Validation, Verification and MER Case Study

Prof. Chris Johnson,
School of Computing Science, University of Glasgow.
johnson@dcs.gla.ac.uk
http://www.dcs.gla.ac.uk/~johnson

University of Glasgow

Introduction

- . Definitions and Distinctions
- Verification:
  - does it meet the requirements?
- Validation:
  - are the requirements any good?
- Testing:
  - process used to support V&V.

University of Glasgow

Definitions and Distinctions

- B.5.3.6 Verification and Validation.

This sub-process evaluates the products of other Software Modification sub-processes to determine their compliance and consistency with both contractual and local standards and higher level products and requirements. Verification and validation consists of software testing, traceability, coverage analysis and confirmation that required changes to software documentation are made. Testing subdivides into unit testing, integration testing, regression testing, system testing and acceptance testing.

MOD DEF STAN 00-66

Integrated Logistic Support: Part 3, Guidance for Software Support.

University of Glasgow

Definitions and Distinctions

- Misuse of terms?

A. The certification/validation process should confirm that hazards identified by hazard analysis, (HA), failure mode effect analysis (FMEA), and other system analyses have been eliminated by design or devices, or special procedures. The certification/validation process should also confirm that residual hazards identified by operational analysis are addressed by warning, labeling safety instructions or other appropriate means.

OSHA Regulations (Standards - 29 CFR) Nonmandatory guidelines for certification/validation of safety systems for presence sensing device initiation of mechanical power presses - 1910.217 App B

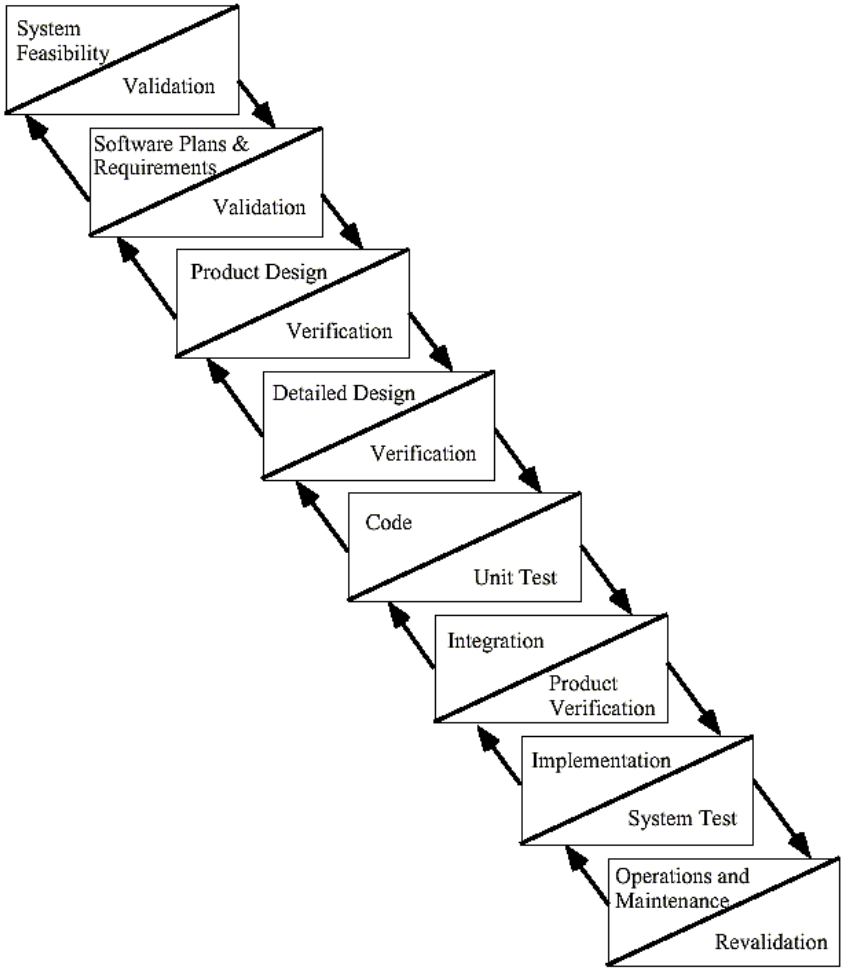
- More like verification?

University of Glasgow

Validation

- During design
  - external review before commission;
  - external review for certification.
- During implementation:
  - additional constraints discovered;
  - additional requirements emmerge.
- During operation:
  - were the assumptions valid?
  - especially environmental factors.
- Validate:
  - PRA's; development processes etc.

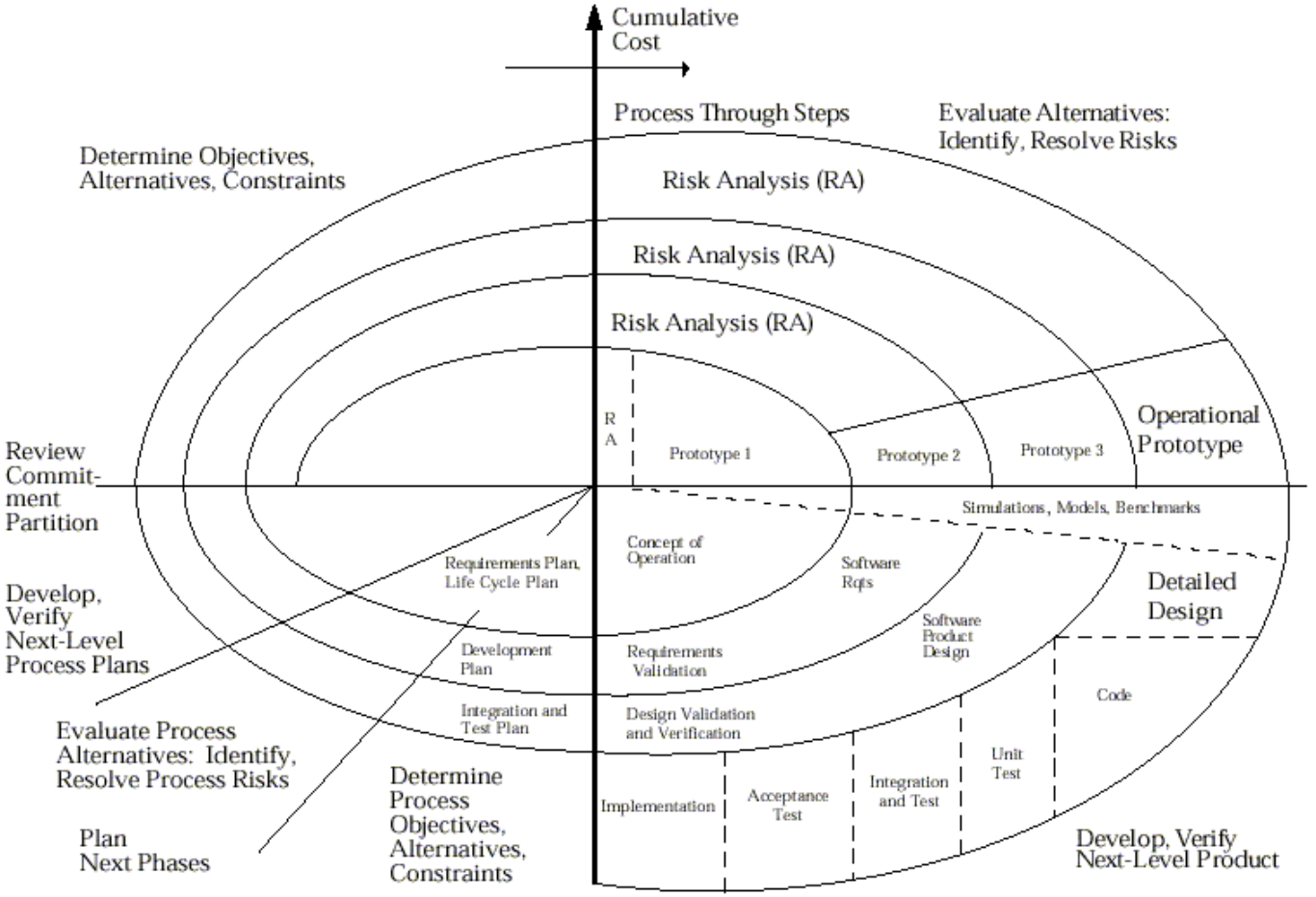
Waterfall Model



- Validation at start & end.

University of Glasgow

Validation: Spiral Model



- Validation more continuous.

Validation: IEC 61508 (Draft)

The following should be considered in an overall safety validation plan:

- Details of when the validation should take place.
- Details of who should carry out the validation.

Identification of relevant modes of system operation, including:

- preparation for use, including setting up and adjustment
- start up; teach; automatic; manual; semi-automatic
- steady-state operation; resetting;shutdown; maintenance
- reasonably foreseeable abnormal conditions.

University of Glasgow

Validation: IEC 61508 (Draft)

- Identification of the safety-related systems and external risk reduction facilities that need to be validated for each mode of the system before commissioning commences.
- The technical strategy for the validation, for example, whether analytical methods or statistical tests are to be used.
- The measures, techniques and procedures that shall be used to confirm that each safety function conforms with the overall safety requirements documents and the safety integrity requirements.
- The specific reference to the overall safety requirements documents.
- The required environment in which the validation activities are to take place.
- The pass/fail criteria.
- The policies and procedures for evaluating the results of the validation, particularly failures.

Validation: MOD DEF STAN 00-60

- D.4.1.6 Validation.

At the earliest opportunity support resource requirements should be confirmed and measurements should be made of times for completion of all software operation and support tasks. Where such measurements are dependent upon the system state or operating conditions, averages should be determined over a range of conditions. If measurements are based on non-representative hardware or operating conditions, appropriate allowances should be made and representative measurements carried out as soon as possible. The frequency of some software support tasks will be dependent upon the frequency of software releases and the failure rate exhibited by the software.

MOD DEF STAN 00-66

Integrated Logistic Support: Part 3, Guidance for Software Support.

Validation: MOD DEF STAN 00-60

- D.4.1.6 Validation (cont.)

Measurements of software failure rates and fault densities obtained during software and system testing might not be representative of those that will arise during system operation. However, such measurements may be used, with caution, in the validation of models and assumptions. For repeatable software engineering activities, such as compilation and regression testing, the time and resource requirements that arose during development should be recorded. Such information may be used to validate estimates for equivalent elements of the software modification process. For other software engineering activities, such as analysis, design and coding, the time and resource requirements that arose during development should be recorded. However, such information should only be used with some caution in the validation of estimates for equivalent elements of the software modification process. The preceding clauses might imply the need for a range of metrics

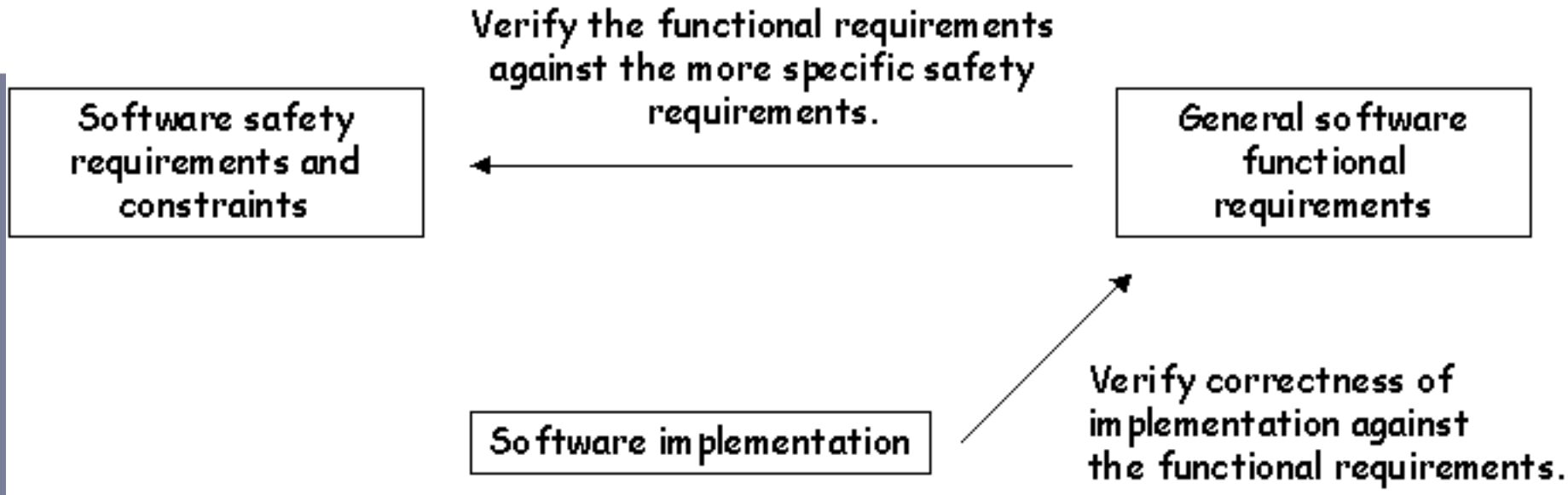
MOD DEF STAN 00-66, Integrated Logistic Support: Part 3, Guidance for Software Support.

Validation: Summary of Key Issues

- Who validates validator?
  - External agents must be approved.
- Who validates validation?
  - Clarify links to certification.
- What happens if validation fails?
  - Must have feedback mechanisms;
  - Links to process improvement?
- NOT the same as verification!

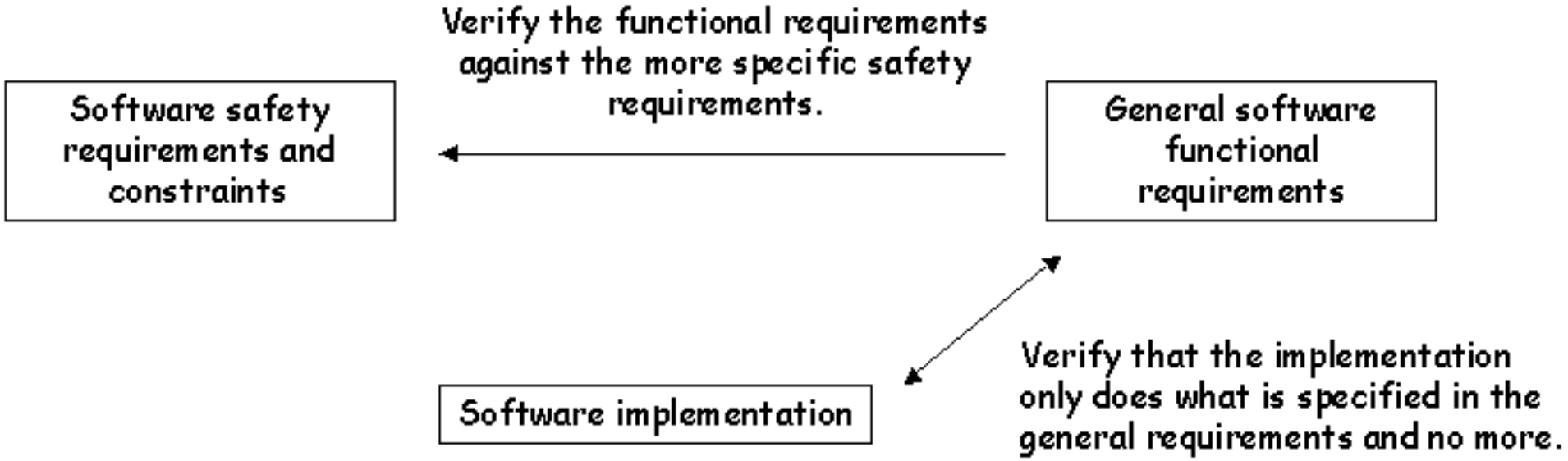
University of Glasgow

Verification: Leveson's Strategies



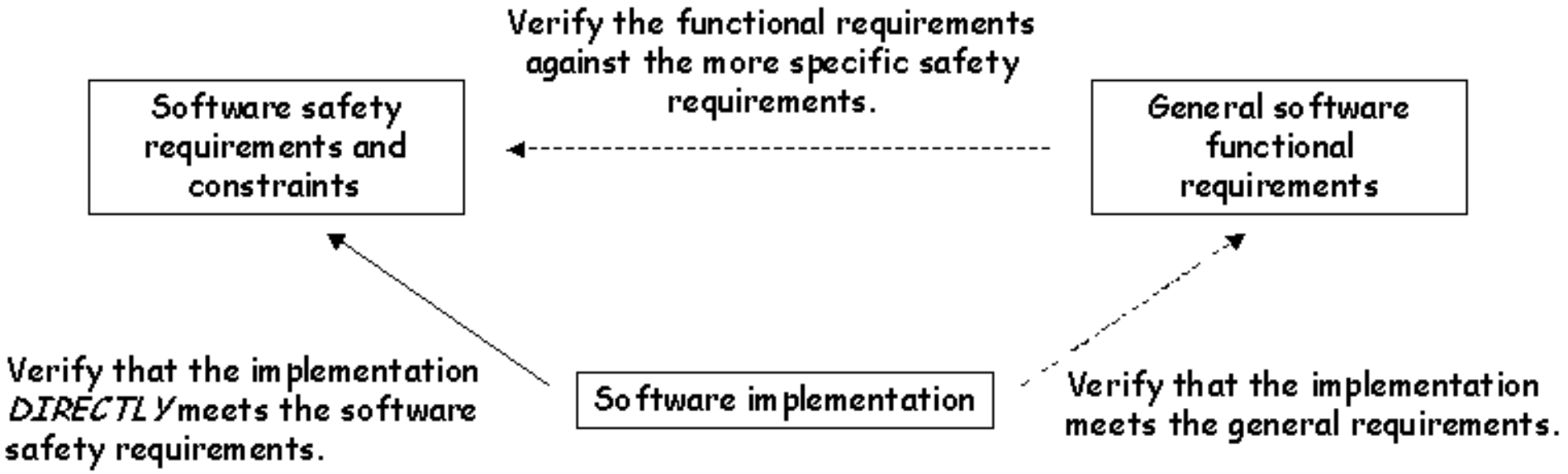
- Show that functional requirements
  - are consistent with safety criteria ?
- Implementation includes hazards not in safety/ functional requirements.

Verification: Leveson's Strategies



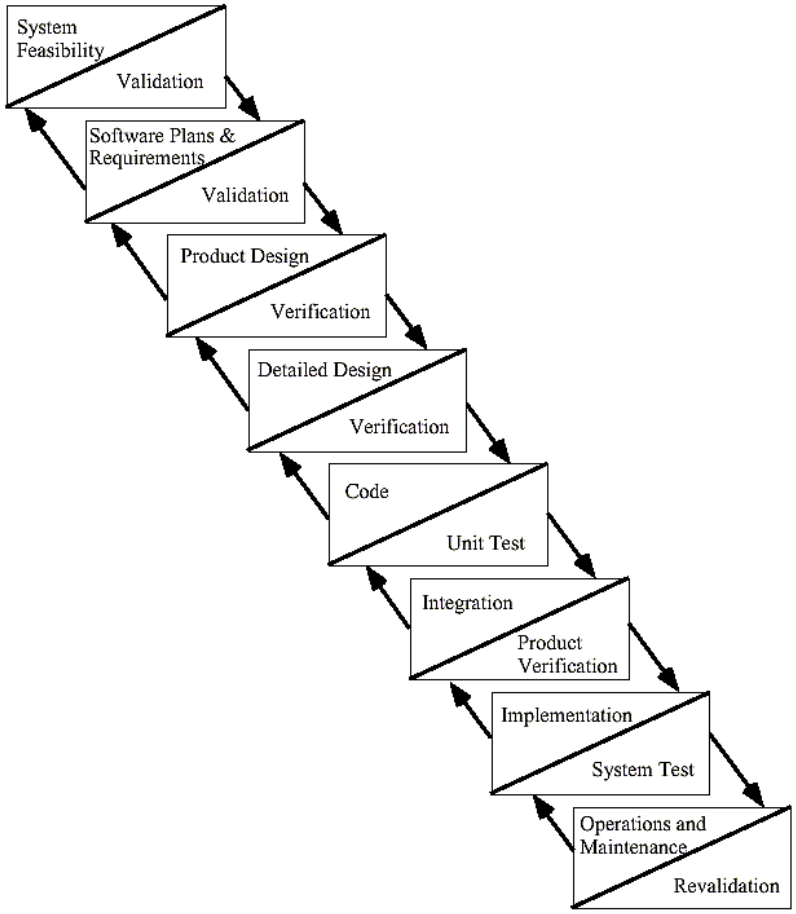
- Show that implementation is
  - same as functional requirements?
- Too costly/time consuming all safety behaviour in specification?

Verification: Leveson's Strategies



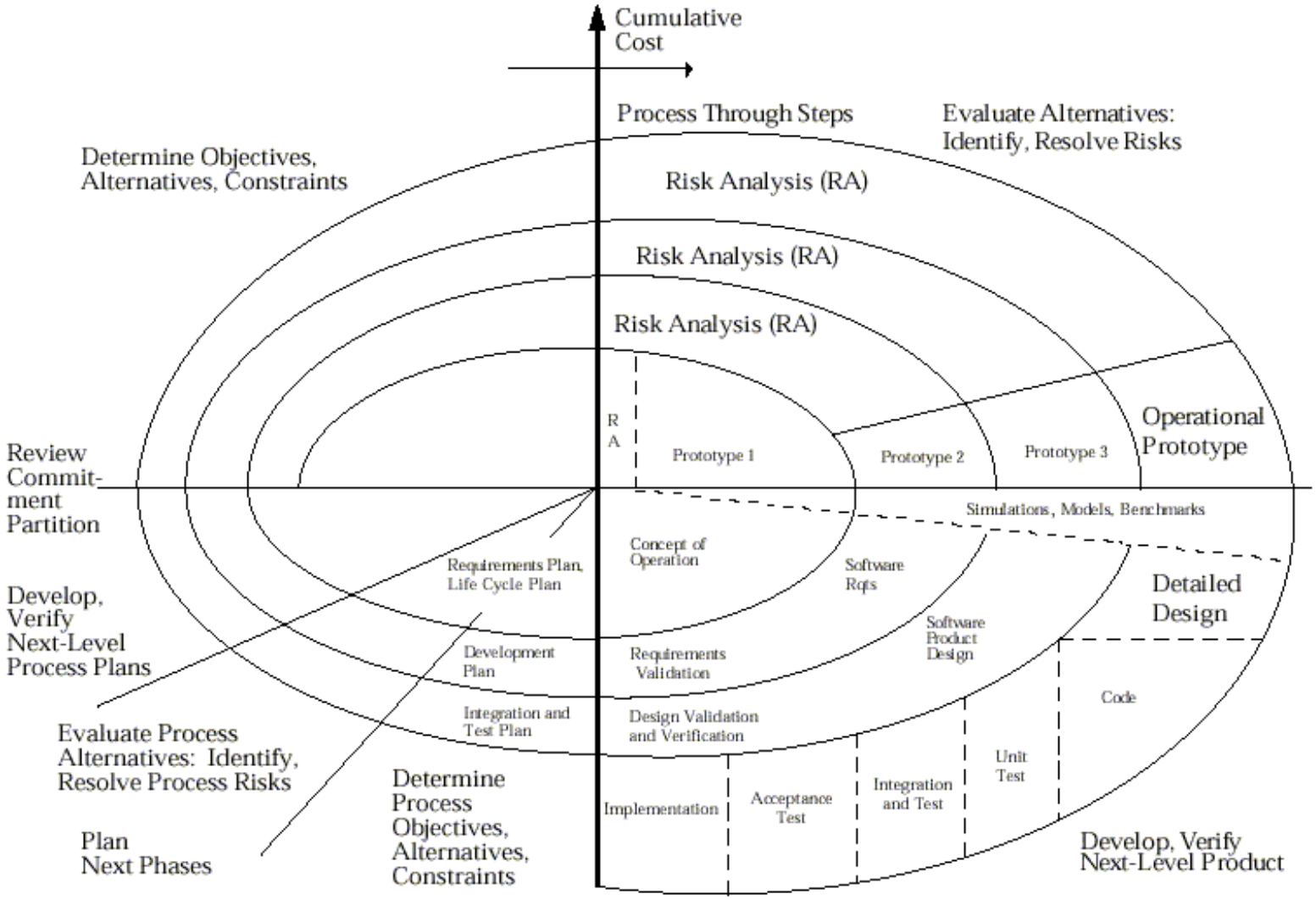
- Or show that the implementation
  - meets the safety criteria.
- Fails if criteria are incomplete...
  - but can find specification errors.

Verification: Lifecycle View



- Sseveral stages in waterfall model.

Verification: Lifecycle View



University of Glasgow

Verification

- Verification as a catch-all?
- A recurrent cost, dont forget...
  - verification post maintenance.
- Verification supported by:
  - determinism (repeat tests);
  - separate safety-critical functions;
  - well defined processes;
  - simplicity and decoupling.

Verification

- D.5.1 Task 501 Supportability Test, Evaluation & Verification

- D.5.1.1 Test and Evaluation Strategy

Strategies for the evaluation of system supportability should include coverage of software operation and software support. Direct measurements and observations may be used to verify that all operation and support activities - that do not involve design change - may be completed using the resources that have been allocated.During the design and implementation stage measurements may be conducted on similar systems, under representative software modification activity is broadly similar to software development the same monitoring mechanism might be used both pre- and post-implementation. Such a mechanism is likely to be based on a metrics programme that provides information, inter alia, on the rate at which software changes are requested and on software productivity.

MOD DEF STAN 00-66

- Integrated Logistic Support: Part 3, Guidance for Software Support.

Verification

D.5.1.3 Objectives and Criteria.

System test and evaluation programme objectives should include verification that all operation and support activities may be carried out successfully –within skill and time constraints - using the PSE and other resources that have been defined. The objectives, and associated criteria, should provide a basis for assuring that critical software support issues have been resolved and that requirements have been met within acceptable confidence levels. Any specific test resources, procedures or schedules necessary to fulfil these objectives should be included in the overall test programme. Programme objectives may include the collection of data to verify assumptions, models or estimates of software engineering productivity and change traffic.

MOD DEF STAN 00-66, Integrated Logistic Support: Part 3, Guidance for Software Support.

Verification

- D.5.1.4 Updates and Corrective Actions.

Evaluation results should be analyzed and corrective actions determined as required. Shortfalls might arise from:

- Inadequate resource provision for operation and support tasks.
- Durations of tasks exceeding allowances.
- Software engineering productivity not matching expectations.
- Frequencies of tasks exceeding allowances.
- Software change traffic exceeding allowances.

Corrective actions may include: increases in the resources available; improvements in training; additions to the PSE or changes to the software, the support package or, ultimately, the system design. Although re-design of the system or its software might deliver long term benefits it would almost certainly lead to increased costs and programme slippage.

MOD DEF STAN 00-66 Integrated Logistic Support: Part 3, Guidance for Software Support.

University of Glasgow

Verification: Summary of Key Issues

- What can we afford to verify?
  - Every product of every process?
  - MIL HDBK 338B...
- Or only a few key stages?
- If the latter, do we verify :
  - specification by safety criteria?
  - implementation by safety criteria?
  - or both...

University of Glasgow

Verification: Summary of Key Issues

- Above all....
- Verification is about proof.
- Proof is simply an argument.
- Argument must be correct but
  - not a mathematical `holy grail'...

SPIRIT Case Study: Sol 18

- Spirit Mars Exploration Rover (MER).
  - landed on Mars 04:35 UTC 4th January 2004.
  - mission for 90-solar days on Mars.
- Mission interruption:
  - started on Sol 18 (21st Jan);
  - blocks tasks requiring flight computer memory;
  - Sol 33 (6th Feb) normal ops were restored.

University of Glasgow

SPIRIT Case Study: Sol 18 + 19

(S18) 09:00 High-Gain communications start.

(S18) 09:11 uplink errors detected.

signal was unexpectedly lost.

(s18) 11:20 command High-Gain priority comms

no response detected from MER.

(S19) 09:00 no High Gain signal should trigger:

– low gain antenna comms,

(S19) 11:00 still no signal.

University of Glasgow

SPIRIT Case Study: Sol 20

- Ground teams conclude:
  - system level fault before Sol 19.
- Uncertainty but low-bit rate messages received:
  - Suggest rover processing data, not in sleep mode;
  - Autonomous shutdowns failing;
  - Concern that Spirit exhausts battery or overheats.
- Possibility rover in a reboot loop:
  - Boot enables subsequent programs from RAM;
  - Fault occurs before initial commands complete;
  - Triggers attempt to rerun initial commands…

University of Glasgow

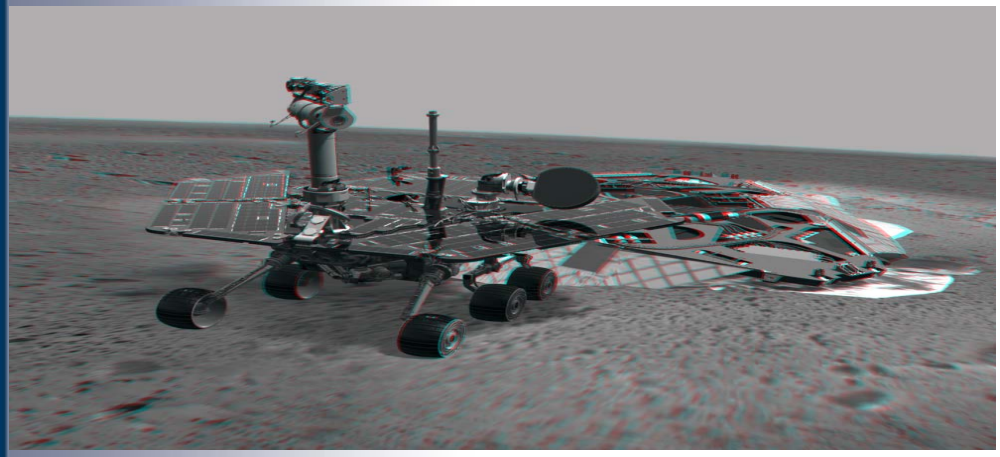
SPIRIT Case Study: Sol 20

- Hardware fault or bug in flash EEPROM boot?
  - Designers had considered this:
  - reboot without accessing the EEPROM.
- Look at file system config in VxWorks OS.
  - Reboot, caused by number of files in flash:
  - MER calibrates instruments & old cruise data.
- Parameter in dosFsLib module:
  - assigns ‘overflow’ data to system memory;
  - Compounded by memPartLib module;
  - Suspend tasks using memory & reset flight computer.

University of Glasgow

SPIRIT Case Study: Sol 20

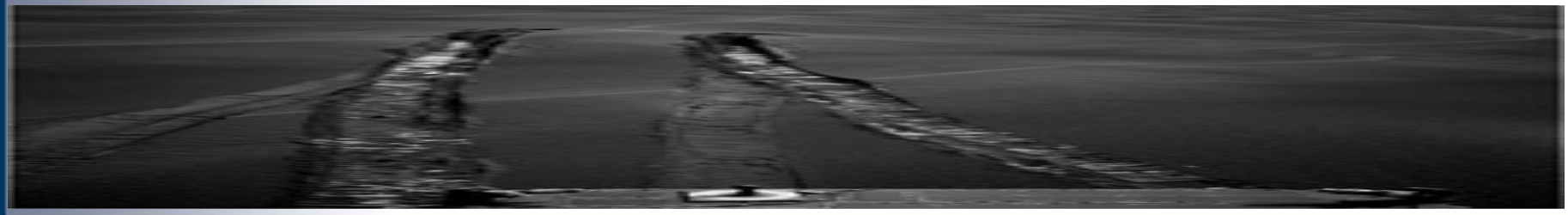
- Too risky to revise dosFsLib/memPartLib
- Potential solutions:
  - manually reallocate system memory;
  - Delete unnecessary directories and files;
  - Over time, create new file system.



University of Glasgow

1. *'Clarify and Validate the Problem'.*

- Initial investigations identify symptoms.
  - 'work arounds' cannot update file management.
- Longer-term analysis:
  - IV&V missed configuration management issues?
  - Teams had set up and used COTS OS;
  - Particular challenge for IV&V;
  - Interface between bespoke software and COTS without commercial source code.



University of Glasgow

2. *'Break down problem & identify performance gaps'.*

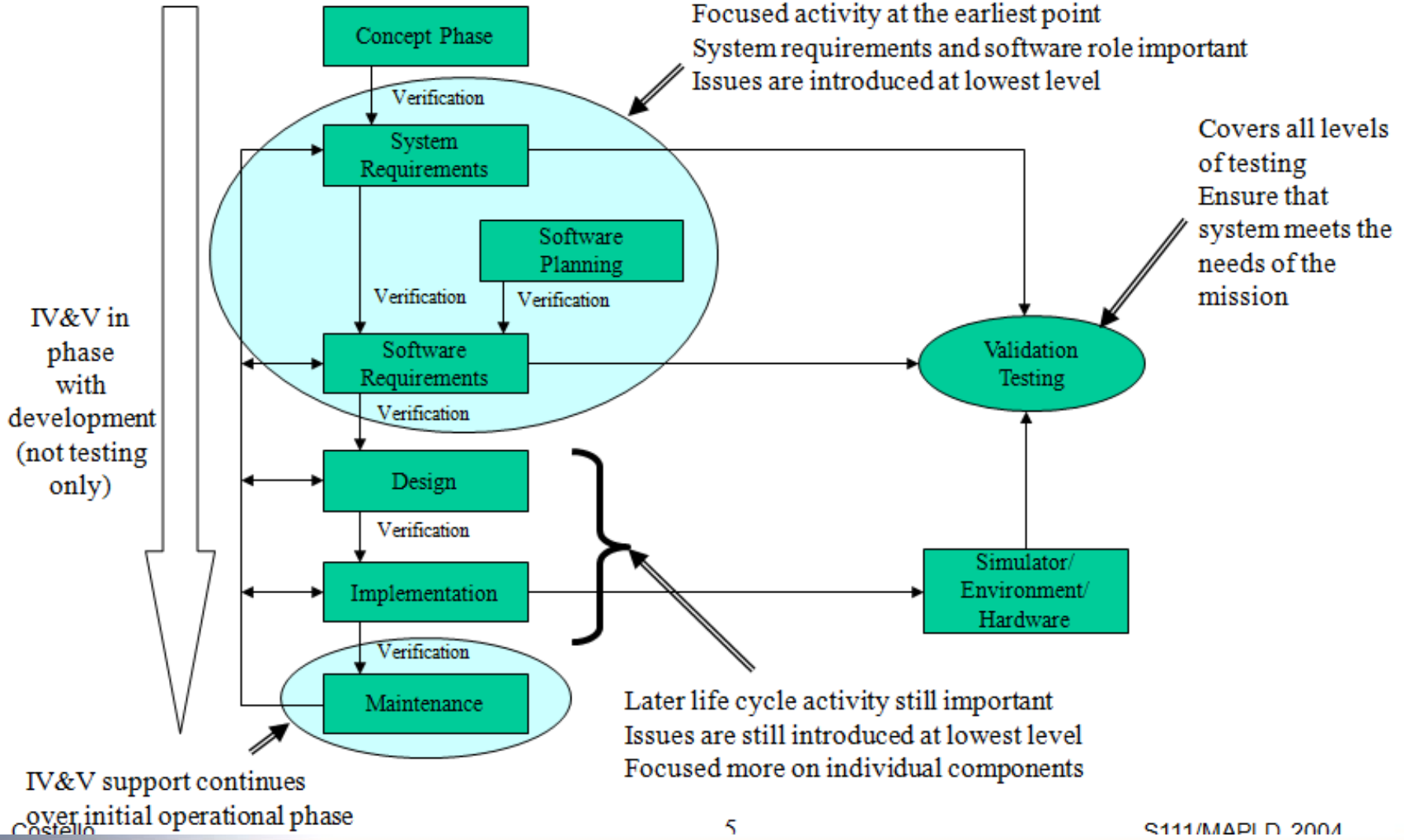
"IV&V Facility support included requirements analysis, interface design evaluations, behavioral architecture analysis, code analysis, test analysis of MER flight software (spacecraft & rovers)...

Upon completion, the IV&V Team had identified 1018 issues of which 347 were mission critical or catastrophic and 6 possible risks to the project that could result in mission failure".

(NASA, 2004)

University of Glasgow

2. *'Break down problem and identify performance gaps'.*



Acknowledgement: NASA IV&V Office

University of Glasgow

3. *"Set Improvement Target".*

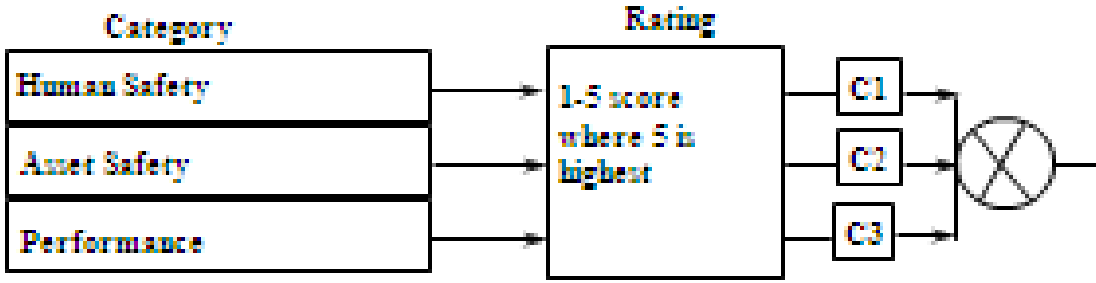
- Risk based MER IV&V planning begins June 2001:
  - File routines critical to system architecture;
  - consequences of bug jeopardize the mission.
  - some elements of OS file handling highly complex.
- But systems relatively mature so risk low?
- Step 3: improve IV&V risk assessment processes?
- Not explicit in lessons learned reports?

IV&V Facility

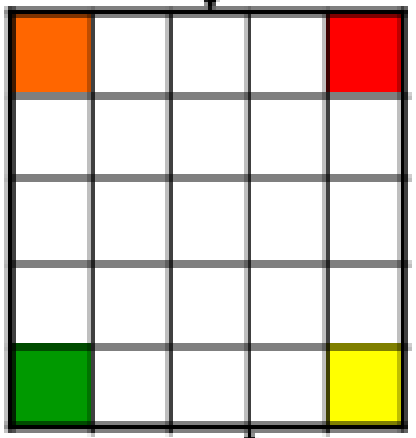
Scope Determination
Software Integrity Level Assessment Process

For each Software Component:

Criticality:

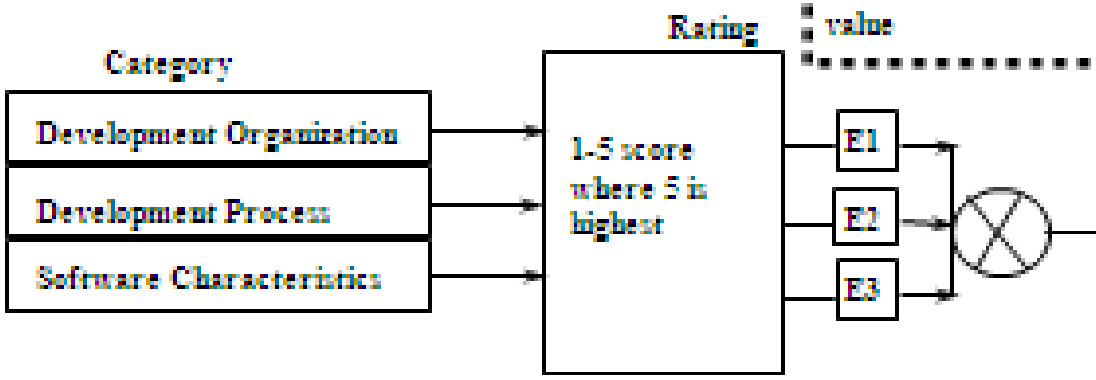


Final values plotted on a 5x5 matrix for reference. Values are also used to cross-reference a task matrix.



Each factor has an associated weight applied before being combined into a final value

Error Potential:



Task selection is from a standardized list of tasks. Allocation is based on criticality value and on error potential value individually.

University of Glasgow

4. ‘Determine Causes & Contributory Factors’.

- Why were these OS routines seen as low risk?
  - IV&V estimate 10 fulltime staff needed;
  - less than 5 allocated in overall project budget.
- Forces IV&V to further prioritize activities;
  - Less time to question original risk assessments;
  - Did not develop full test suites for compliance;
  - Instead focus on scenarios – are these sufficient?
- IV&V identified requirement and test completeness as highest project risk.

4. ‘Determine Causes & Contributory Factors’.

- Further concern about code stability:
  - changes continued until upload;
  - 10% of all code affected by the final update;
  - Including 10% of file handling routines
- 2nd December: IV&V delay software upload.
- 5th December, software uploaded;
  - “IV&V has a less optimistic view of the requirements discovery than does the project” (Costello, 2004).
  - system memory usage; risk tracking, issue tracking, code analysis, requirements analysis, test analysis, code complexity and code stability.

University of Glasgow

5. ‘Develop Countermeasures’.

- Afterwards, IV&V funds increased:
  - Testing not just at the systems level;
  - also on components in ‘high risk’ operations;
  - in accordance with NASA NPD 8730.4A.
- Develop countermeasures (2):
  - Must identify critical software develop. artefacts
  - Lack of explicit software development lifecycle;
  - Perceived lack of MER requirements docs;
  - Knock-on effect undermined IV&V reqs testing;

University of Glasgow

5. ‘Develop Countermeasures’.

- Development teams:
  - Focus on successive software problems;
  - Stem from lack of coherent requirements;
  - Instead of clarifying the requirements first;
  - Tests mitigate risks of each software problem;
  - But if tests miss requirement then risk continues.
- Compounded by ambitious test schedules
  - Studies incomplete by time of upload.
  - IV&V start research into new testing techniques
  - for projects with poor requirements documentation!

University of Glasgow

6. ‘Seeing the Countermeasures Through’

- Development and IV&V teams:
  - Work closer in early software lifecycle;
  - agree on the overall testing philosophy;
  - Agree need for formal software processes.
  - Put configuration management into development teams
- BUT dynamic nature of space systems:
  - hard to draft detailed software requirements
  - Eg when uncertainty over hardware platforms.

University of Glasgow

6. ‘Seeing the Countermeasures Through’

- Contractual/Independence issues:
  - IV&V can’t work directly with subcontractor?
  - Could not access development database;
  - Recorded data on completed tests.
- Contractual/Independence issues.
  - engagement with development teams and
  - need for independence in testing and validation.
- IV&V action plan :
  - reinforced existing policy, (NPD 8730.4),

University of Glasgow

Conclusions

- . Definitions and Distinctions
- Verification:
  - does it meet the requirements?
- Validation:
  - are the requirements any good?
- Testing:
  - process used to support V&V.

University of Glasgow

Any Questions…