Safety Assurance Principled? Prescriptive?

Tim Kelly

The University of York

tim.kelly@york.ac.uk

Context

- Lack of agreement in the details of requirements of software assurance standards for safety-critical systems has long been recognised
- Tension exists between those advocating demonstrating compliance to standards as the principal assurance approach and those that promote the production of assurance cases
- Often incorrectly presented as totally opposing alternative approaches
  - Distracts from the threats to confidence that can exist in both approaches

4+1 Principles

- P1 - Software safety requirements shall be defined to address the software contribution to system hazards.
- P2 - The intent of the software safety requirements shall be maintained throughout requirements decomposition.
- P3 - Software safety requirements shall be satisfied.
- P4 - Hazardous behaviour of the software shall be identified and mitigated (Note - includes prevention and avoidance)
- P4+1 - The confidence established in addressing the software safety principles shall be commensurate to the contribution of the software to system risk (Note - doesn't require probabilistic treatment of software failure)

Observations

- P1-3 can be observed to be at the heart of many standards
- P4 is often less well addressed
- Many standards attempt to address P4+1 through SILS or (e.g.) varying evaluation to be commensurate to severity of consequences
  - Questions still remains as to whether the different levels of evaluation genuinely relate to reduction in risk in the 'end product'

Generic vs. Specific Application of Principles

- intent of principles is not that they are addressed generically (e.g. by appeal to generic processes or adherence to standards)
- should be evidenced specifically
- requirements and processes of a standard may be capable of demonstrating principles, but may still fall short in practice
  - consider Requirements Review
- application of standards cannot be considered in a tokenistic sense, as a talisman of confidence
- An area where confidence can be lost, also where assurance cases can help

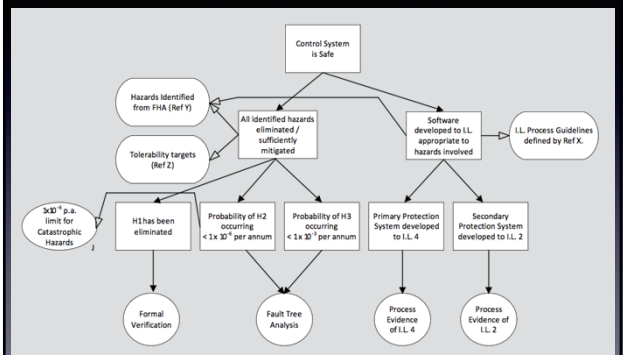
Generic vs. Specific Application of Principles

- Significant issue re: P4+1
  - Standards established a general set of requirements for varying requirements, processes and techniques according to criticality (e.g. severity)
- Generality is potentially a problem
  - Is it what's required in a specific case - e.g. applicability of MCDC metrics?
  - Opportunity cost of doing something that doesn't add to confidence
- Some mechanisms to address, e.g.
  - PSAC, SAS in DO-178C, Justification of selection from amongst 'loose' SIL recommendations in IEC 61508

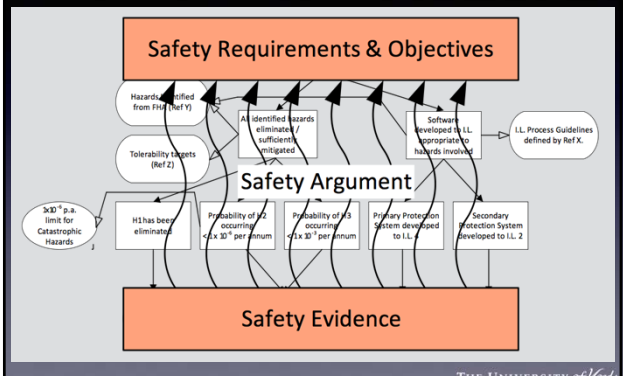
How Principles relate to Assurance Cases

- Example definition of a safety case case:
  - 'a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that the system is safe when forming part a system for a given application in a given environment.'
    (UK Defence Standard 00-56)
  - Requires interpretation when applied to software (e.g. assurance re: "hazardous software failure modes")

Example: Goal Structuring Notation



Example: Goal Structuring Notation



Complementarity (or why Assurance Cases are a good idea)

- the requirements of standards can often require interpretation in their enactment
- recording the justification (for a specific project) of the specific (judgement-oriented) aspects of the enactment of a standard

Targeting Assurance Case Effort

- P1 - assurance cases are well suited to the (inevitably subjective) justification of the adequacy of the identified software safety requirements
- P2 - well suited to the hard problem of the justification of maintenance of intent in traceability structures
- P3 - well suited to the justification of the adequacy of evidence (e.g. the appropriateness and trustworthiness of specific forms of evidence for requirements satisfaction)
- P4 - usefully targeted at the justification of the management of unintentionally hazardous side effects of otherwise intentional design commitments
- P4+1 - directly relates to the notion of a confidence / meta argument

Summary

- There are 4+1 fundamental principles of software safety assurance.
- In many current standards:
  - P1-3 served well, P4 & 4+1 not so well
- Standards can suffer from problems relating to specific enactment and judgement
  - Standards can't remove (subjective) judgement
  - Assurance cases are good at explicitly representing and recording judgements
- Crass to say it's either-or