

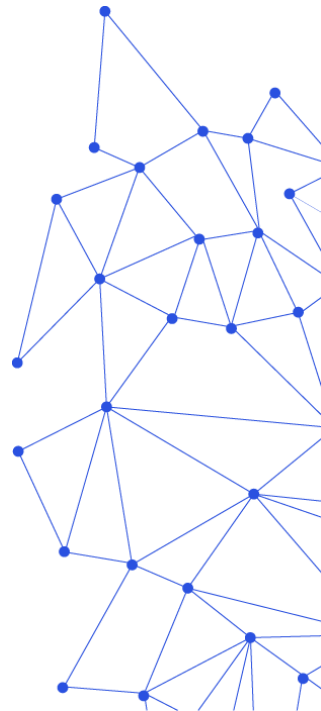
Integrated Formal Methods for Constructing Assurance Cases

Carmen Cârlan (carlan@fortiss.org)

Tewodros A. Beyene (fortiss GmbH, TU München)

Harald Ruess (fortiss GmbH, TU München)

fortiss GmbH
An-Institut Technische Universität München



Standardized verification methods

Testing, static analysis

- **Software testing** is used to demonstrate that the software satisfies its requirements and to **demonstrate with a high degree of confidence that errors** that could lead to unacceptable failure conditions, as determined by the system safety assessment process, **have been removed**. (DO-178C)
- 6.4.4. Test coverage analysis (DO-178C)
- An **analysis** may examine in detail the functionality, performance, traceability, and safety implications of a software component, and its relationship to other components within the system or equipment.
 - **Conformance of the source code to coding standards**
 - **Accuracy and consistency of the source code**

(DO-333)

State-of-the-art innovative verification workflows

Verification workflows

- ***Coverage Closure: The integration of formal-methods based tools [FShell tool] **with** industrial **testing** software [RapiCover] [...] in order to generate extra test cases and **increase code coverage results.** (nellis 2015)***

State-of-the-art innovative verification workflows

How to use their results for certification?

- *Formal analysis can be used to satisfy many of the verification objectives, completely in some cases and only partly in others. In this last case, the **verification plan should describe how the combination of formal analysis and other methods satisfies the objective completely.***

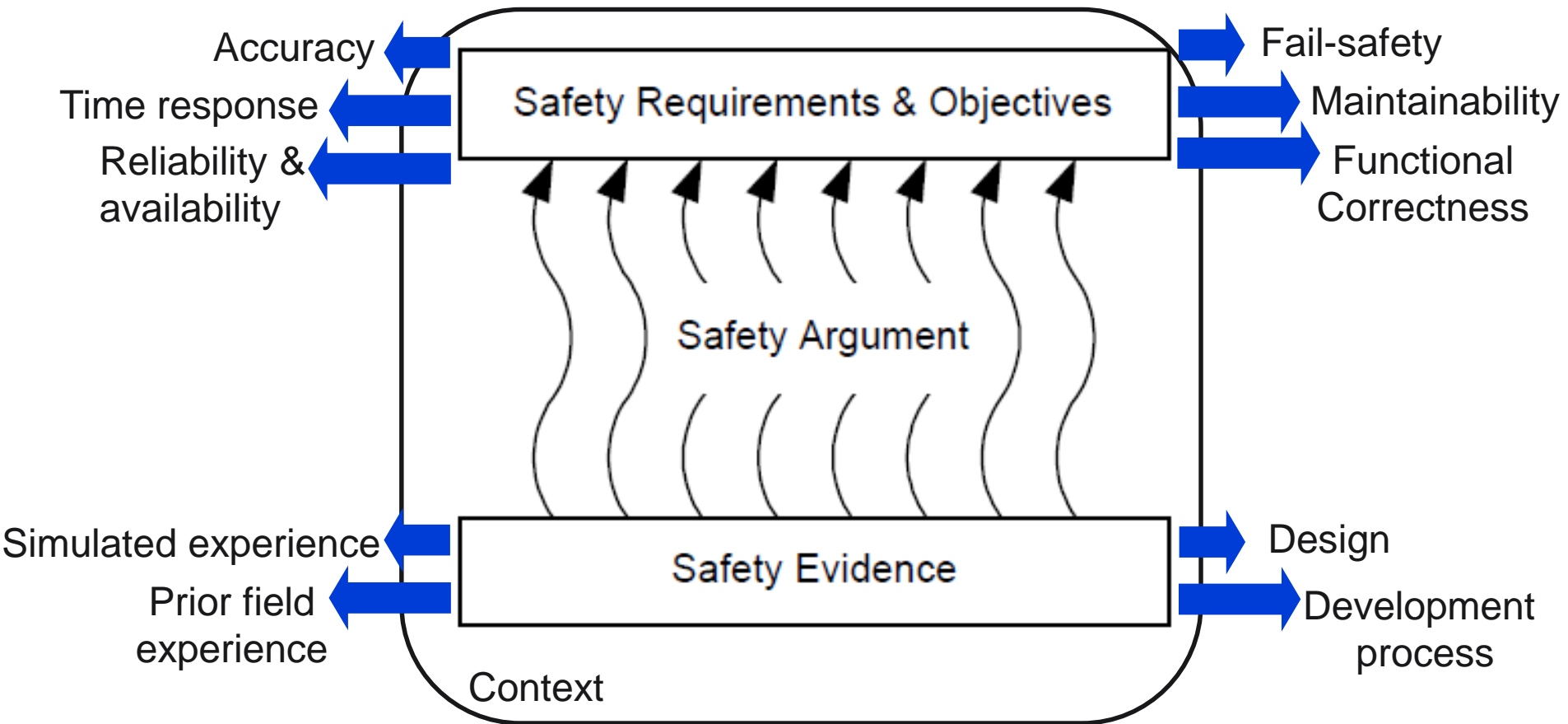
(DO-333)

- *One technique for presenting the rationale for using an **alternative method** is an **assurance case**, in which arguments are explicitly given to link the evidence to the claims of compliance with the system safety objectives.*

(DO-178C)

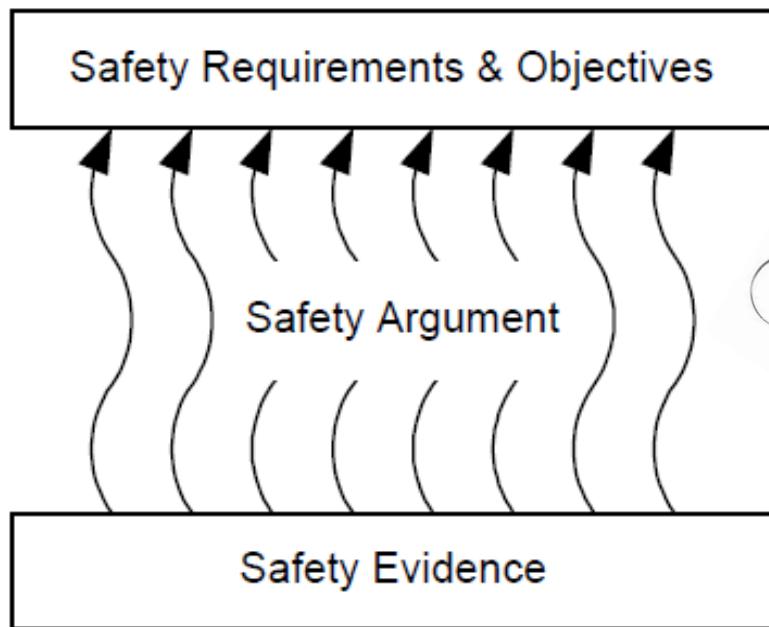
Assurance cases

Arguing the assurance of a system



Goal Structuring Notation (GSN)

Depicting assurance cases



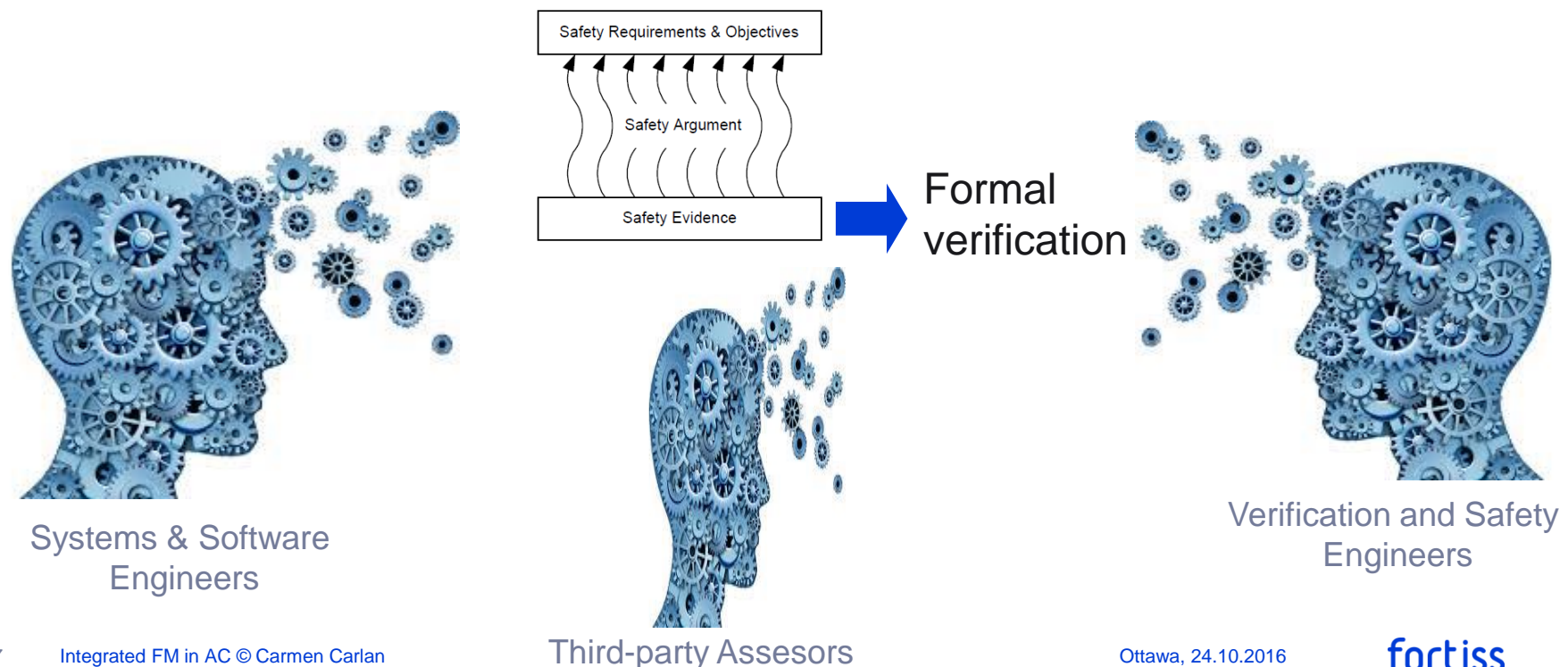
Goals which are in fact claims about a property of the system or some subsystem, are broken down in sub-claims and, at the end of the safety argumentation structure, supported by evidence (**solutions**)

The safety argumentation structure also points out the **context** of the safety claims, the **strategies** used in order to satisfy certain system safety claims and the rationale behind the argumentations, by documenting **justifications** of a certain statement or by stating certain **assumptions** about the system's properties.

Bringing Formal Methods Closer to Daily Development

Our goal (ASSURE 2016, WoSoCer 2016, FVPE 2016)

Develop an assurance case pattern which can be used at the interface between practicing engineers (developers, verification engineers, safety managers and third party assessors), enabling them to adopt heterogeneous, but complementary (formal) verification methods in daily development



Innovative verification workflows

Using results for certification

- The use of (formal) verification methods in verification activities is well established in various dedicated safety standards and the certification objectives they must fulfill are defined.
- Defects in the verification process (e.g., incomplete coverage) may lead to assurance deficits (= certification objectives are not satisfied).
- Safety standards promote the use of integrated formal methods when a single method cannot achieve the verification objective **without specifying how**.
- **How to use outputs from integrated formal methods as evidence in assurance cases, which are used in certification of safety-critical systems?**

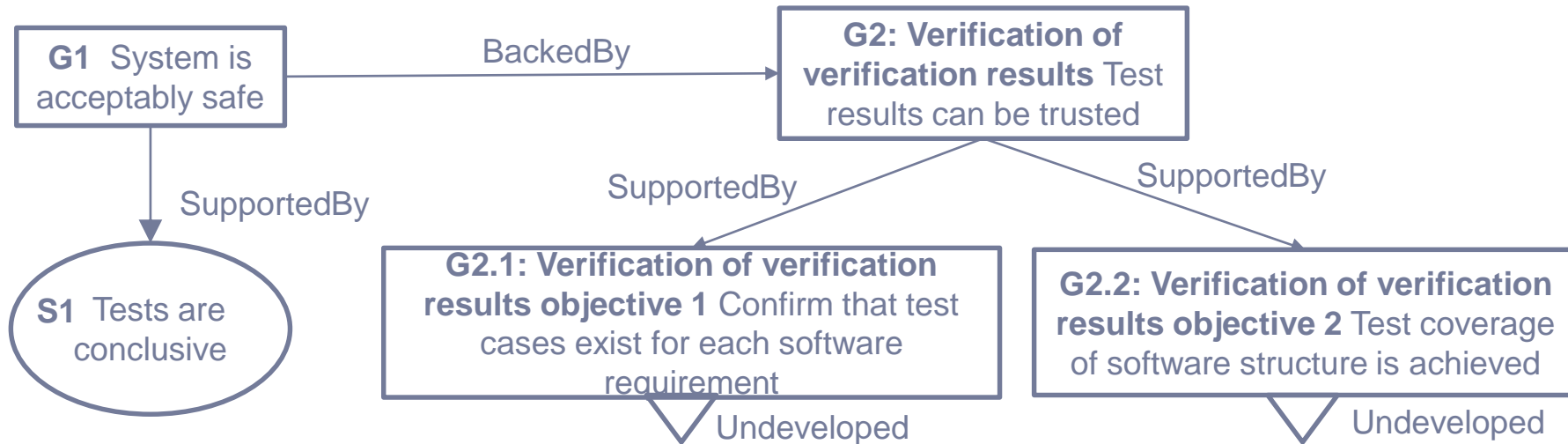
Standardized verification methods complemented by state-of-the-art innovative verification methods

Defect-based assurance

- Identifying the certification process objectives of a standardized verification method, based on the *Verification of verification results* process from DO-178C/DO-333
- Mapping the identified certification objectives to potential **process**/product **defects**, which may prevent the satisfaction of the objectives
- Identifying or building a verification workflow, integrating formal methods, which eliminates (some of) the identified potential **process**/product **defects**
- Constructing explicit assurance case patterns for the software verification workflow

State-of-the-practice (1)

Verification by testing

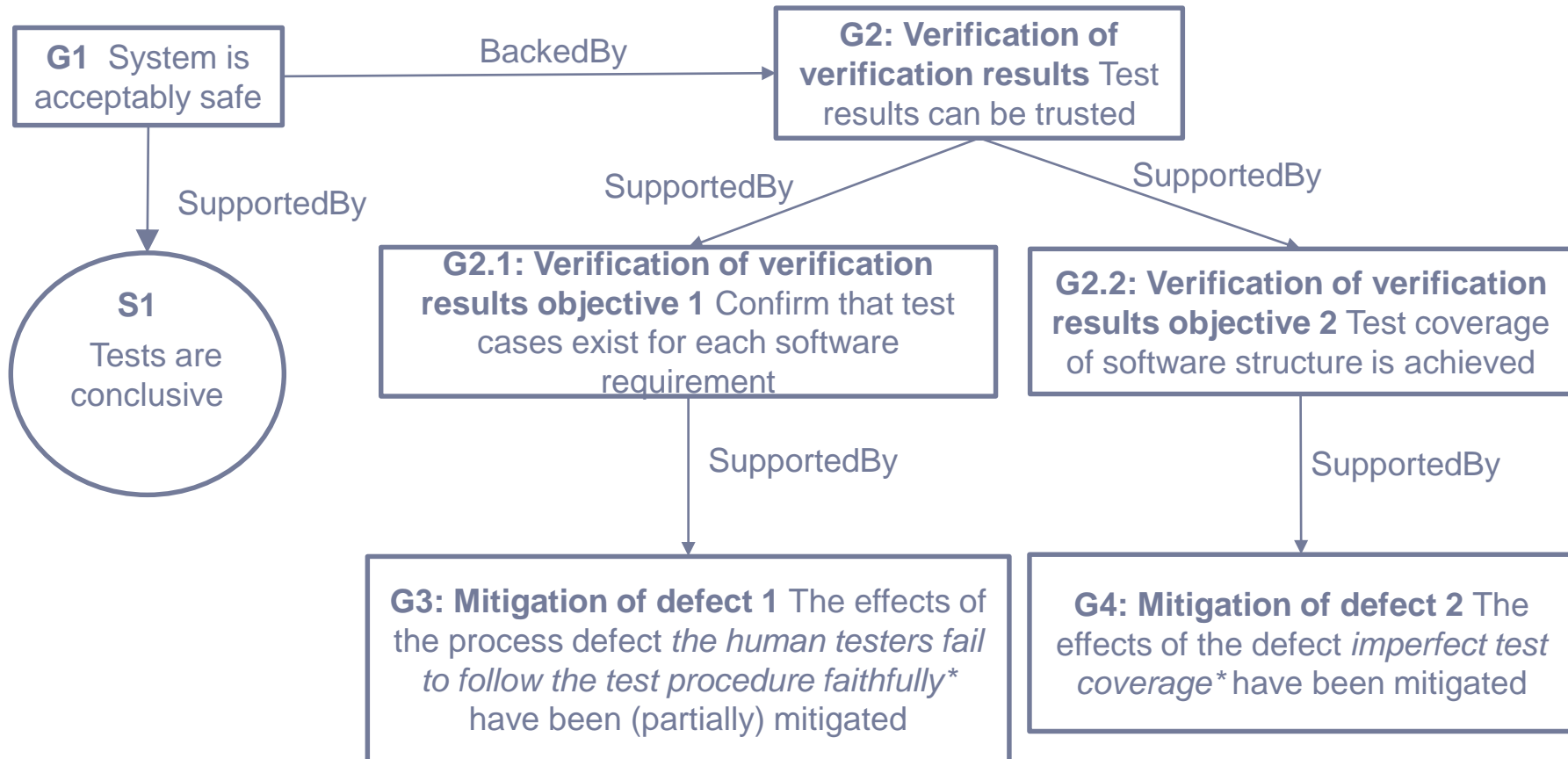


Conformance of the implemented behavior with requirements

Analysis of the coverage of the implemented behavior

Potential process defects

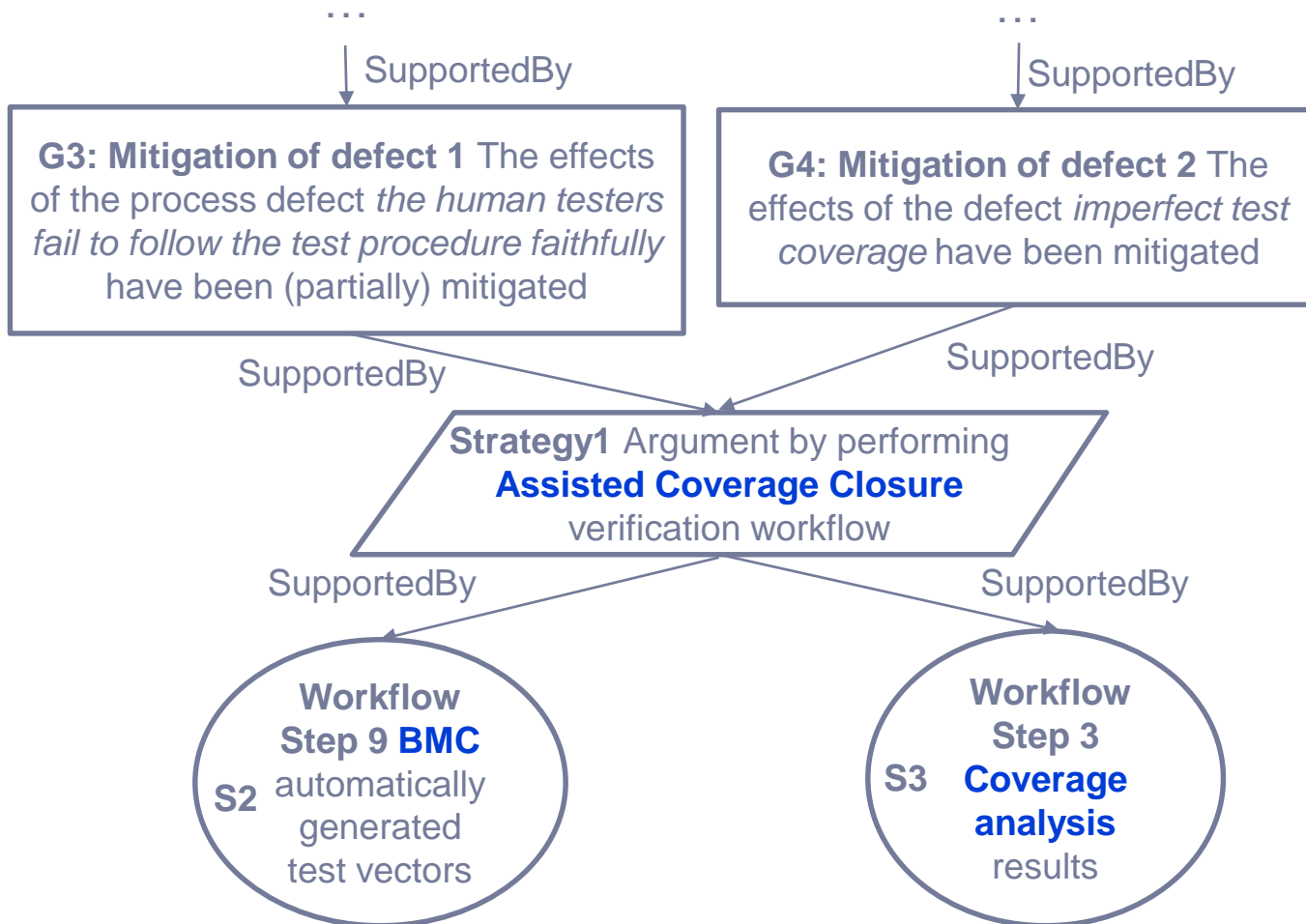
Verification by testing



* *A new approach to creating clear safety arguments* (hawkins 2011)

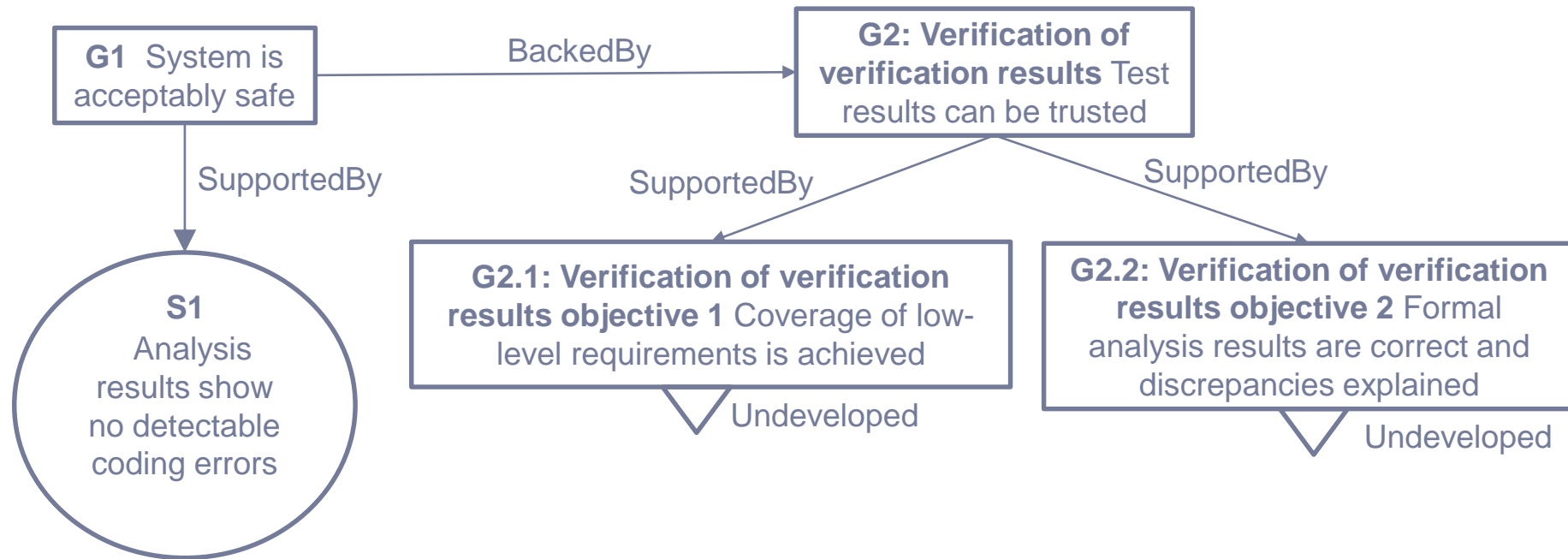
Arguing the mitigation of potential defects using the rationale behind innovative methods

Heterogeneous evidence output by the code coverage workflow



State-of-the-practice (2)

Verification by static analysis

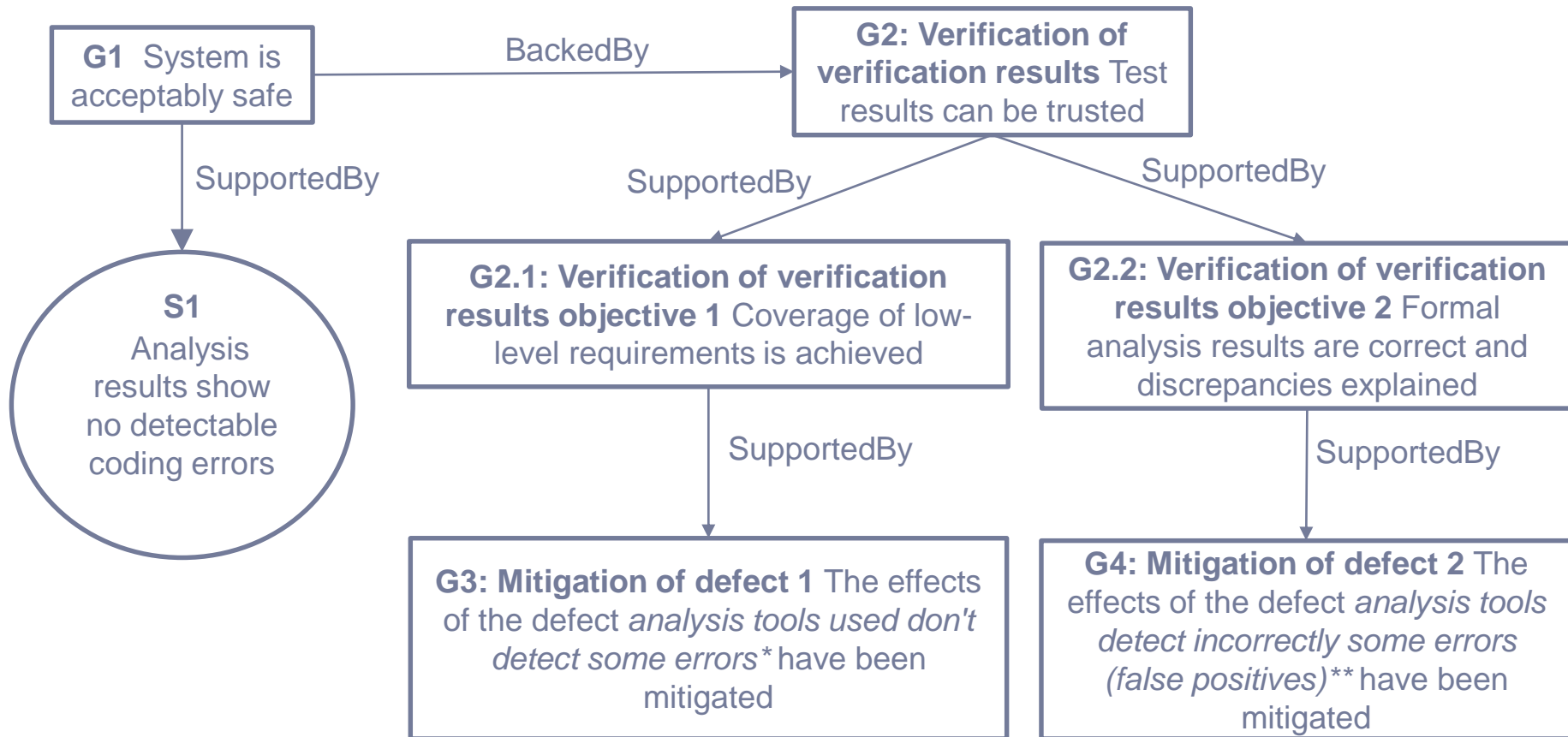


Conformance of the implemented behavior with requirements

Analysis of the coverage of the implemented behavior and of correctness

Potential process defects

Verification by static analysis

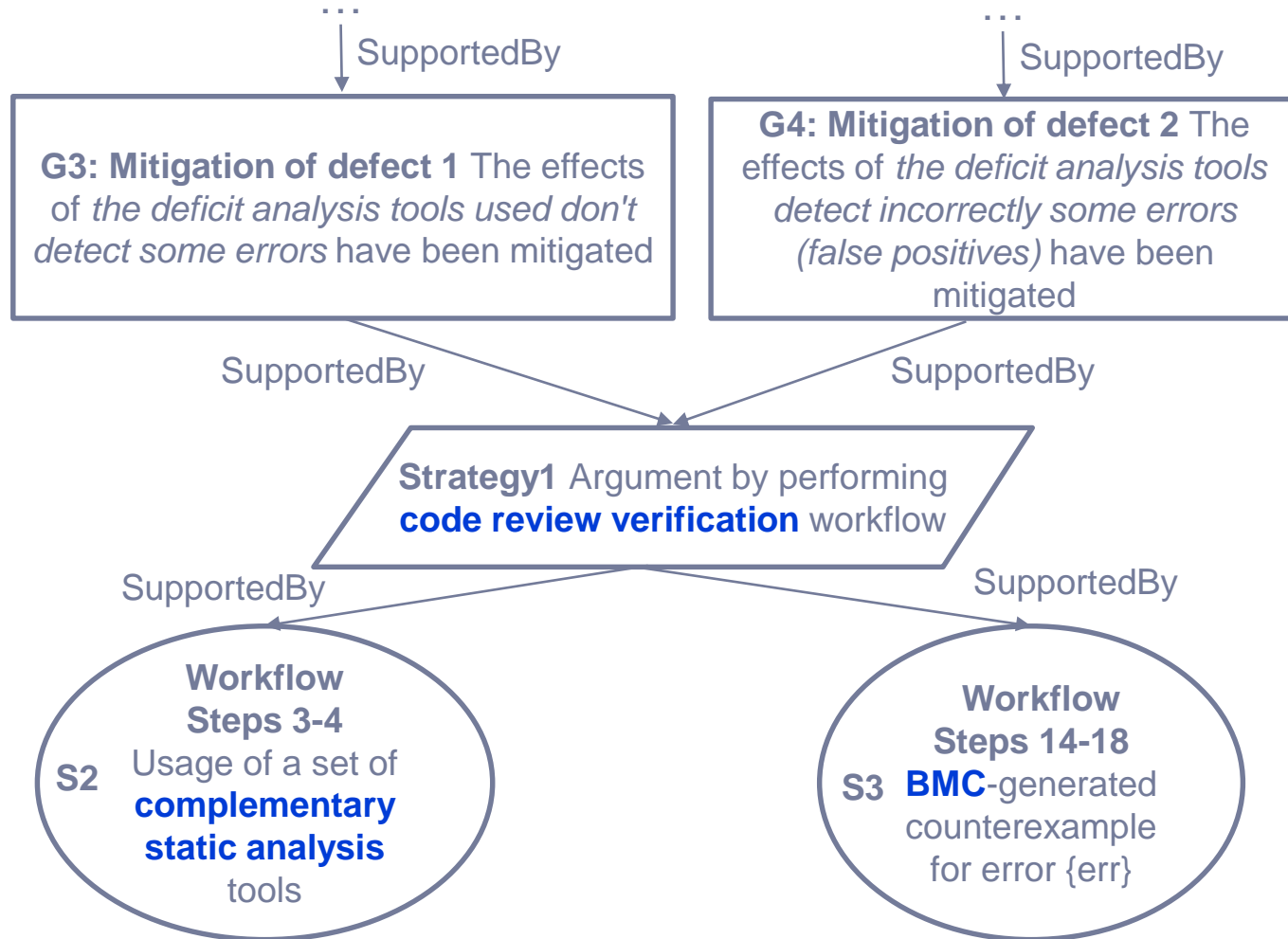


**Eliminative argumentation ... (goodenough 2015)*

*** A few billion lines of code later ... (bessey 2010)*

Mitigation of potential defects using the rationale behind innovative methods

The code review workflow



Open Questions

Integrated Software and Assurance Engineering

- How can results of the state-of-the-art verification methods/workflows be used as assurance evidence?
 - How to argue the suitability of the usage of a certain verification method for the satisfaction of a certain certification objective?
 - How to document the rationale behind satisfying certification objectives related to verification process?
 - What other heterogeneous backing evidence from other verification methods could compensate for deficits of a traditional verification methods?
- What is the extent to which all assurance comes down to showing the elimination/mitigation of hazards/faults as opposed to the presence of positive properties?
- What is the relationship between assurance cases and defect models?
 - How defects in verification process may contribute to missing product defects?
 - How would a mapping of certification objectives to product defects would look like?
 - How can results from defect-based verification be integrated in assurance cases?
 - How can a defect model be instantiated as an assurance case?

Wrap-up

Integrated Software and Assurance Engineering

- Today
 - Discharge assurance deficits with results from heterogeneous methods
 - Explore the complementary nature of verification methods regarding assurance argumentation
 - Documenting the rationale of the compliance with certification objectives by a structural correspondence of the workflow description and the generated assurance case
- Tomorrow
 - Instantiation and validation of the code review workflow
 - Better integration of formal methods in the development and verification of safety-critical systems
 - Defect-based assurance