Reliability Growth in Software Development Processes: A BBN Analyses of the Concept

International Probabilistic Safety Assessment and Management Conference PSAM 9, 18 – 23 May 2008

Mario Brito: mpb2o07@noc.soton.ac.uk

National Oceanography Centre, Southampton
UNIVERSITY OF SOUTHAMPTON AND NATURAL ENVIRONMENT RESEARCH COUNCIL

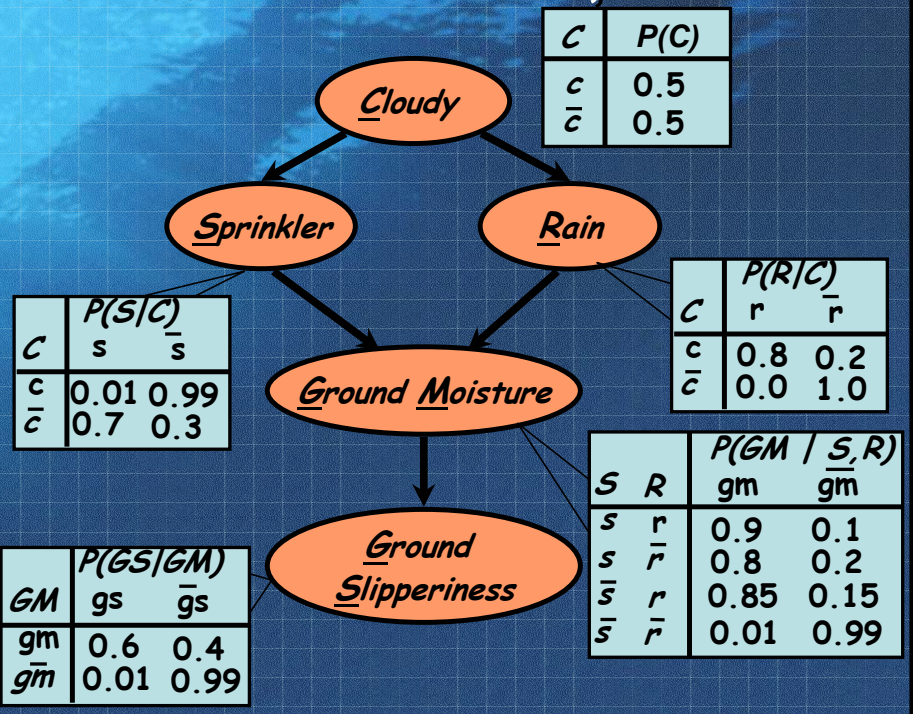
www.noc.soton.ac.uk

Our motivation… What is going on in terms of using BBNs to support software reliability arguments?

- There two schools of thought when it comes to predicting software reliability:
  - Product based. Software reliability growth models Musa (1987) and Littlewood (1991) and Statistical testing methods Miller et. al. (1992).
  - Process based. Software quality and safety standards. ICE61508, DEF 0055, DO 178 and ISO 9001 CMMM.
- Researchers have developed and applied BBN based applications on both fields Ganesh et. al. (2007) and Hall et . al. (1992).
- Our literature survey tell us that the use of BBNs to estimate the quality/reliability/integrity of the software development process was first proposed by Hall, et al (1992). Since then tens of publications were made available on this topic, Fenton et. al. (2004), Littlewood et. al. (2006), Acuna et. al. (2002), Panzakar et. al. (2005), Gran (2002), Cockram (2001) and Xuan (2005).

Mario Brito: mpb2o07@noc.soton.ac.uk

Before we move on… How do we build BBNs, What do they represent?



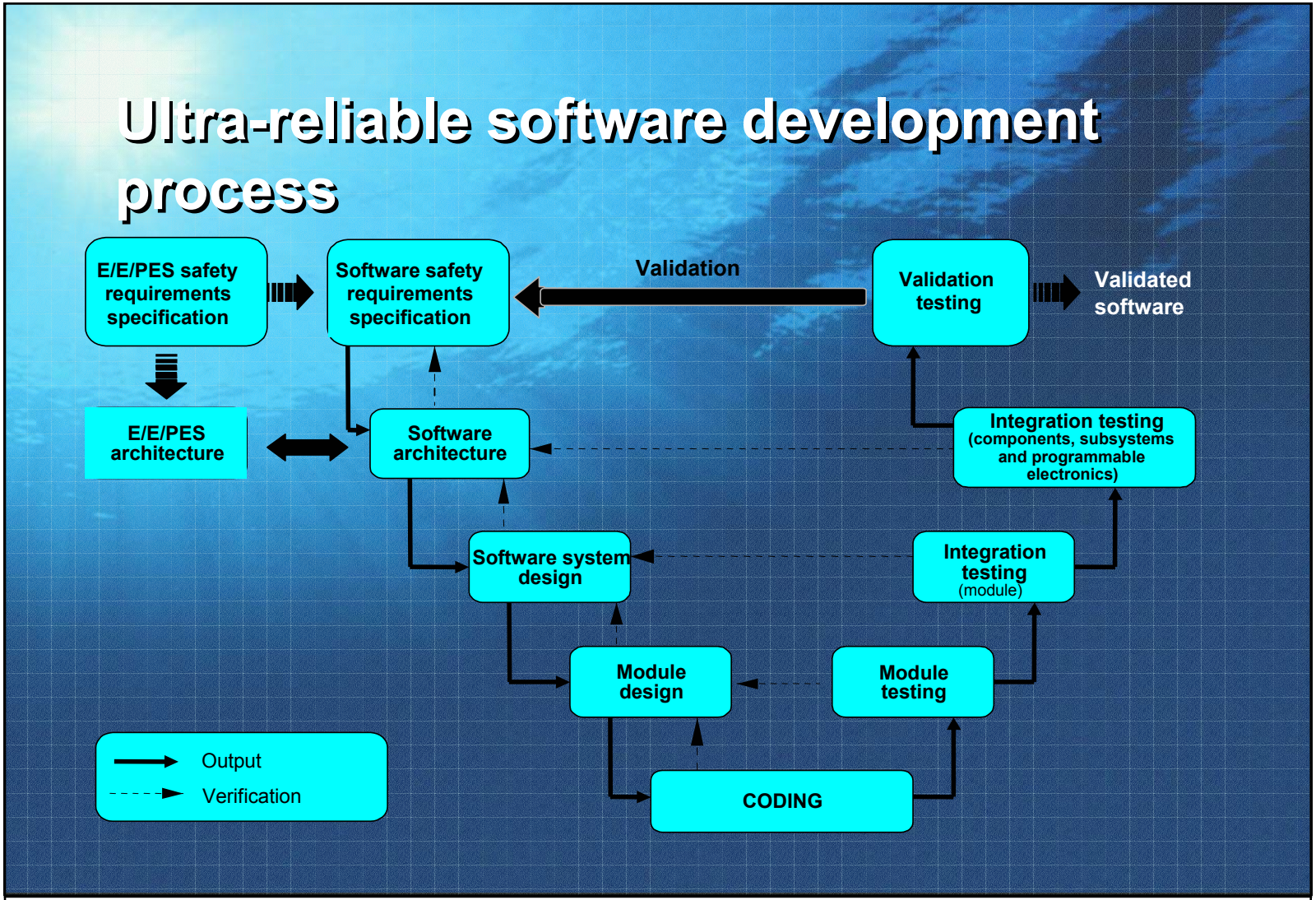
Network structure:

- Directed acyclic graph (DAG).
- Nodes - random vars.
- Edges – causal (direct) influence.
- Defines a unique distribution in the factored form:

$$P(C,S,R,GM,GS) = P(C)P(S|C)P(R|C)P(GM \mid S,R)P(GS \mid GM)$$

Mario Brito: mpb2o07@noc.soton.ac.uk

Ultra-reliable software development process



Mario Brito: mpb2o07@noc.soton.ac.uk

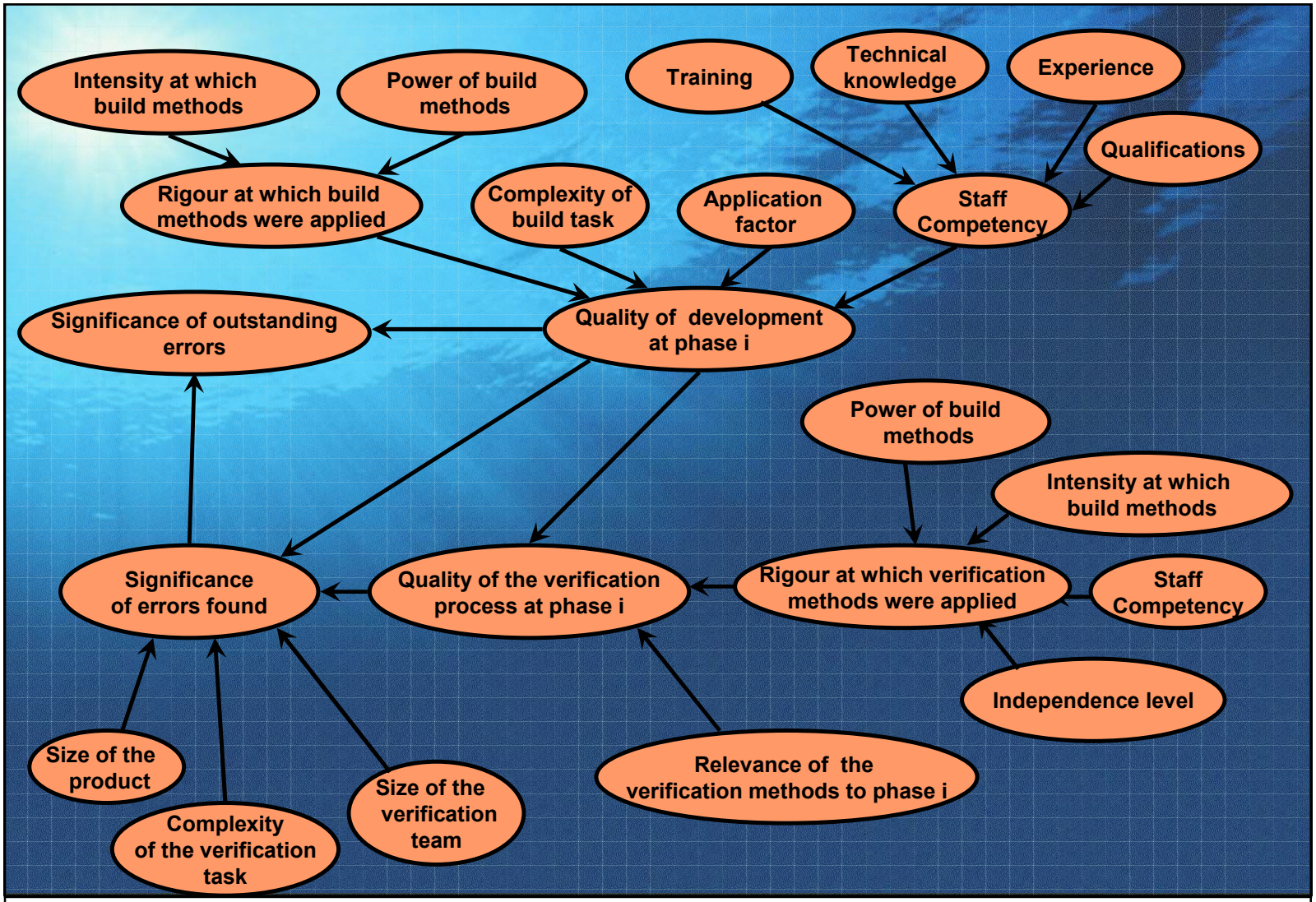
Ultra-reliable software development process

Technique/Measure	SIL 1	SIL 2	SIL 3	SIL 4
1 Computer-aided specification tools	R	R	HR	HR
2a Semi-formal methods	R	R	HR	HR
2b Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	---	R	R	HR

a) The software safety requirements specification will always require a description of the problem in the natural language and any necessary mathematical notation that reflects the application.
b) The table reflects additional requirements for specifying the software safety requirements clearly and precisely.
c) Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/ measures are indicated by a letter following by a number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

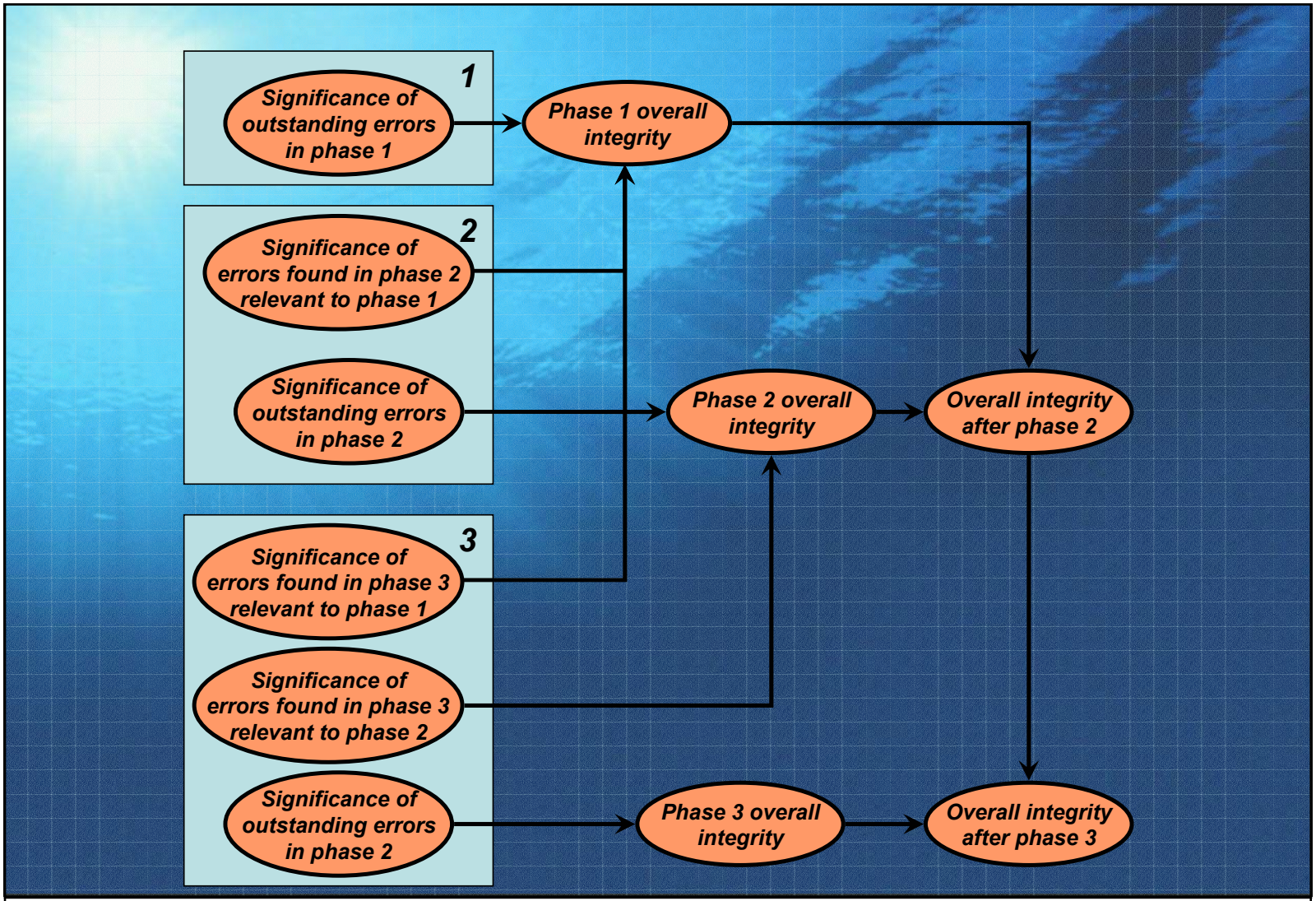
Software safety requirements specification (see part 3 section 7.2)

Mario Brito: mpb2o07@noc.soton.ac.uk



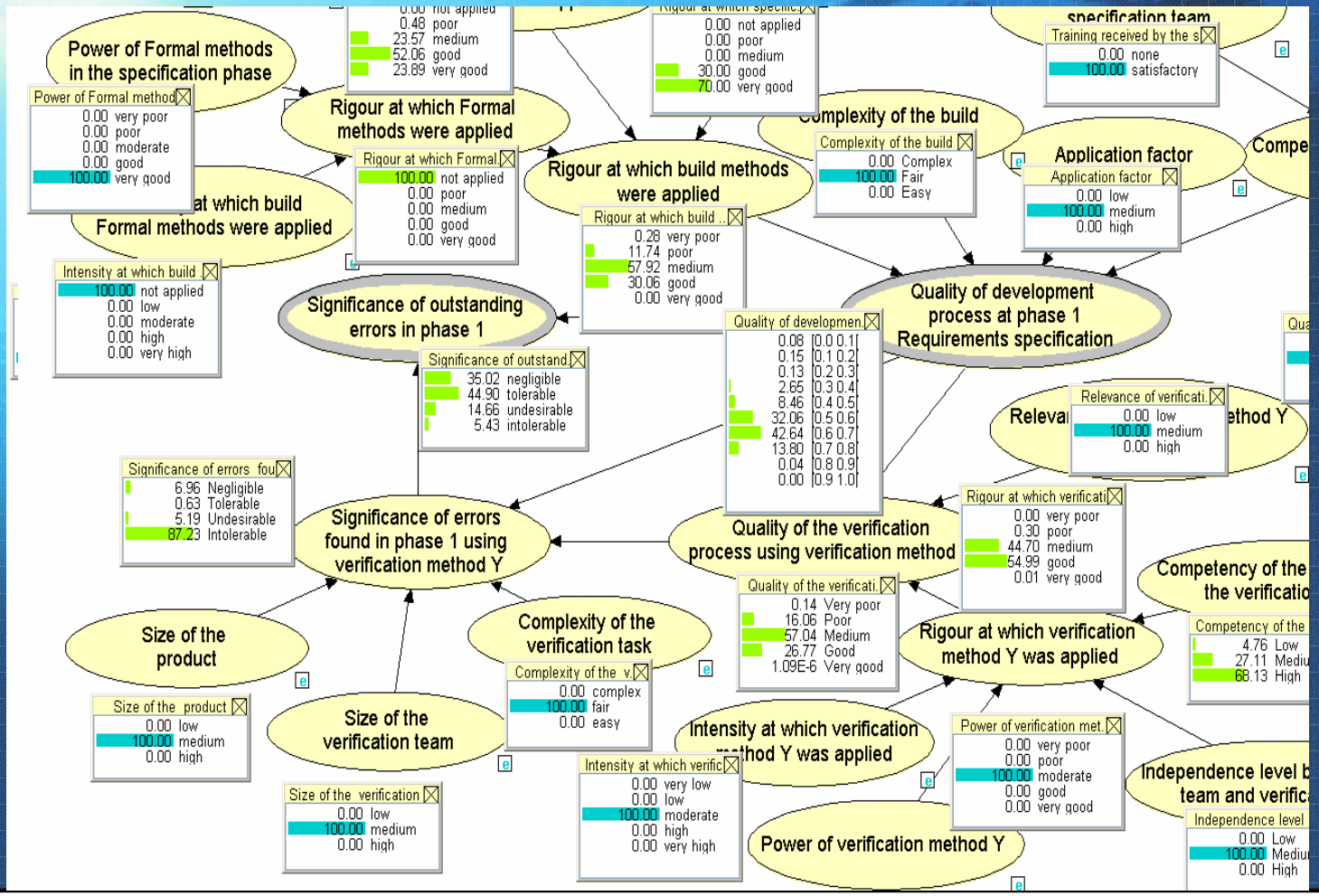
Mario Brito: mpb2o07@noc.soton.ac.uk

www.noc.soton.ac.uk



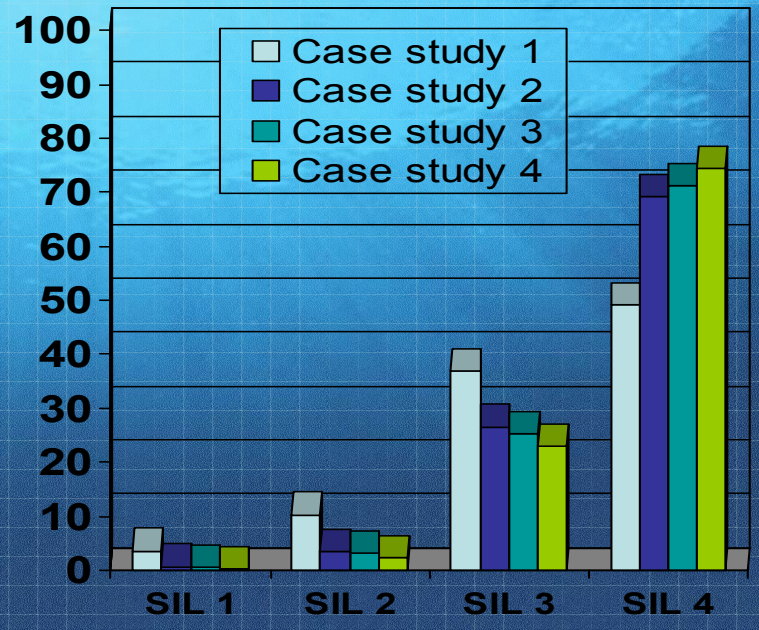
Mario Brito: mpb2o07@noc.soton.ac.uk

Model implemented in Hugin 6.5



Mario Brito: mpb2o07@noc.soton.ac.uk

Case study 1 – Formal methods not applied; Case study 2 – Formal methods applied at a moderate intensity Case study 3 – Formal methods applied at a high intensity; Case study 4 – Formal methods applied at a very high intensity. For all cases semi-formal methods have been applied at a low intensity



Case study	Confidence (%)			
	SIL 1	SIL 2	SIL 3	SIL 4
1	100	96	86	49
2	100	99	96	69
3	100	99	94	71
4	100	100	98	75

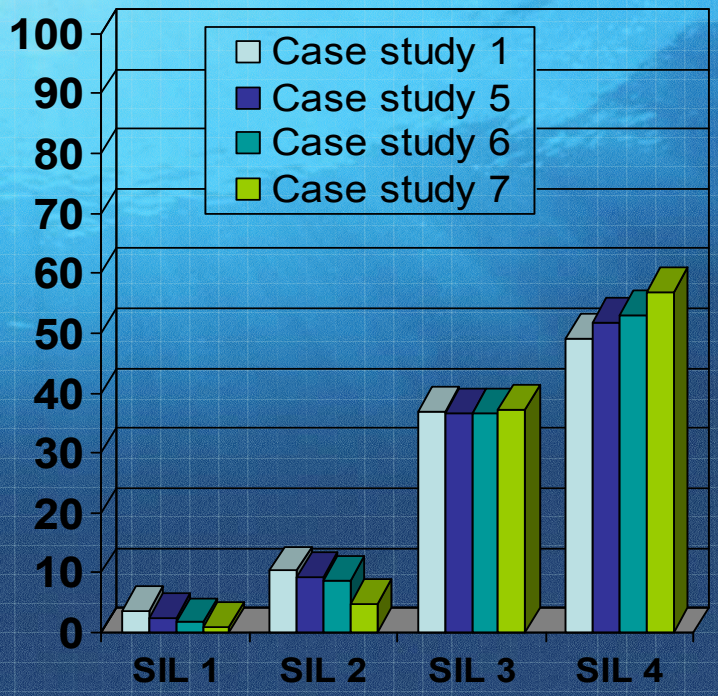
36% increase in belief that the criticality of the outstanding errors is 'negligible'

41% increase in belief that the Quality of the development process is between 60% & 70%

Mario Brito: mpb2o07@noc.soton.ac.uk

www.noc.soton.ac.uk

Case study 1 – Formal methods not applied in phase 1; Case study 5 – Improvement of the review process through the use of a better verification technique; Case study 6 – Improvement of the review process though the use of an independent team, better qualified and more experienced; Case study 7 –Improvement of the review process in phase 2



Case study	Confidence (%)			
	SIL 1	SIL 2	SIL 3	SIL 4
1	100	96	86	49
5	100	98	89	52
6	100	98	90	53
7	100	99	94	57

8% increase in belief that the software product complies with SIL 3

Mario Brito: mpb2o07@noc.soton.ac.uk

www.noc.soton.ac.uk

Conclusions and Outlook

- BBNs facilitate the modelling of complex problems. It injects transparency into the problem; the reasoning becomes compact and easy to comprehend. It allow us to highlight emergent properties.
- Reliability growth in software development processes captures the effect of the review processes on the integrity claim of any phase of the development lifecycle, including earlier phases.
- This phenomenon is observed in any product design, not only software.
- Tools such as HUGIN, NETICA, MSBNx, XBaies2, SEAMED, AGENA would allow the modeling the proposed network.
- Developing the network structure is not as hard as populating the node probability tables. Formal methods for eliciting probabilities reduce expert bias. In addition, one could use either mathematical (e.g. linear weighted opinion pool or log weighted opinion pool) or behavioural methods to aggregate expert probability judgments.

Mario Brito: mpb2o07@noc.soton.ac.uk

Thanks!

Mario Brito: mpb2o07@noc.soton.ac.uk

www.noc.soton.ac.uk