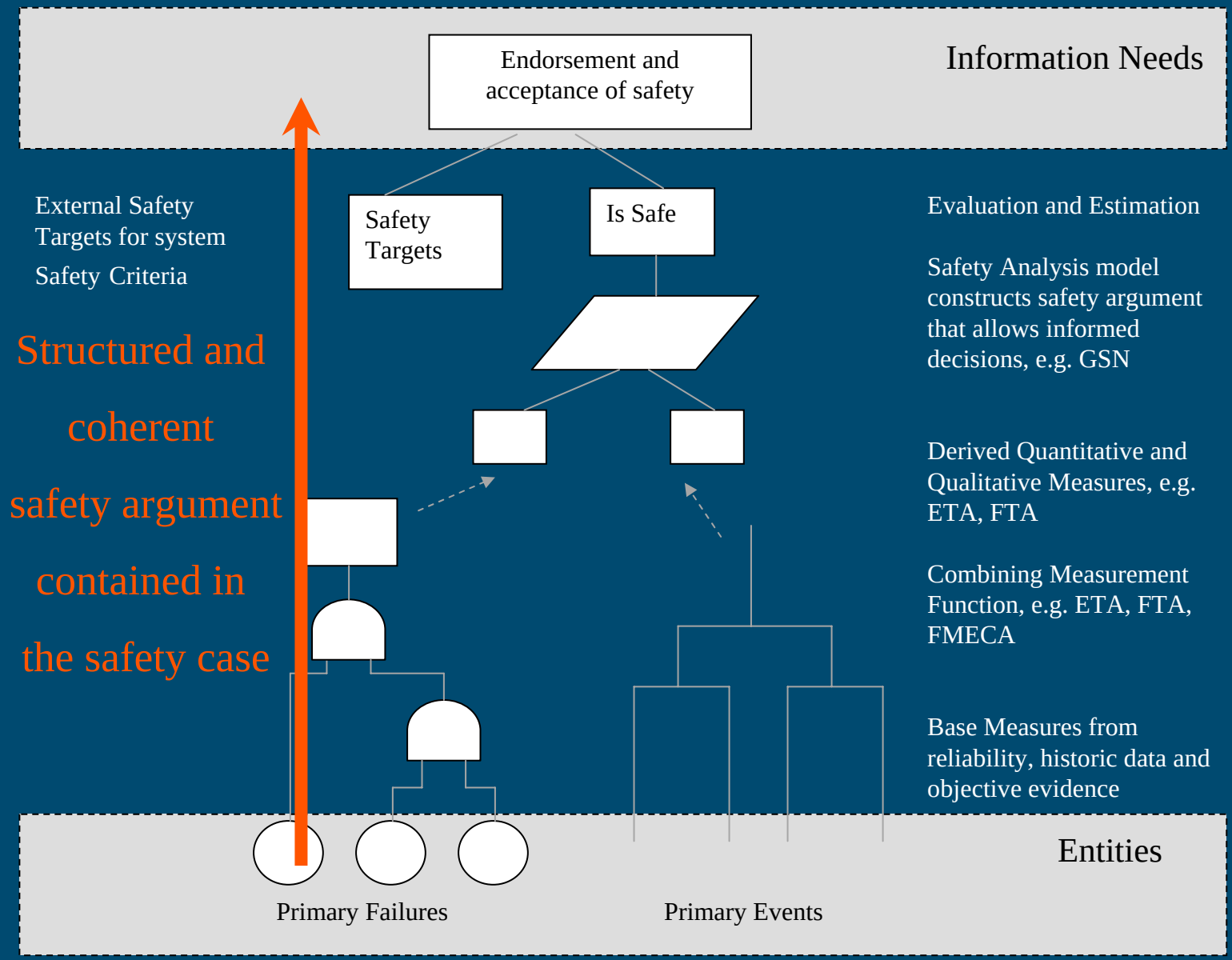
Complex Electronic Element = software + hardware

THE SOFTWARE PART IS GUIDANCE ONLY


Development Evidence Control Loop
Standard
Risk Management Evidence Control Loop
Requirements Analysis
Safety Requirements
Design
Derived Safety Requirements
Implementation
Evidence
Generating
Influencing
Derives new
Leading to specification of
Hazard Id & Hazard Anal
Hazards and Accidents
Risk Estimation
Safety Integrity Requirements
Risk &ALARP Evaluation
Mitigation/Criteria
Risk Reduction
Engineered Mitigation
Risk Acceptance

MOD Evidence based approach

- The strategy MOD uses for presentation of a system's safety is the safety case
- The safety case approach for system certification is evidence based
- For any safety case there are potentially many valid strategies to arguing/demonstrating that a system is safe, either graphically or with structured natural language statements, however, all of these strategies depend upon sound evidence
- Software evidence is usually qualitatively based


Endorsement and acceptance of safety
Information Needs
Safety Targets
Is Safe
External Safety Targets for system Safety Criteria
Structured and coherent safety argument contained in the safety case
Evaluation and Estimation
Safety Analysis model constructs safety argument that allows informed decisions, e.g. GSN
Derived Quantitative and Qualitative Measures, e.g. ETA, FTA
Combining Measurement Function, e.g. ETA, FTA, FMECA
Base Measures from reliability, historic data and objective evidence
Entities
Primary Failures
Primary Events

Software Evidence (Classes)

- Software evidence should support the System approach to demonstrating safety
  - it is the system that is being certified
- The Software evidence can be grouped four areas:

- Qualitative process evidence, an indirect qualitative measure, e.g. software developed by a specific technique;
- Quantitative predictive analysis Historic evidence, Failure data, maintenance records, reliability modelling
- Demonstration, Test evidence, A quantitative measure of the product, e.g. qualification testing
- Analytical, direct mathematical qualitative/quantitive measures of the product

Qualitative Process Evidence

- Process evidence is one of the most prevalent arguments used for software safety, it supported by:
  - Verifiable historic records of success, th software process measurement program
  - Quality audit records pr dance with an accredited quality system, e
  - The writte process procedures supported by an org ssment scheme, e.g. CMMI
  - Proc oducts, internal and delivered artefacts
- But... The general assertion that the “process addresses this area” looks very poor when considering a specific failure mode or safety property. It is a weak argument.

Circumstantial Evidence

Quantitative Prediction Evidence

- Historic evidence implies reliability-based data using models
  - Time related Quantitative models
  - Non time related Quantitative models
  - Qualitative models
- There is considerable debate on th… of time related software reliability models due to proble… software's environment changes
  - UK nuclear power i… dicate that reliability can never be better than 10^{-4} … for failure on demand safety critical software
- The use of software non-time related quantitative and qualitative reliability modelling is gaining wide academic support, e.g Bayesian Belief Networks

Previous Character

Demonstration Evidence

- For software there are two basic forms of Demonstration evidence that of *dynamic* and *static* testing.
  - Dynamic allows:
    - Model based testing (testing of s ns)
    - The assessment of quality a sing test coverage statistics
    - Demonstrate specifi s, modes and mitigation
    - but .. “testing ca onstrate the presence of errors, it cannot demo r absence”, Dijkstra
  - Static allows
    - subjective ma ual static analysis, eg Fagan inspections
    - Semi-automatic, or tool supported, for software code, identifies classes of known program errors.

Witness Testimony

Analytical (Proof)

- Proof evidence is generated by the application of formal methods or in some cases advanced static test tools
  - As with software testing there are various forms of mathematical proof evidence
  - All claims made for a mathematical proof need to be viewed in the context that they are produced
- Many in Industry are sceptical of FM due to project scale, additional costs or systems special nature
  - This can never be dismissed but successful formal method projects indicates that this view is progressively less credible
- It is the strongest form of evidence and preferred by MOD

Forensic Evidence

Balancing the Evidence

- Process evidence. Does not show safety, must be viewed as circumstantial,
- Reliability evidence. It is based on historic data, can be viewed as past convictions
- Demonstration Test Evidence. Witness evidence carries considerable weight. But the witness limitations and viewpoint have to be considered. You can only test for the presence of errors not the absence.
- Analytical (Proof). Regarded by MOD as the strongest form of evidence .. It is after all called proof! It is driven by science just like forensic evidence. However it needs to be put in to context before use.

Diversity - it works but has limits

- There are two common diverse evidence based arguments.
  - Engineers can design in diversity by multiple channels, redundancy
    - Indicates that the balance of evidence favours design diversity in software, reliability is improved but is “less reliable than a simple assumption of statistical independence would suggest”, Littlewood et al 2001
  - Analysis can generate additional verification evidence using diverse test and analysis techniques
    - truly diverse fault finding techniques are used, it is possible for the diversity to be more effective than it would be under an assumption of statistical independence, Littlewood, et al 2000

Summary of software Def Stan 00-56

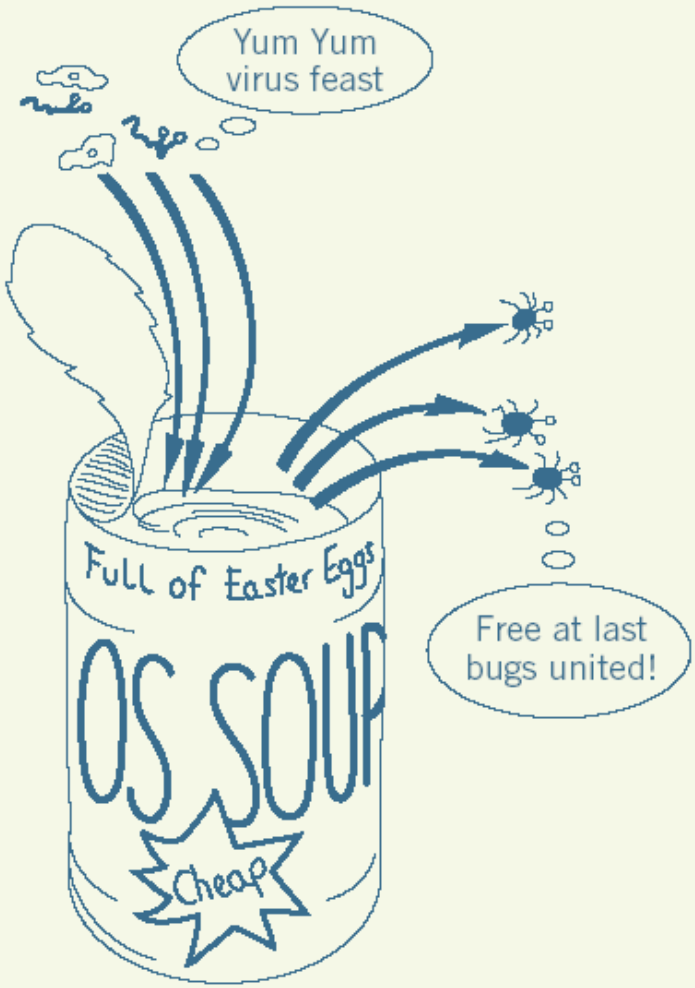
- The Safety argument is based on evidence. The type of evidence provided for primary arguments should be based on the precedence:
  - Analytical evidence (mathematically based methods, SCA etc);
  - Quantitative analysis (availability, reliability - in context);
  - Demonstration evidence and review evidence (testing dynamic and static);
  - Qualitative evidence of good design, and process evidence.

Legal equivalent?

- Forensic
- Character
- Witness
- Circumstantial

Evidence based Conclusions

- The current MOD strategy for safety management is based on project safety cases
  - The safety arguments within the safety case are supported by sound safety evidence
  - For software many forms of objective and diverse evidence exist
  - Some evidence, e.g. process, is not conclusive but is more analogous to circumstantial evidence – albeit useful.
  - Even proof, which is considered the best form of evidence, needs additional support and context before it should be trusted.
  - The best software safety arguments in safety cases are diverse and supportive in nature. Usually qualitative.
- Nothing gets certified or released without the safety case containing a convincing evidence base argument



Some Cots Operating Systems
are better than others

Questions?