# Evidence Based Certification: The Safety Case Approach

Tim Kelly

E-mail: **tim.kelly@cs.york.ac.uk**

High Integrity Systems Engineering Group
Department of Computer Science
THE UNIVERSITY of York





## Overview

- Brief History of Safety Cases
  - Goal-based vs. prescriptive regulatory approaches
  - Problems with software process assurance
- Safety Case Purpose and Structure
  - Arguments and Evidence
- Sufficiency of Assurance Cases
  - Argument types
  - Trustworthiness of Evidence
- Safety Case Development Process
  - Incremental Safety Case Development
- Assurance Case Benefits and Risks

# A Brief History of (UK) Safety Cases

- Number of serious accidents, e.g.
  - Windscale Nuclear Accident (late 1950s)
  - Piper Alpha Off-shore Oil and Gas Platform Disaster (1990s)
  - Clapham Rail Disaster (1990s)
- Prompted reconsideration of how safety is managed in the safety-critical sector
  - Industries were **not** ignorant of safety
  - Safety standards **existed** – but often based on **prescriptive** codes
  - **What Was Missing:** Systematic and thorough consideration of safety, and communication of this to a regulator

COMPLETENESS

- Prescription
  - Designers / operators claim safety through satisfaction of the ***regulator's*** requirements
- 'Goal-based' standards
  - Up to the ***designers*** to demonstrate that they have an adequate argument of safety in support of high level objectives (e.g. ALARP)





# Problems with Software Process Assurance

- Prescribed processes do not necessarily lead to achievement of a specific level of integrity
- Poor correlation between prescribed techniques and failure rate
  - Implicit belief in 'risk reduction'
- Note – more generally we have the problem of software failure rate prediction (e.g. correlation between code metrics and in-service experience)
- Prescription in safety standards hinders the adoption of new process approaches that could improve flexibility and predictability of system development
  - e.g. Model Driven Development
- These problems have led toward increasing adoption of **Software** Safety Cases

# ∴ Motivation for Safety Cases

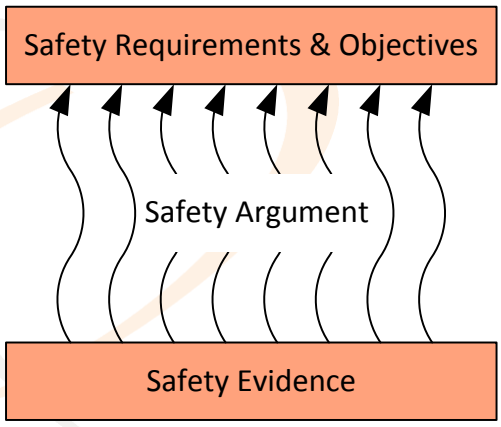
- **Completeness** – hard to judge …
  - … when evidence is *distributed* and *diverse*
  - … when arguments are *implicit*
- **Rationale** behind prescriptive requirements missing
- **Knowledge Imbalance** – developers know more about their products than the regulators
- Some existing forms of assurance are increasingly considered too **indirect**





# Evidence-Based Certification: Safety Cases

- **Well structured and reasoned safety arguments**
  - Demonstrate satisfaction of safety objectives derived from hazard analysis
  - Justify acceptability of safety based on product-specific and targeted evidence
  - (Potentially) justify the determination of objectives and the selection of evidence



# The Purpose of a Safety Case

- **A safety case presents the argument that a system will be acceptably safe in a given operating context**
- 'System' could be ...
  - Physical (e.g. aero-engines, reactor protection systems)
  - Procedural (e.g. railway operations, off-shore)
  - Software
- Safety cases increasingly adopted in the defence, automotive, rail, oil and gas, and process industries





# Some Safety Case Definitions

- *"A safety case is a comprehensive and structured set of safety documentation which is aimed to ensure that the safety of a specific vessel or equipment can be demonstrated by reference to:*
  - *safety arrangements and organisation*
  - *safety analyses*
  - *compliance with the standards and best practice*
  - *acceptance tests*
  - *audits*
  - *inspections*
  - *feedback*
  - *provision made for safe use including emergency arrangements"*

**(JSP 430 Issue 1)**

- *"A Safety Case is a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment."*

**(UK Defence Standard 00-56 Issue 4)**

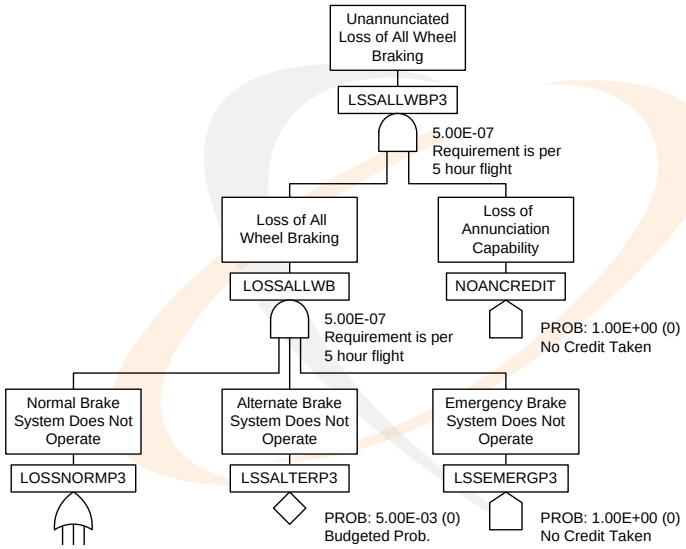
# Argument & Evidence

- ***Supporting Evidence***
  Results of observing, analysing, testing, simulating and estimating the properties of a system that provide the *fundamental* information from which safety can be inferred
- ***High Level Argument***
  *E*xplanation of how the available evidence can be reasonably interpreted as indicating acceptable safety – usually by demonstrating compliance with requirements, sufficient mitigation / avoidance of hazards etc
- Argument without Evidence is **unfounded**
- Evidence without Argument is **unexplained**

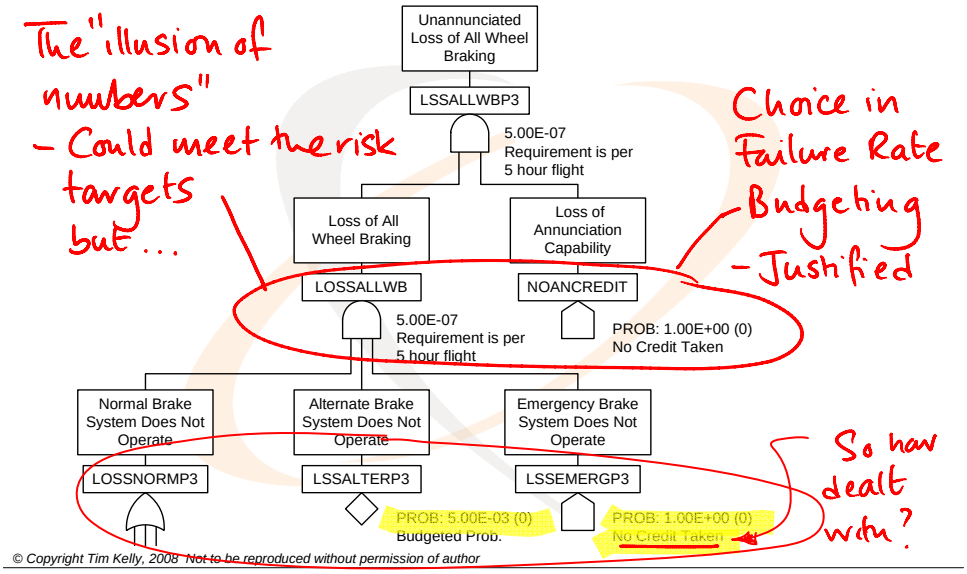




# Fault Tree Analysis Example 1



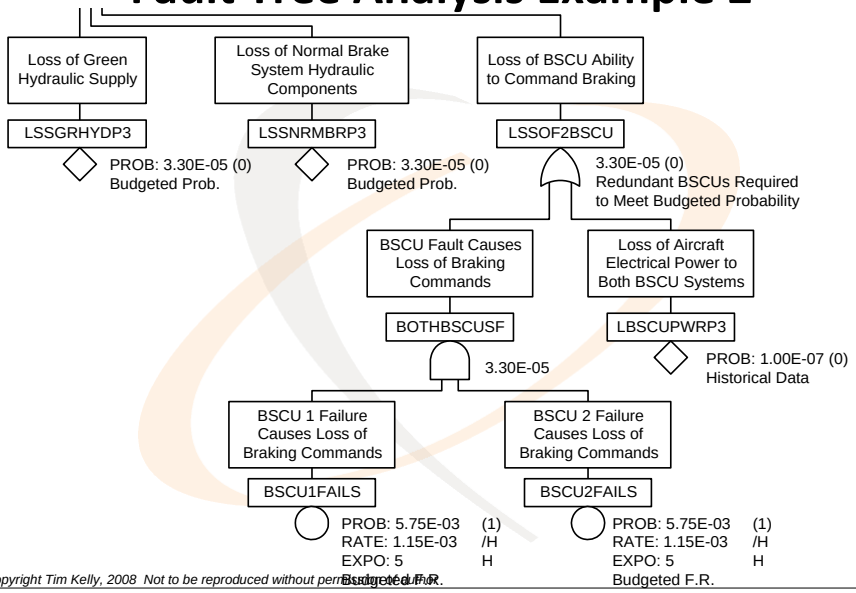
# Fault Tree Analysis Example 1



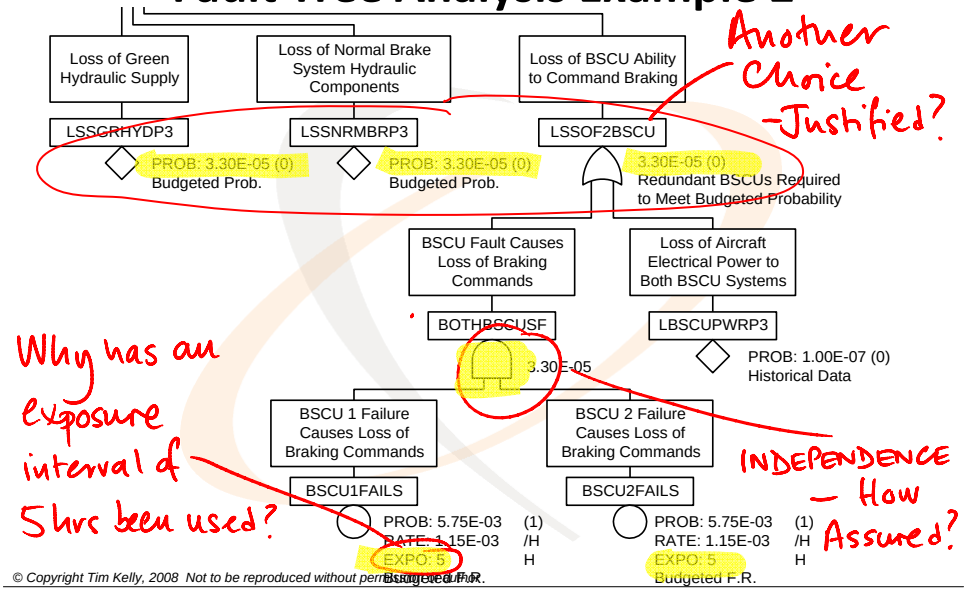





# Fault Tree Analysis Example 2



# Fault Tree Analysis Example 2








# The Goal Structuring Notation (GSN)

## Purpose of a Goal Structure

To show how **goals** are broken down into sub-goals,

and eventually supported by evidence (**solutions**)

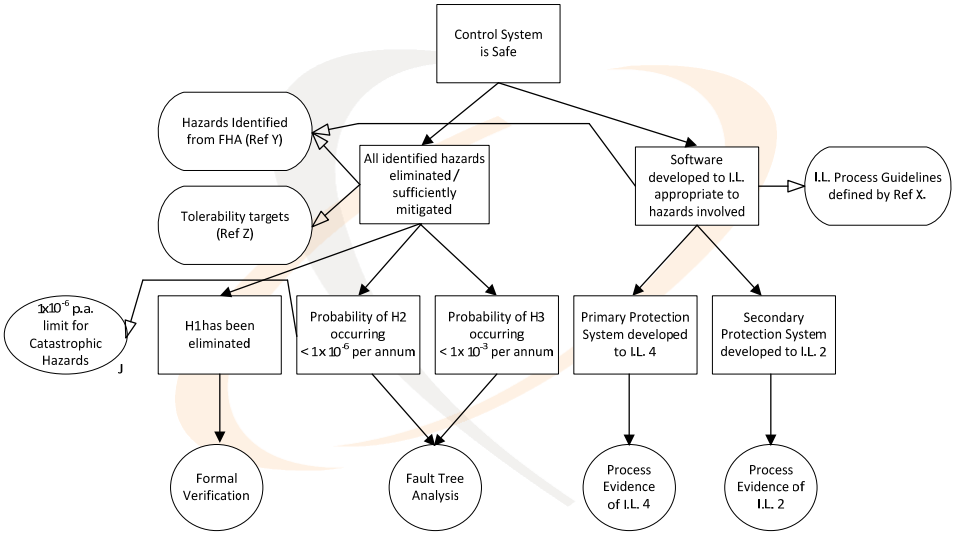
whilst making clear the **strategies** adopted,

the rationale for the approach (**assumptions**, **justifications**)

A/J

and the **context** in which goals are stated

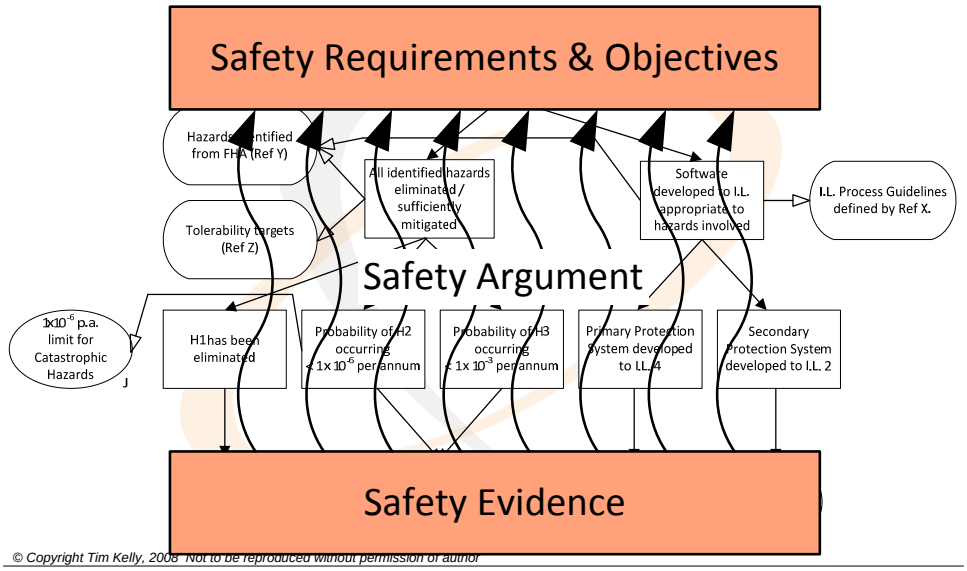
# A Simple Goal Structure







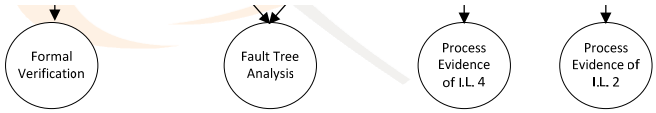

# A Simple Goal Structure



# A Simple Goal Structure

Take away the argument & what are you left with? The "BAG of Evidence"

← Note – This is not a safety case!





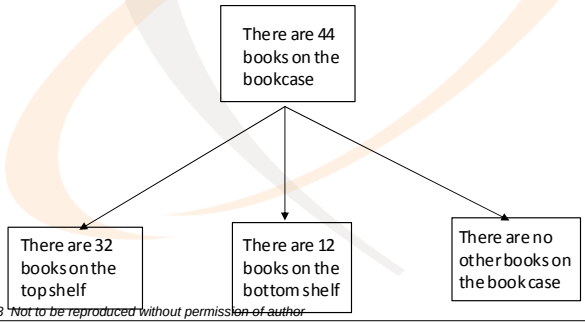



# (Further) Motivation for Safety Cases

- Completeness – hard to judge …
  - … when evidence is distributed and diverse
  - … when arguments are implicit
- Rationale behind prescriptive requirements missing
- Knowledge Imbalance – developers know more about their products than the regulators
- Some existing forms of assurance are too indirect
- The role of evidence can otherwise be unclear
- The assumptions and implicit judgements in evidence need to be presented explicitly and argued

# *Sufficiency* of Assurance Arguments

- Assurance Arguments can be split into two types
  - Deductive arguments
  - Inductive arguments
- Deductive arguments
  - If premises are true, then the conclusion must also be true.

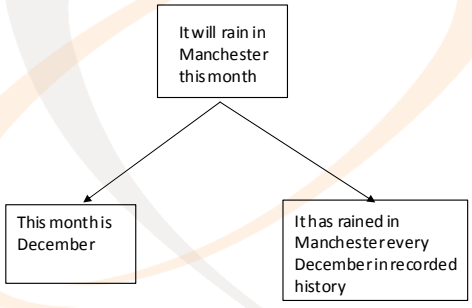






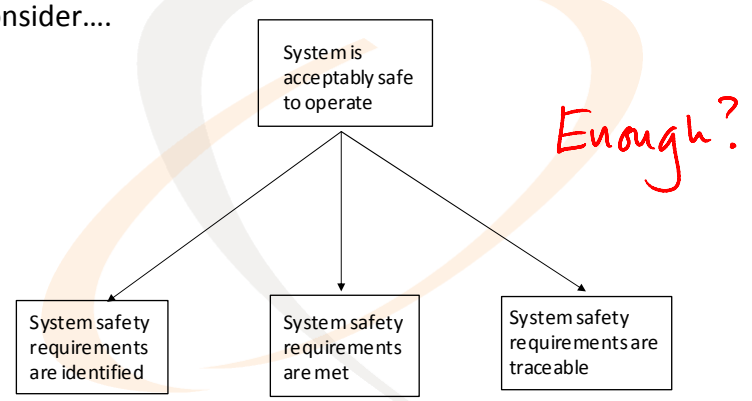

# *Sufficiency* of Assurance Arguments

- Inductive arguments
  - The conclusion follows from the premises not with necessity but only with probability.



# Sufficiency of Assurance Arguments

- It is more common to see safety arguments which are inductive in nature
  - Consider….



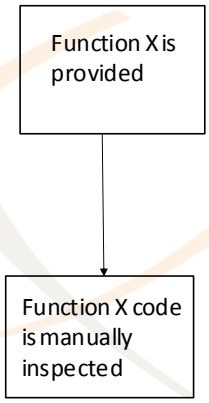

  - The premises give us ***confidence*** in the truth of the conclusion





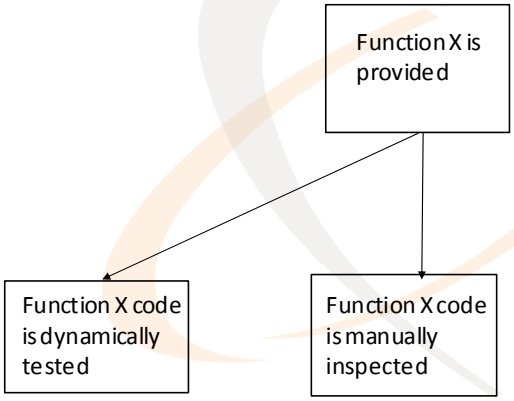
# Confidence in a Claim

- We can increase the confidence that we have in the truth of a goal by providing more support…



# Confidence in a Claim

- We can increase the confidence that we have in the truth of a goal by providing more support…

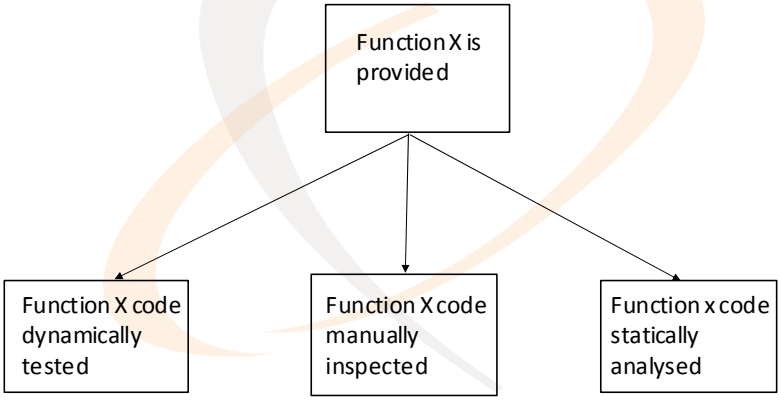







# Confidence in a Claim

- We can increase the confidence that we have in the truth of a goal by providing more support…



# Safety Evidence Assurance

- **Relevance**
  - How relevant is a piece of evidence or argument to the conclusion being sought
  - e.g. safety of reusable component with failure-free operational record
- **Coverage**
  - To what extent does the argument / evidence presented 'cover' the conclusion
  - e.g. limited testing
- **Trustworthiness**
  - Thoroughness of evidence generation
  - e.g. staff competency & tool qualification





# Trustworthiness of Evidence

Number of possible factors to consider:

Thoroughness – related terms: depth / rigour of analysis

- "Buggy-ness" – how many "faults" are there in the evidence presented
  - High faults (related to safety case "intent") = loss of confidence
- Level of Review
- In case of hand-generated evidence:
  - Experience of Personnel
  - Competency of Personnel
- In case of tool-derived evidence
  - Tool Qualification and Assurance
  - NB – Importance distinction between tools where output forms part of product vs. those with ancillary role

# Evidence in Defence Standard 00-56

- Deductive, Inductive, Judgemental distinction drawn (in Part 2):
  - "9.5.5.1 In general, arguments based on explicit, objective evidence are more compelling than those that appeal to judgement or custom and practice. It is therefore recommended that any argument should be developed in accordance with the following precedence:
    - **Deductive**, where the conclusion is implicit in the evidence used to support the argument.
    - **Inductive,** where the argument is firmly based on the evidence presented, but extrapolates beyond the available evidence.
    - **Judgmental,** where expert testimony, or appeal to custom and practice is necessary to support the conclusion."





# Evidence in Defence Standard 00-56 2

- Quantity and Quality of Evidence:

*"11.3 Provision of Evidence*
*11.3.1 Within the Safety Case, the Contractor shall provide compelling evidence that safety requirements have been met. Where possible, objective, analytical evidence shall be provided.* ***The quantity and quality of the evidence shall be commensurate with the potential risk posed by the system and the complexity of the system****. For safety requirements that lead to the realisation of mitigation strategies, the* ***quantity and quality of the evidence shall be commensurate with the level of Risk Reduction resulting from that safety requirement****."*

# Evidence in Defence Standard 00-56 3

- DS 00-56 Issue 3 recognises role for **counter**-evidence:

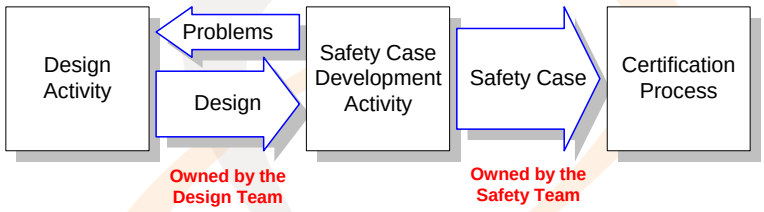
> "9.5.6 Throughout the life of the system, the evidence and arguments in the Safety Case should be **challenged in an attempt to refute them**. Evidence that is discovered with the potential to undermine a previously accepted argument is referred to as counter-evidence. The process of searching for potential counter-evidence as well as the processes of recording, analysing and acting upon counter-evidence are an important part of a robust Safety Management System and should be documented in the Safety Case."





# Safety Case Development Process 1

- Safety cases sometimes mistakenly produced after design completion




**Resulting Problems:**

- Potentially large amounts of re-design
  - late in the design lifecycle
  - occasionally product must be discarded and redeveloped
- Less robust safety arguments – "Apologetic"/"Excusing" rather than **positive** appeal to design features
- Lost safety rationale

# Safety Case Development Process 2

- To avoid **wasted effort**, projects should be considering from the earliest possible opportunity:

  ***"How will we argue that this system is safe?"***

  ***"What is the safety case going to look like?"***

- Development of the safety case early should be initiated early on in a project and carried through
  - from requirements definition
  - through to entry into service and beyond





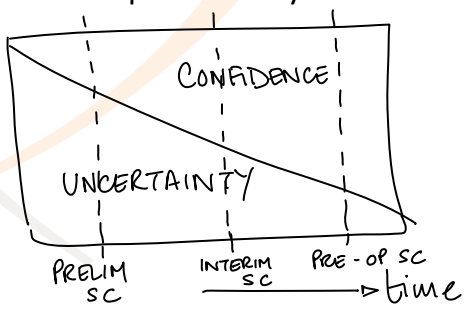
# What Some UK Standards Say

Examples:

- *"The Safety Case should be initiated at the earliest possible stage in the Safety Programme so that **hazards are identified and dealt with while the opportunities for their exclusion exist**"* (DS 00-56 Issue 2)
- *" … the Safety Case will initially identify the **means by which safety will be achieved and demonstrated**; at later stages detailed arguments and supporting evidence will be **developed and refined.**"* (DS 00-56 Issue 4 Part 2)
- *"The Safety Case is to be prepared in **outline** at presentation of the Staff Requirement and is to be updated at each major procurement milestone up to and including hand-over from the procurement to the maintenance authority … Ideally there should be a **seamless development of the Safety Case from one phase to the next**" (JSP 430 Issue 1)*

# Phased Safety Case Production

- Increasingly common requirement to produce safety cases at a number of intervals during system development
  - **Preliminary** Safety Case
  - **Interim** Safety Case
  - **Pre-Operational** Safety Case
- Sometimes, additional Safety Cases required at key project milestones
- Stages reflect increasing confidence
  - means of managing *project* uncertainty & risk








# Preliminary Safety Case

- Define scope of consideration for safety case
- Identify key safety issues associated with system
  - system hazards
  - safety requirements / applicable standards
- Define approach being taken to safety argument
  - setting out key safety objectives
  - pointing forward to forms of evidence to be used
  - what you will do when further design info. available
- Define (safety-relevant) development procedures
  - e.g. for specific Software Integrity Levels /Level of Concern
- **Discussing PSC widely with stakeholders (including regulator) provides an early opportunity to find out if there any *fundamental* flaws in the approach being adopted**
  - **Looking for the non-negative answer**

# Interim Safety Case

- Increase confidence that safety case as outlined (Preliminary Safety Case) can be supported
- How design as proposed ensures *safety properties* & avoids *hazards* as outlined in Preliminary Safety Case
  - protection measures / safeguards / interlocks
  - design features – e.g. memory protection mechanisms
- Incorporate evidence from 'Preliminary Validation Activities'
  - e.g. Hazard and Operability Study over a detailed architectural proposal.

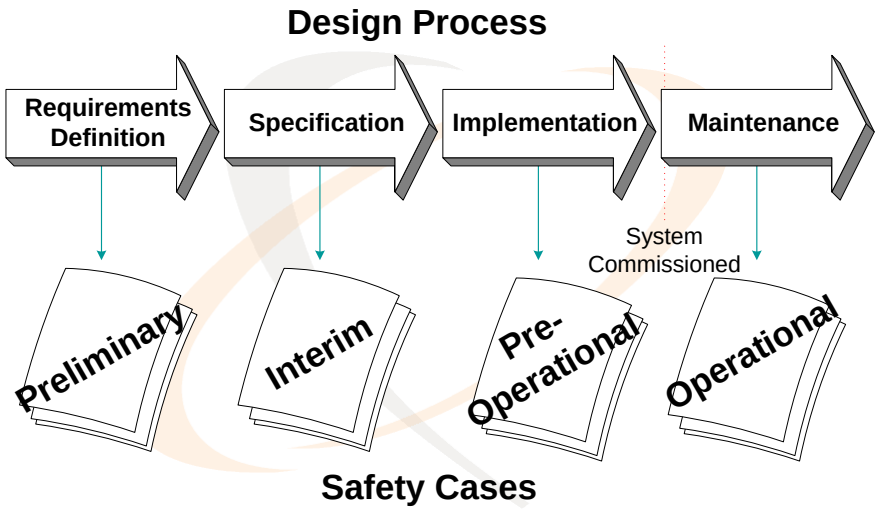




# Pre-Operation Safety Case

- Provide sufficient assurance that system ***as constructed*** is sufficiently safe to be allowed to operate in its intended operational context for the first time
- Present final (substantially more detailed) arguments based on the observation, measurement, testing and analysis of the implemented system where possible and *reasonable* assumption where it is not
- Description of 'work-arounds' for any identified 'problems'

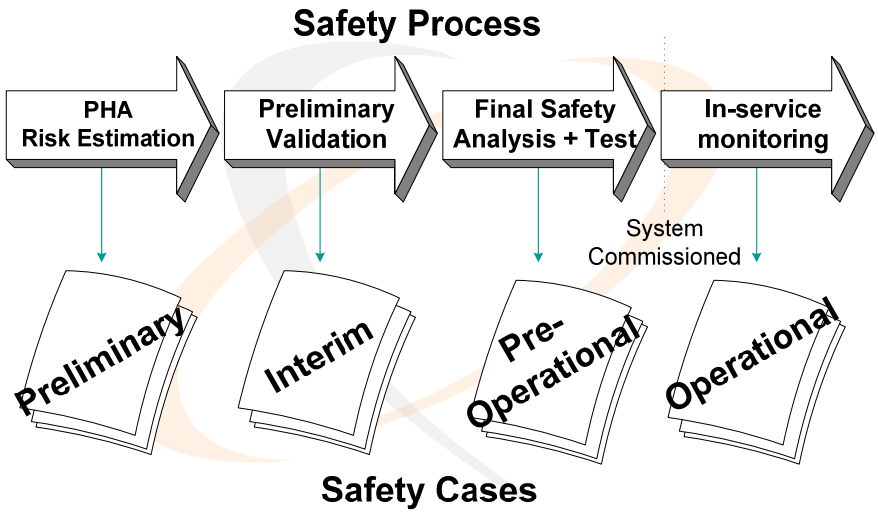
# Safety Case and the Design Process



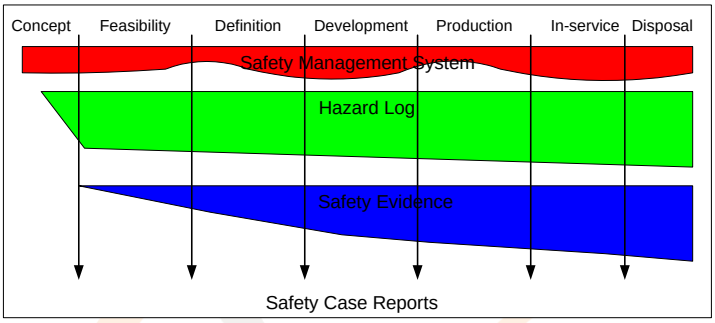





# Safety Case and the Safety Process



# Phased Safety Cases as 'Snapshots'




- Safety Case Reports present "snapshots" as safety information builds up
- If comparing options (e.g. at tender), there may be multiple safety cases





# When to do Phased SC Development?

- Phased SC Development should be employed wherever certification risk and uncertainty is high, e.g.
  - **Novel System**
    - New technologies used
    - New construction methods
  - **Novel Safety Argument**
    - Deviation from standards
    - Novel forms of evidence used
- In either case, phased approach should be used to define, and obtain (tentative) acceptance of certification approach
- Note – uncertainties or disagreements in certification approach are often not apparent (until too late!) without early articulation of the assurance argument

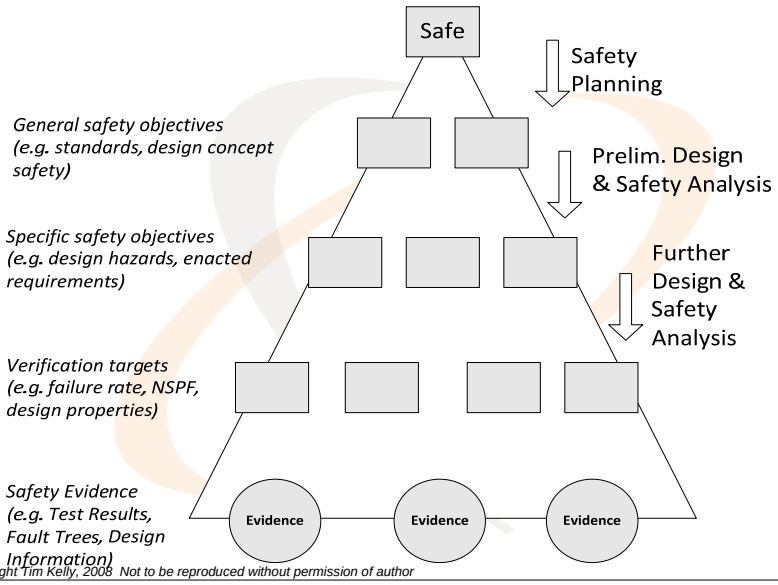
# When to Articulate the Assurance Argument?

- Q: At what stage in a project is it worth attempting to articulate the assurance argument?
- Answers:
  - Early on (high level) to get a clear picture (and gain agreement) of argument structure
    - Useful as a scoping exercise and effort allocation
  - As project is progressing, in order to monitor status towards completion of an acceptable argument
  - At end of project in order to present the final argument and evidence that exists





# Progression of an Assurance Argument



# Assurance Case Benefits

- *Mitigation for the following project risks*
  - Excessive iterations involved in reaching agreement on the sufficiency of the evidence
  - Effort spent on project (e.g. in performing analyses) that do not really provide appropriate assurance
  - Disproportionate effort allocated across safety development and assurance activity (rabbit holes!)
    - also needs understanding of ALARP
  - Duplication of effort (inefficient) when apportioning responsibility
  - Assurance objectives 'falling down the cracks' when apportioning responsibility





# Assurance Case Problems?

- Prescription had many flaws but "people knew what they were supposed to do"
  - Helps project predictability (cost and timescales)
- Subjective assurance arguments, including explicit arguments of "good enough" could be the subject of debate with multiple stakeholders
  - Aim is mutual acceptance of a subjective position
  - Counter-argument: the assumptions are always there!
- Assurance arguments, by putting all of your arguments clearly and transparently in one place, will be open to (legal) attack
  - Counter-argument: to not have 'pulled it all together' could be seen as negligent; Assurance cases increasingly recognised as best-practice

# Summary

- Safety cases introduced because although safety was being considered, evidence generated, codes followed etc. it was often hard to see an overall (systematic, defensible) assurance argument
  - Exploiting reality that developers have more knowledge about what makes their product safe than the regulators
- Safety cases require clearly articulated **argument**, supported by references to **evidence**
- Arguments must be judged for **sufficiency**
- Incremental assurance case development can be effective feedback for design, focus evidence production effort, and for **project risk-reduction**
  - Good idea even when approach not mandated