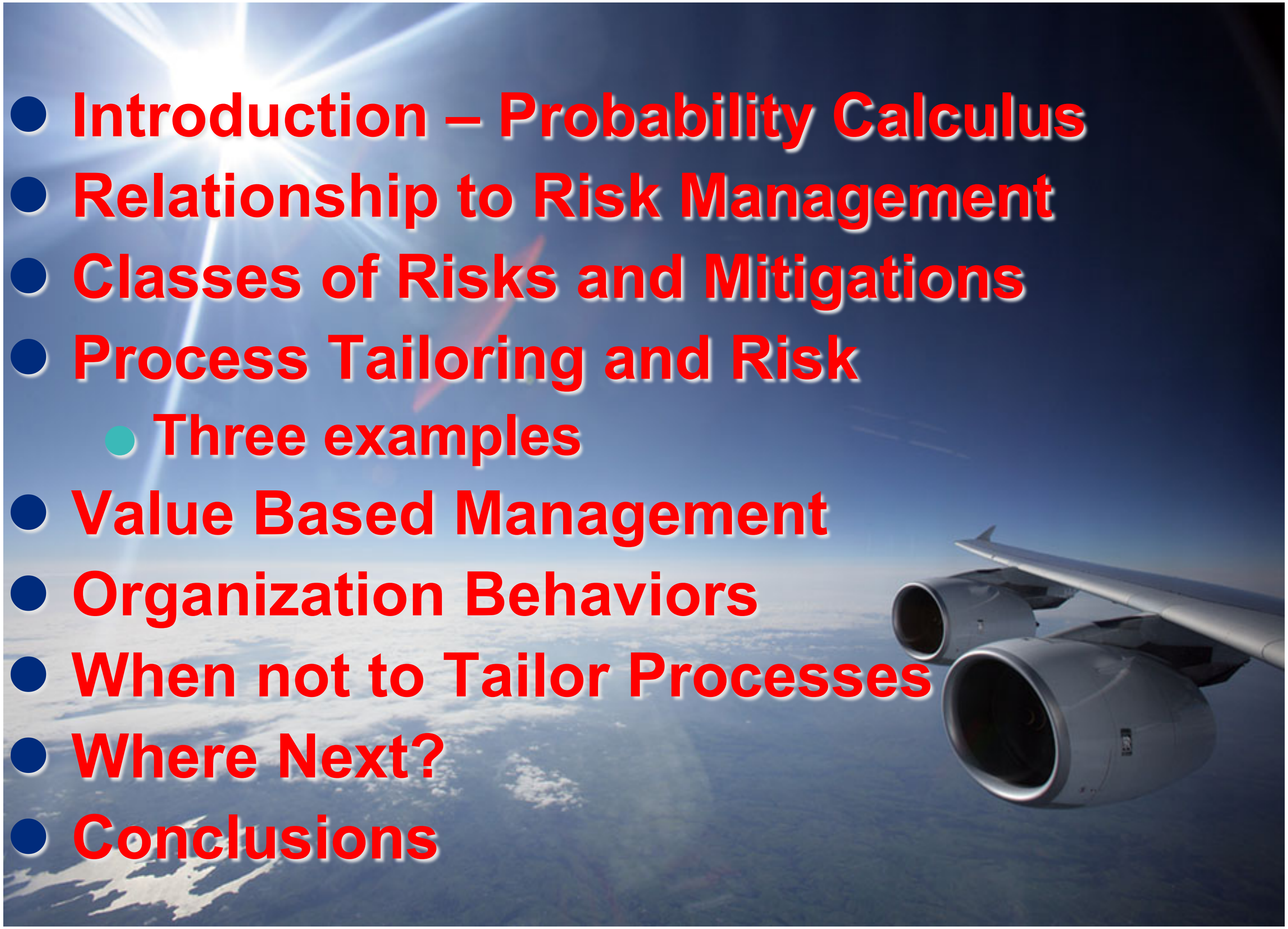
When is Enough Enough? Tailoring Processes in Systems Engineering

Andrew C Pickard,
Head of Process, Controls Engineering
Andrew J Nolan,
Chief of Software Improvements - Software Centre of Excellence
Rolls-Royce plc

Presentation Structure



- Introduction – Probability Calculus
- Relationship to Risk Management
- Classes of Risks and Mitigations
- Process Tailoring and Risk
    - Three examples
- Value Based Management
- Organization Behaviors
- When not to Tailor Processes
- Where Next?
- Conclusions

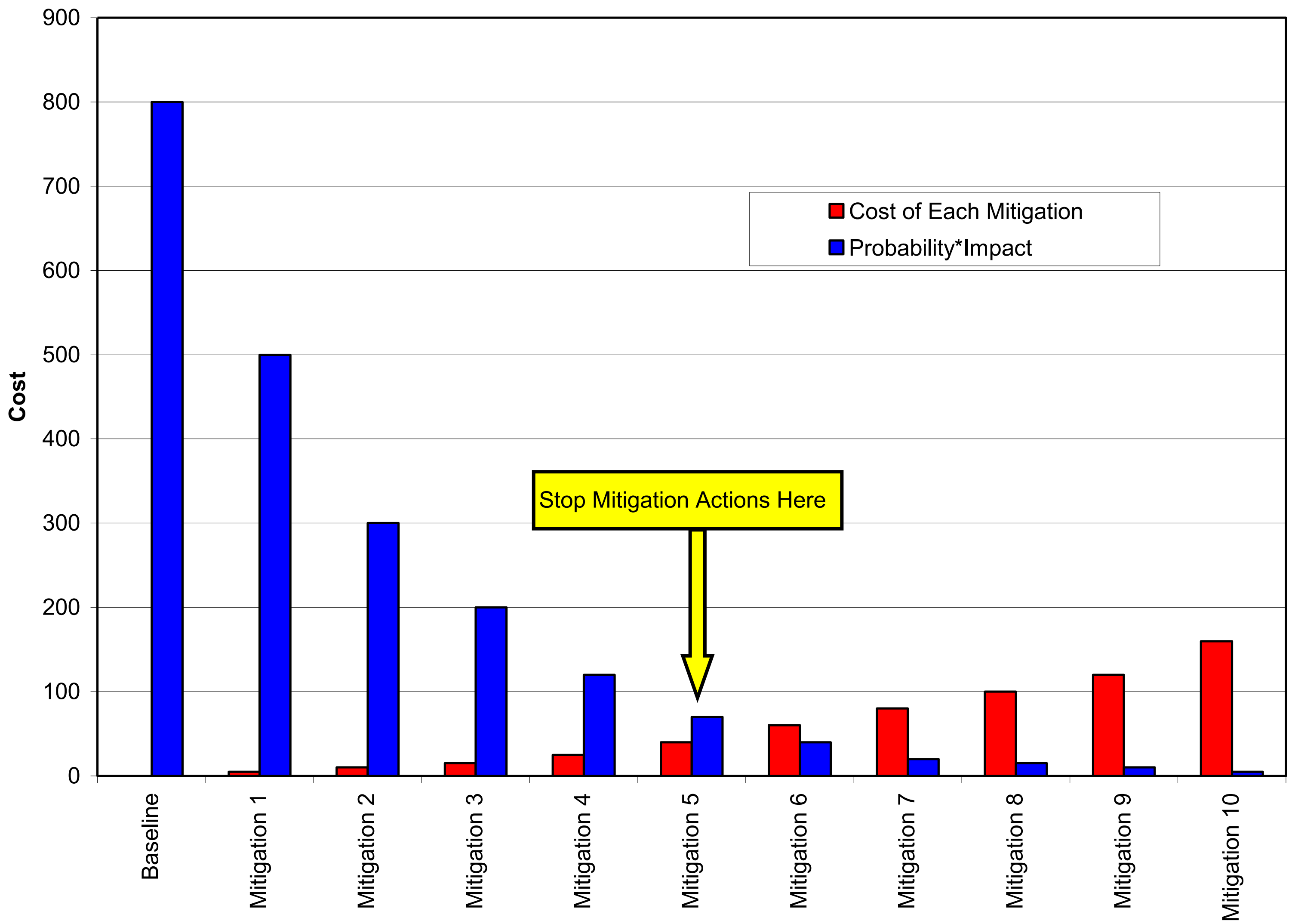
Probability Calculus

- US Federal Judge Legal Hand, 1947 defined Probability Calculus
  - Barge Accident in New York Harbor
  - United States v. Carroll Towing Co.
- Three variables:
  - The probability (P) of the event happening
  - The loss (L) if the event happens
  - The burden (B) of taking precautions to prevent the event happening
- Liability attaches when:

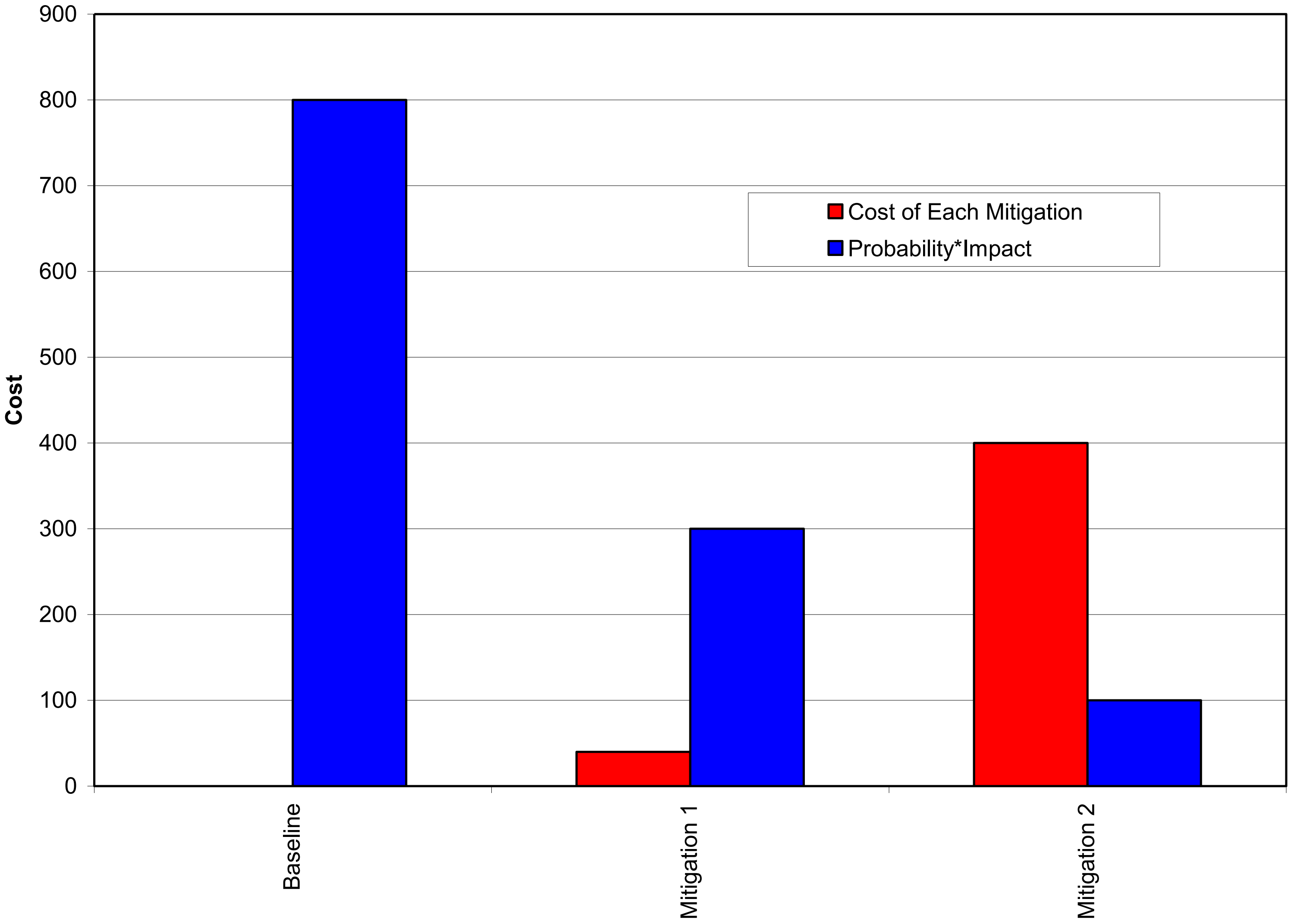
$$B < P * L$$

Under the Hand formula, it is unreasonable to not take precautions, or to exercise preventive care, whenever the cost of doing so is less than the expected loss.

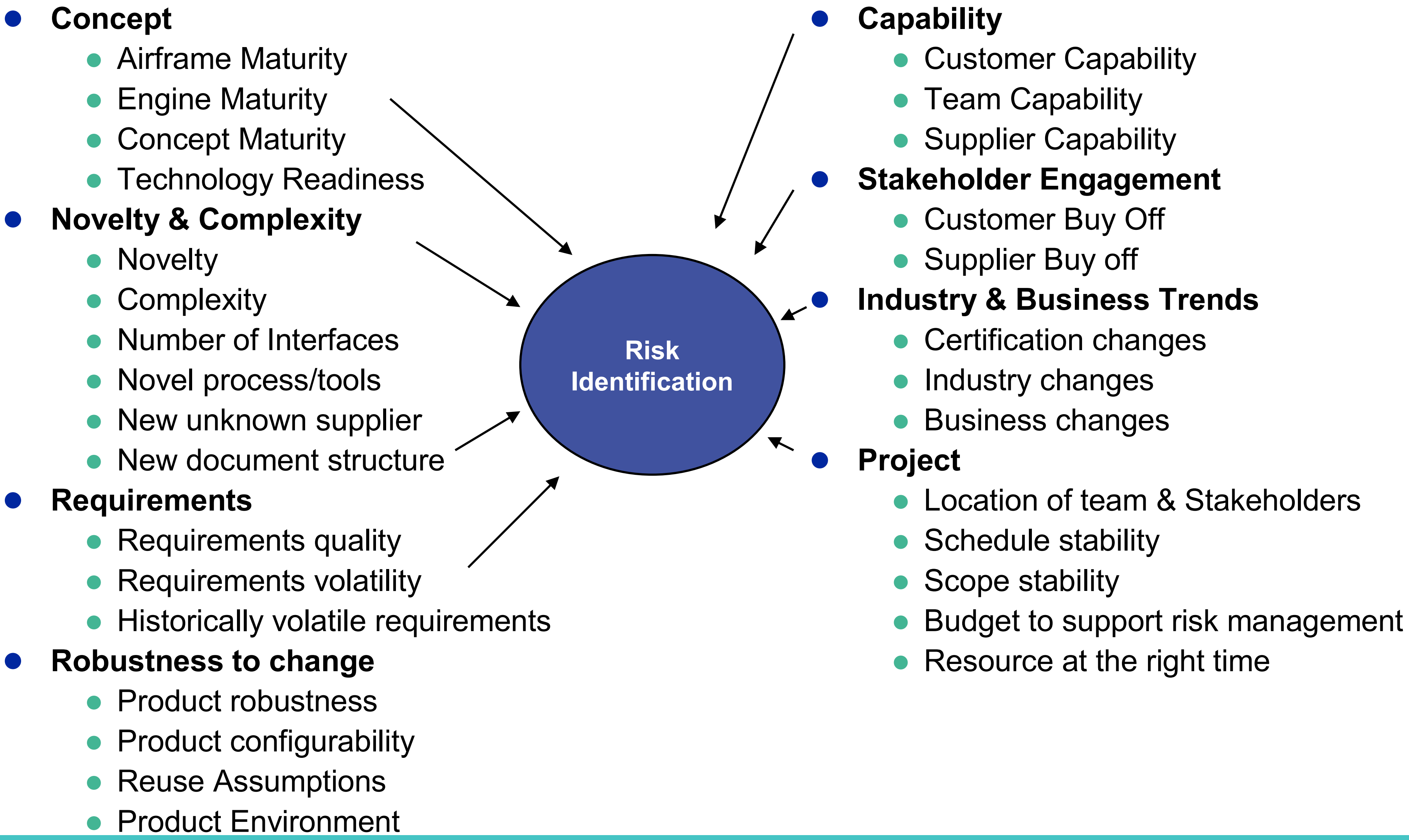
How Many Mitigation Actions?



How Many Mitigation Actions?



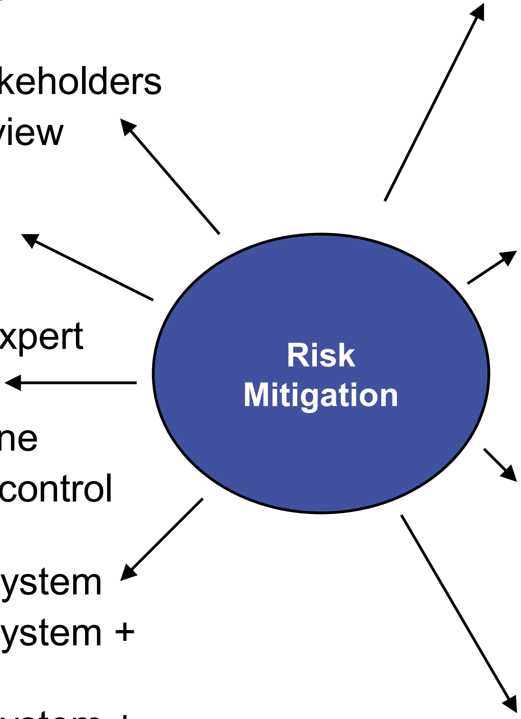
Common Risk (and Opportunity) Classes



Risk Identification

- **Concept**
  - Airframe Maturity
  - Engine Maturity
  - Concept Maturity
  - Technology Readiness
- **Novelty & Complexity**
  - Novelty
  - Complexity
  - Number of Interfaces
  - Novel process/tools
  - New unknown supplier
  - New document structure
- **Requirements**
  - Requirements quality
  - Requirements volatility
  - Historically volatile requirements
- **Robustness to change**
  - Product robustness
  - Product configurability
  - Reuse Assumptions
  - Product Environment
- **Capability**
  - Customer Capability
  - Team Capability
  - Supplier Capability
- **Stakeholder Engagement**
  - Customer Buy Off
  - Supplier Buy off
- **Industry & Business Trends**
  - Certification changes
  - Industry changes
  - Business changes
- **Project**
  - Location of team & Stakeholders
  - Schedule stability
  - Scope stability
  - Budget to support risk management
  - Resource at the right time

Common Mitigation Classes



- **Architecture Trade Study**
  - IPT - Controls
  - IPT -- Controls & Stakeholders
  - Concept proposal review
- **Review**
  - Friendly review
  - Independent review
  - Review by Domain Expert
- **Early proof of concept**
  - Prototype - stand alone
  - Prototype in existing control system
  - Modelling - Control System
  - Modelling - Control System + Engine
  - Modelling - Control System + Airframe
- **Find & Fix**
  - Airframe Test Rig or Aircraft
  - Engine Test Rig Exposure
  - Integration Test Exposure

- **DFX - Design for volatility**
  - Robust Design
  - Configurable design
  - Plug & Play architecture
  - Auto code generation
- **Design Guidance**
  - Design Guide
  - Lessons Learnt
  - Learn from historic projects
  - RIPL
- **Stakeholder engagement**
  - On site stakeholder representation
  - Visibility of stakeholder risks
  - Joint risk management sessions
  - Stakeholder reviews
- **Plan for volatility**
  - Delay the Function
  - Plan for design iteration
  - Delay freeze of design/requirements
  - If all else fails, plan in contingency

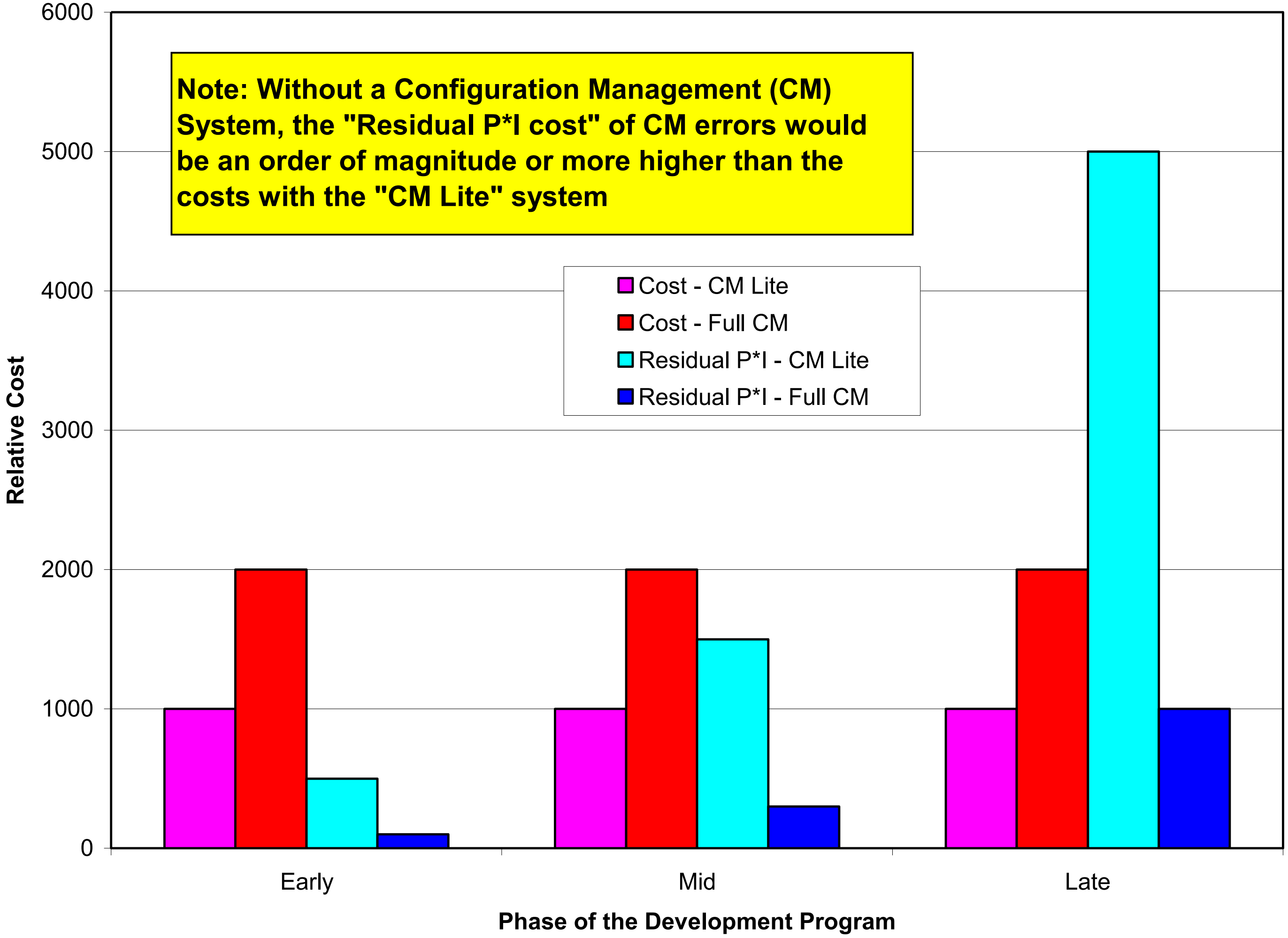
Examples of Process Tailoring

1. Selecting which activities to perform to minimize risk associated with changes introduced in a software build
2. Selecting between a "full" and "simplified" process for Configuration Management depending on the phase of a system development program
3. Selecting which Verification and Validation activities add most value during a product development program

Example Risk Mitigation Plan for A Software Build

CR #	CR Title	Source of Risk or Uncertainty	Risk	Impact	Score	Mitigation	Priority Development	In-Depth Review	Proto-type	Find and Fix
1	Change Request 1	Concept Maturity	9	9	81	Early proof of concept	Yes	Yes	No	No
2	Change Request 2	Requirements Quality	3	9	27	Review with IPT	Yes	Yes	No	Yes
3	Change Request 3	Concept Maturity	1	9	9	Functional Model	Maybe	Maybe	Maybe	No
4	Change Request 4	Supplier Capability	3	1	3	In-depth review with Supplier, Find and Fix	No	Yes	No	Yes
5	Change Request 5	Team Capability	1	9	9	In-depth review, Find and Fix	No	Yes	No	Yes
6	Change Request 6	Novelty and Complexity	3	1	3	Find and Fix	No	No	No	Yes
7	Change Request 7	Novelty and Complexity	9	9	81	Prototype, Find and Fix	Yes	No	Yes	Yes
8	Change Request 8	Team Capability	3	9	27	In-depth review	Maybe	Yes	No	No
9	Change Request 9	Team Capability	9	1	9	Find and Fix	No	No	No	Yes
10	Change Request 10	Novelty and Complexity	1	9	9	Find and Fix	Maybe	No	Maybe	Yes

Full Configuration Management or “CM Lite”?



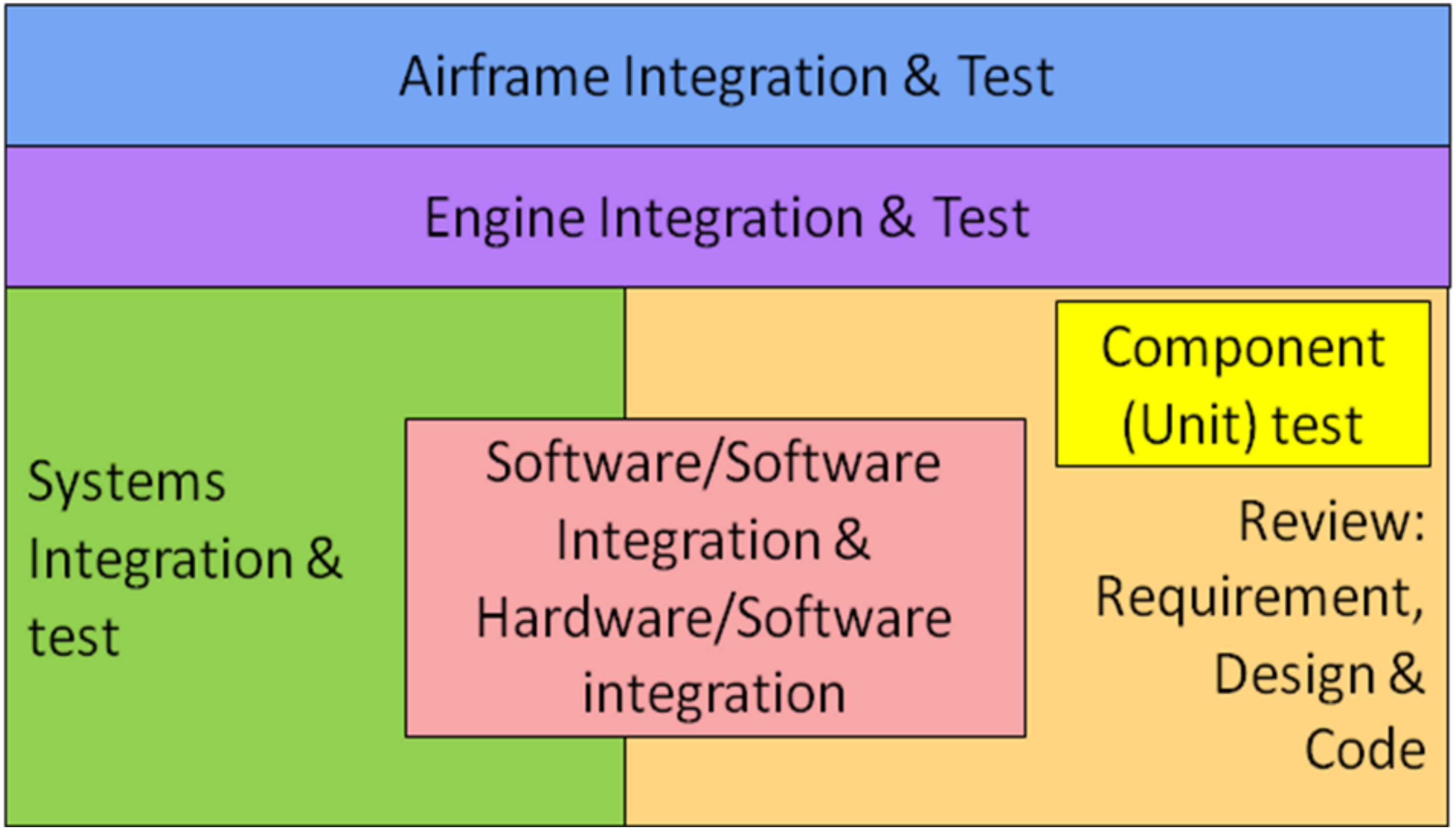
Assumptions

- Execution of changes through the Full Configuration Management System costs twice as much as through the CM Lite system.

- The configuration error rate is five times higher in the CM Lite system than in the Full Configuration Management System.

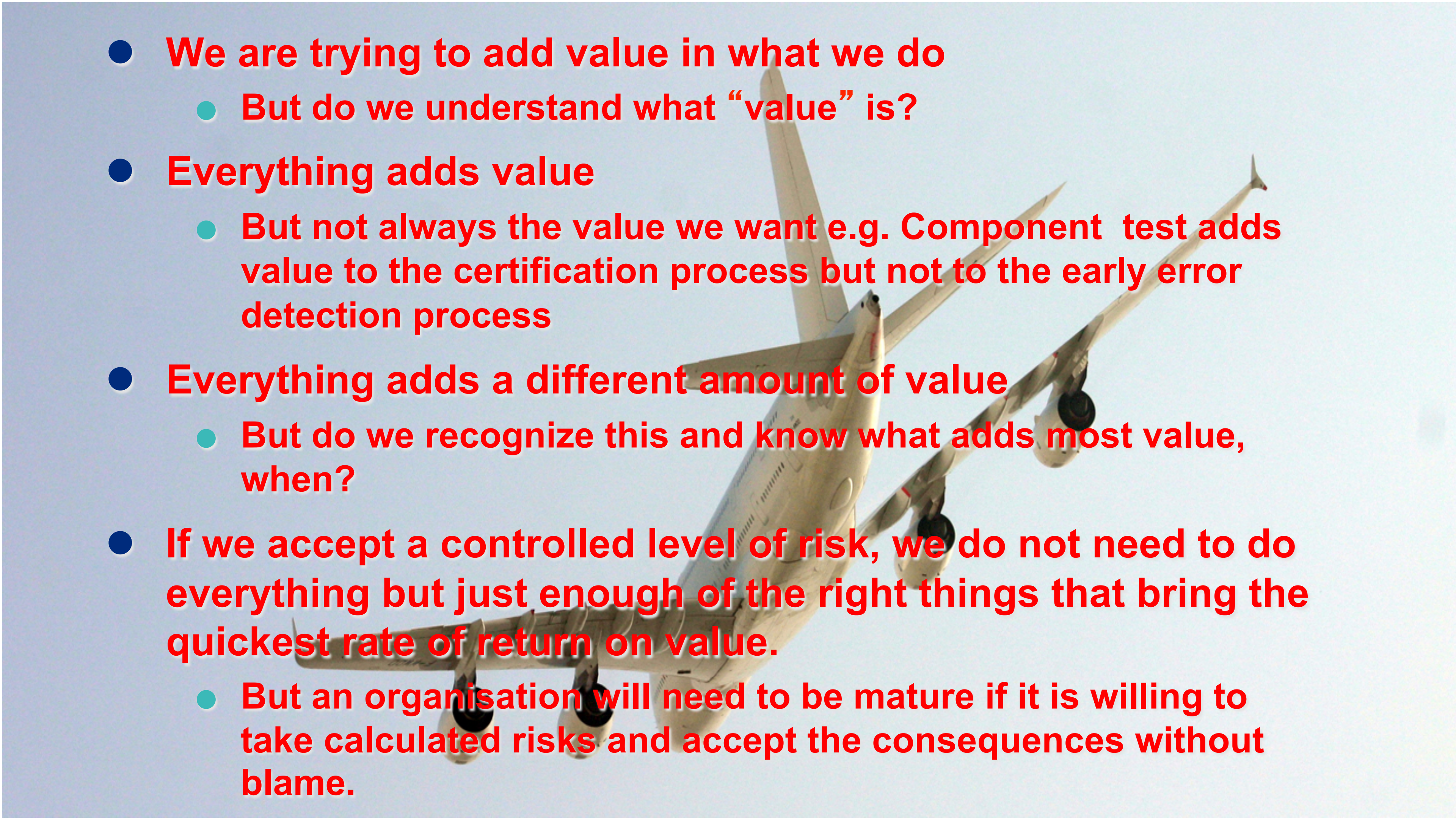
- The cost impact of an error escape increases by a factor of 3 mid-program and by a factor of 10 late-program compared to early in the program

V&V Methods and Error Types


Airframe Integration & Test
Engine Integration & Test
Systems Integration & test
Software/Software Integration & Hardware/Software integration
Component (Unit) test
Review: Requirement, Design & Code

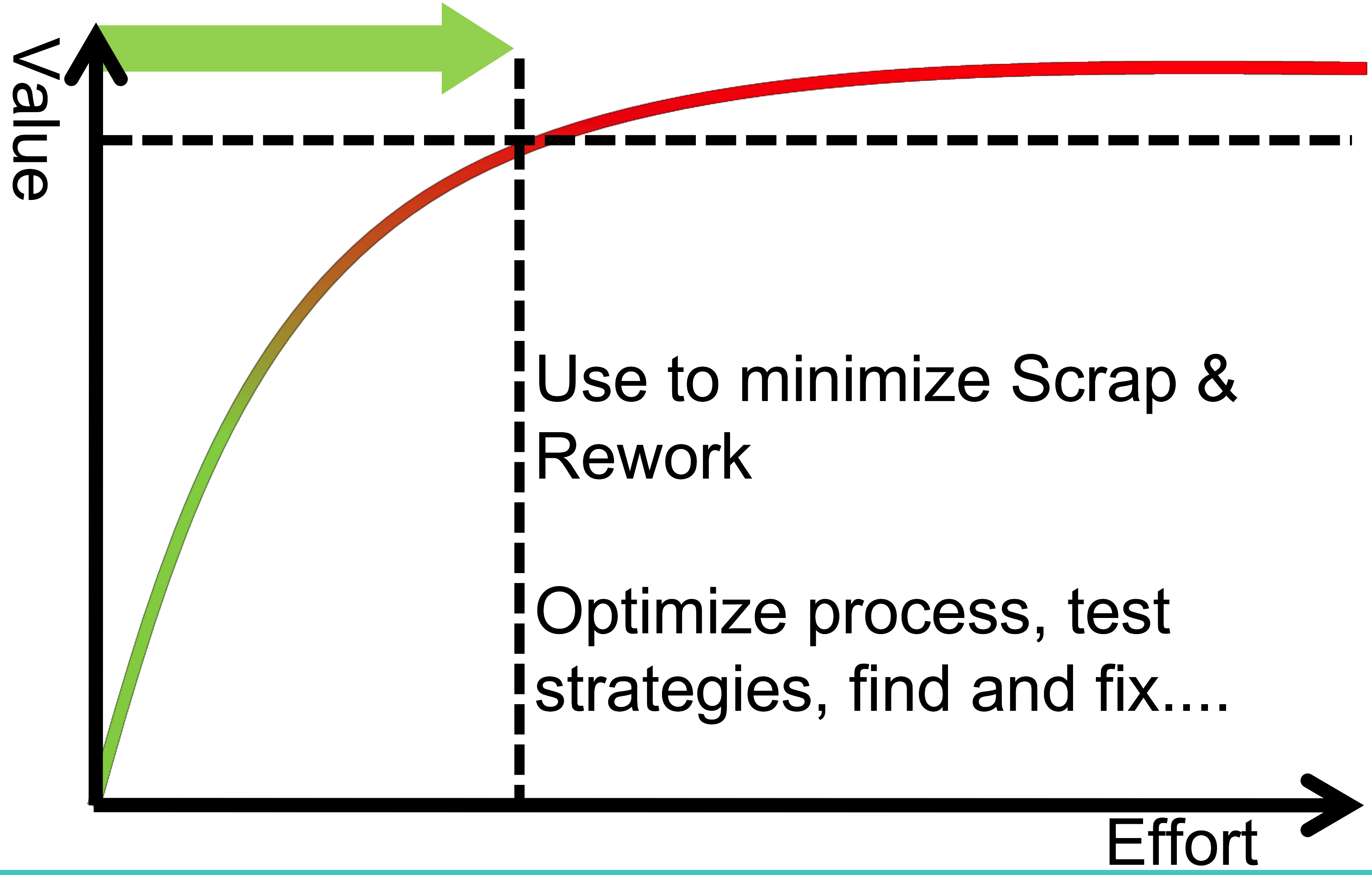
The areas shown represent the coverage of each V&V method (not to scale). The diagram was created from a correlation analysis of the various error classes found by each V&V method.

Value Based Management



- We are trying to add value in what we do
  - But do we understand what “value” is?
- Everything adds value
  - But not always the value we want e.g. Component test adds value to the certification process but not to the early error detection process
- Everything adds a different amount of value
  - But do we recognize this and know what adds most value, when?
- If we accept a controlled level of risk, we do not need to do everything but just enough of the right things that bring the quickest rate of return on value.
  - But an organisation will need to be mature if it is willing to take calculated risks and accept the consequences without blame.

Balancing Processes and Risks



Use to minimize Scrap & Rework

Optimize process, test strategies, find and fix....

Organizational Behaviors

- Success of Value Analysis will depend on the capability of the team to understand the trades

But:

- A risk-averse organisational culture will conflict with the principle of Value Based Management.
- An organisation has to be mature if it is willing to take calculated risks and accept the consequences without blame.

When not to use Probability Calculus

Safety Critical Systems!

- Processes that are used to ensure the safety of the system
- Strict limits on the probability of occurrence of events that could result in a hazardous condition
- All mitigations required to achieve this level of probability of occurrence have to be applied and cannot be tailored out

Where Next?

- OPTIONAL – “Opportunities for Process Tailoring by Identification Of Non-value-added Activities in Life”
- Examples for two processes used in System Development:

Process	Potential Consequences of not doing Enough	When to do Less
Project Management	Inadequate project management may result in poor or no requirements capture, missing review gates,work being planned and executed with insufficient time or resource, or delivery of a "Something in Time" solution with known deficiencies	Repeater job, simple task, low risk. Maturity of product not critical (e.g. prototype, R&T)
Sub-System Design	Inadequate sub-system design may result in the wrong requirements being set for commodity designs, resulting in a sub-system that has attributes that do not meet the design intent (performance, weight, cost, reliability, etc.) or show unacceptable emergent properties	Well defined architecture, high levels of re-use, instantiation of a Product Family, mature suppliers, configurable commodities.

Conclusions

- Systems Engineering Processes are normally written with the mindset of developing a completely new complex system
- When developing a simple system or making changes to a system, application of the full Systems Engineering processes is likely to result in excessive program costs and timescales
- Risk management and Probability Calculus offer a logical process for choosing which processes to apply and the level of rigor of those processes.
- Probability Calculus has a legacy of being used in legal cases to establish liability when mitigating actions could have been taken to avoid an undesirable event.
- Three examples of process tailoring are shown
- An exception is shown where Probability Calculus should not be used.
- The concept of Value Analysis is introduced