SIEMENS

Corporate Technology

Approaches to meet Certification Requirements for Mission-Critical Domains

By S.Vinogradov, V.Okulevich, M.Gitsels

Presented by Sergey Vinogradov

Software Engineering Conference (Russia), 16th Nov. 2006

Content

- Certification in Mission-Critical Domains
  Elements, Requirements
- Compliance with safety standards
- V&V techniques used in OOO Siemens CT SE
  Static Analysis, Code Coverage Analysis: Examples
- Our experience makes the case
  Evidence Improvement for Certification
- Conclusions
  Our vision on further perspectives

Certification: Elements of Software Certification

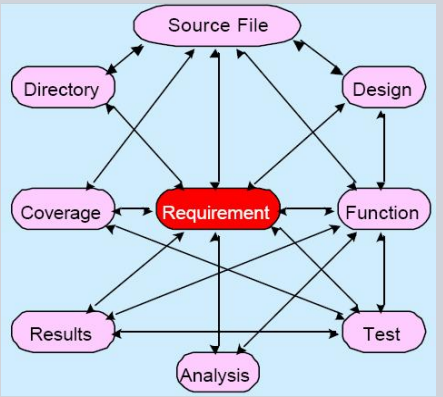
Elements of Software Certification

- Product
  - Evaluation, measurement of the product
  - Uncertainty reduced by reference, measurement methods
- Process
  - Standards to define what must be performed to build software systems
  - Assessment of organizations for conformance
- People
  - Skilled people to get processes right
  - People with knowledge beyond computer science/swe
  - Licensing “software engineers” by state examinations

SIEMENS

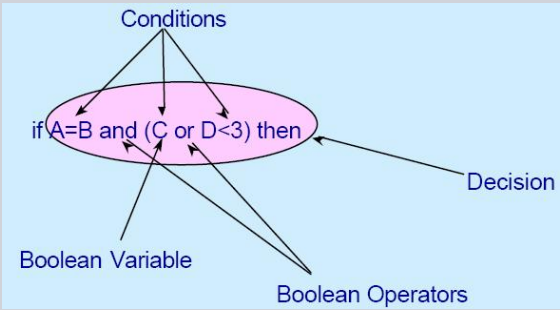
Certification Requirements: Examples

At development process level



- Focus on Requirements, Test, Traceability
- Use of Tools to manage process
  - Requirements captured electronically
  - Traceability information added (or conjured by system)
  - Checklists, documents, test templates generated automatically
- Application of Design and Coding Policy
  - Reviews

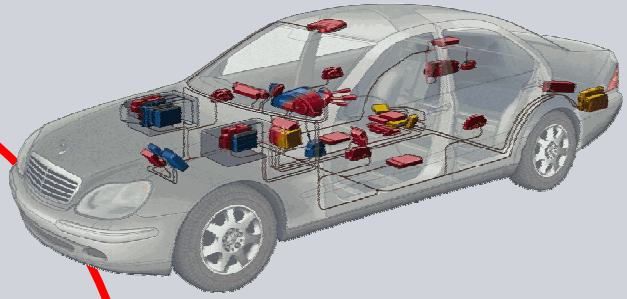
At software product level



- Definition of components integrity levels
- Traditional V&V intensified with a combination of diverse development techniques
- Application of advanced assurance techniques

SIEMENS

Dependability in Siemens products



Integrity

Safety

Maintainability

Dependability in Siemens Products

Availability

Reliability

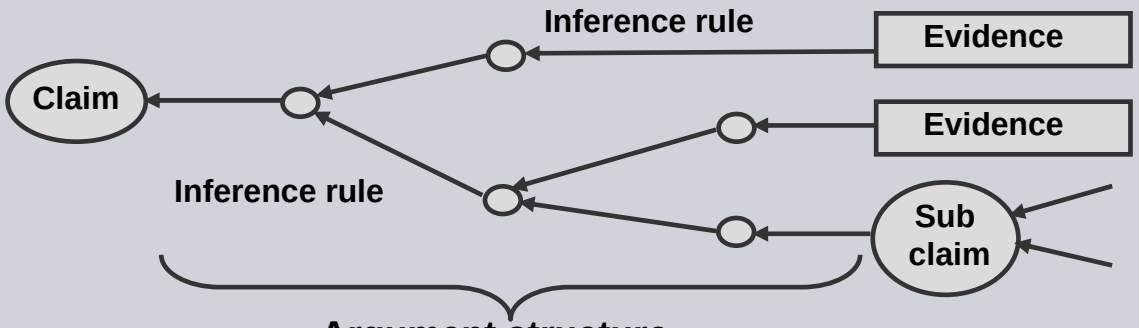
Confidentiality

Almost all Siemens Business Units develop and sell products that perform safety-critical operations. Most of these products nowadays contain an ever increasing software part.

SIEMENS

Evidence Improvement for Certification

Claims Proving Structure from Safety Cases (UK) ADELARD

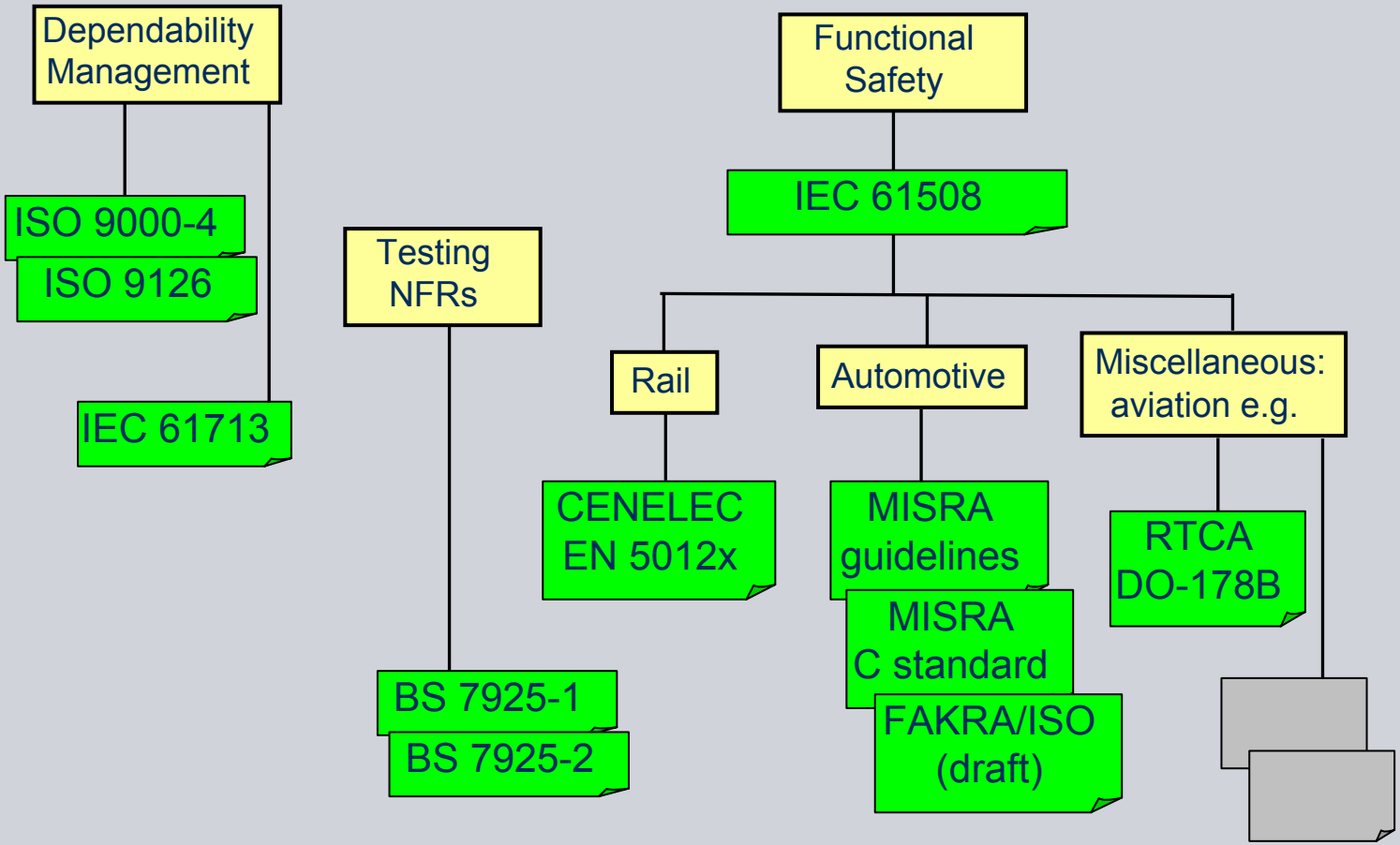


Claims	Argumentation	Product & Process Evidence
- functional properties - non-functional - safety - reliability - availability - maintainability - security	- static code analysis - formal proofs - MTTF, reliability testing - compliance to safety standards - coverage analysis	- design documentation - testing evidence - COTS product evaluations - …

Compliance to safety standards

SIEMENS

Compliance to Safety Standards and Guidelines



SIEMENS

Compliance to Safety Standards and Guidelines

IEC 61508, DO-178B

- Sound basis for certification
- One of the ways to gain software ‘approval for use’
- Product and Process Examination
  - Six processes to be performed (Planning, Development, QA, etc.)
  - Traceability between requirements, design, code, tests, etc.
  - Independence of some activities performed (IV&V)
  - Product evaluation tasks determined by integrity level
- Integrity levels
  - Task selection, degree of task performance
- Uses an overall safety lifecycle model as the technical framework for the activities necessary for ensuring safety

We mean to use standards recommendations for not safety-related software

SIEMENS

Corporate Technology

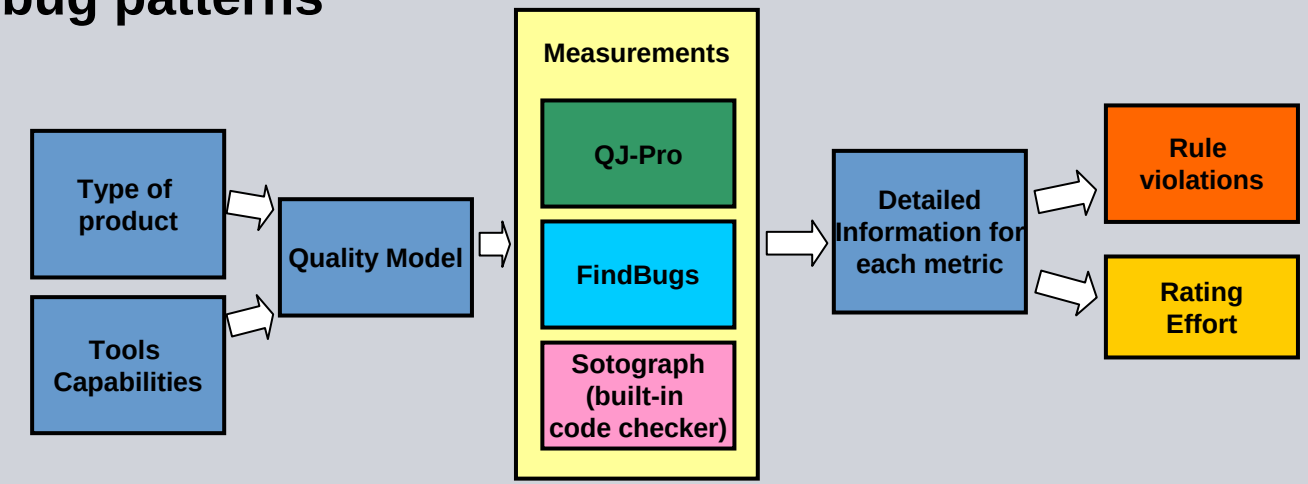
Application of Static Code Analysis

SIEMENS

Static Code Analysis (tool-based)

- **Subjects** program texts to scrutiny and review in order to detect inconsistencies and omissions
- **Checks**
  - conformance to coding standards
  - misuse of programming language
  - best practices (e.g. defensive programming)
  - code structure
- **Identifies bug patterns**

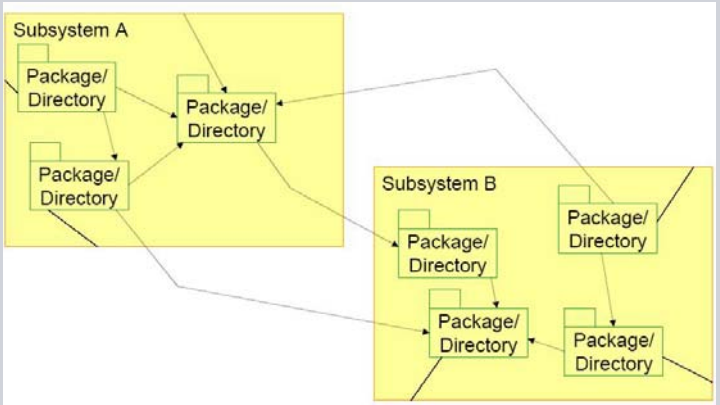
- How could we improve/ supplement it?
- Add software structure assessment →



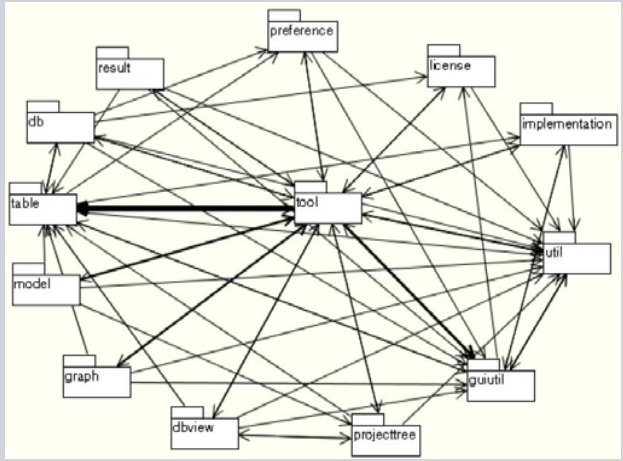
SIEMENS

Software Structure Assessment

- **Goals**
  - Identify architecture violations on the source code level
  - Evaluate maintainability and understandability of software under analysis
  - Supplement traditional static analysis



- **Means**
  - Aggregation of fine-grained source code artifacts to subsystems
  - Definition and check of rules how subsystems are allowed to interact (call-references, attribute usage, inheritance etc.)
  - Cycles based analysis
    - How many cycles?
    - How large is a cycle?
    - Which are high cyclically coupled artifacts?
  - Graphs for visualization



SIEMENS

Static Analysis: Sample project results

Static Analysis

- By applying code analyzers with various capabilities there were identified:
  - Code structures with potential side-effects
    - missing ‘default’ statements,
    - usage of deprecated classes and methods,
    - methods fail to close streams etc.
  - Maintainability issues
    - high complexity expressions, dummy comments, confusing class/method names

Cycle-based analysis

- On package level there were found 3 cycle groups
  - One of subsystems is considered not maintainable in case of further evolutions due to chaotic relationships between members

SIEMENS

Lessons learned

- **High level view based on Quality Model**
- **Detailed rating information**
  - Risk assessment (based on expert evaluation)
  - Effort (to fix findings)
- **Identified places in code with potential side-effects**
  - Inappropriate objects comparison
  - Unhandled states of the system (missing ‘default’ statements)
  - Software misbehavior

- **Conclusions**
  - Might include elements of code review technique (safety standards requirement)
  - Difficult to implement large sized software system without considerably breaking the planned architecture
  - Necessity of tool based regular architecture conformance checking

SIEMENS

Corporate Technology

Application of Code Coverage Analysis

SIEMENS

Coverage Analysis – Rigorous V&V Technique

It is the process of

- Finding areas of a program not exercised by a set of test cases
- Creating additional test cases to increase coverage
- Determining a quantitative measure of code coverage, which is an indirect measure of quality

Condition/Decision coverage

- All decisions must be executed
- All decisions with possible outcomes
- All conditions with all possible outcomes

If A=0 and B<2 and C>5 then P; …

A=0	B<2	C>5	P
T	T	T	T
F	T	T	F
T	F	T	F
T	T	F	F

Each outcome must be tested once

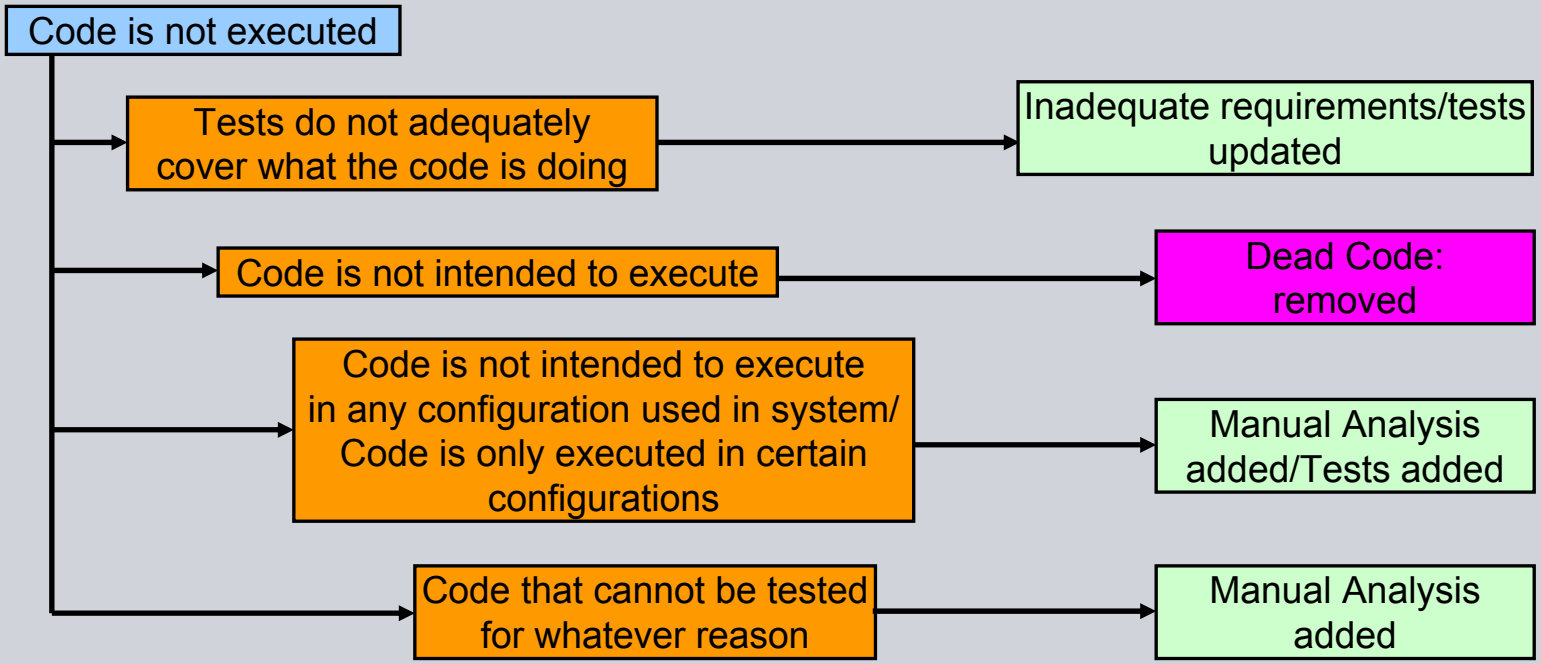
As a result

- We are able to show that each condition has its intended effect on the outcome of a decision

SIEMENS

Coverage Analysis – Targets

- Mission-critical software has the highest target – 100% coverage, revealed not exercised code structure is processed in the following way:



- The lowest targets assume having another testing strategy
  - E.g. attain some coverage through the entire test program
  - Strive for high coverage in any particular area.

SIEMENS

MCDC Coverage Analysis – Example

Report before analysis:

```
decision line 57: if statement
        if( (channel>=0) && (channel<NUM_ADC_CHAN) )
        T       F
    1: *tt      fx   ((channel>=0)) &&    PARTIALLY covered
```

Report after analysis:

```
decision line 57: if statement
        if( (channel>=0) && (channel<NUM_ADC_CHAN) )
        T       F
    1: *tt      fx   ((channel>=0)) &&    PARTIALLY covered
```

#ANALYSIS:

- **Functions invoked with project defined constant parameters**
- **Many functions are called and configured with sets of constant data that have been defined for the project**
- **These configuration parameters prevent many functions from covering all paths**
- **These paths are not traversed at any time and can be shown that they will never be traversed within context of this project.**

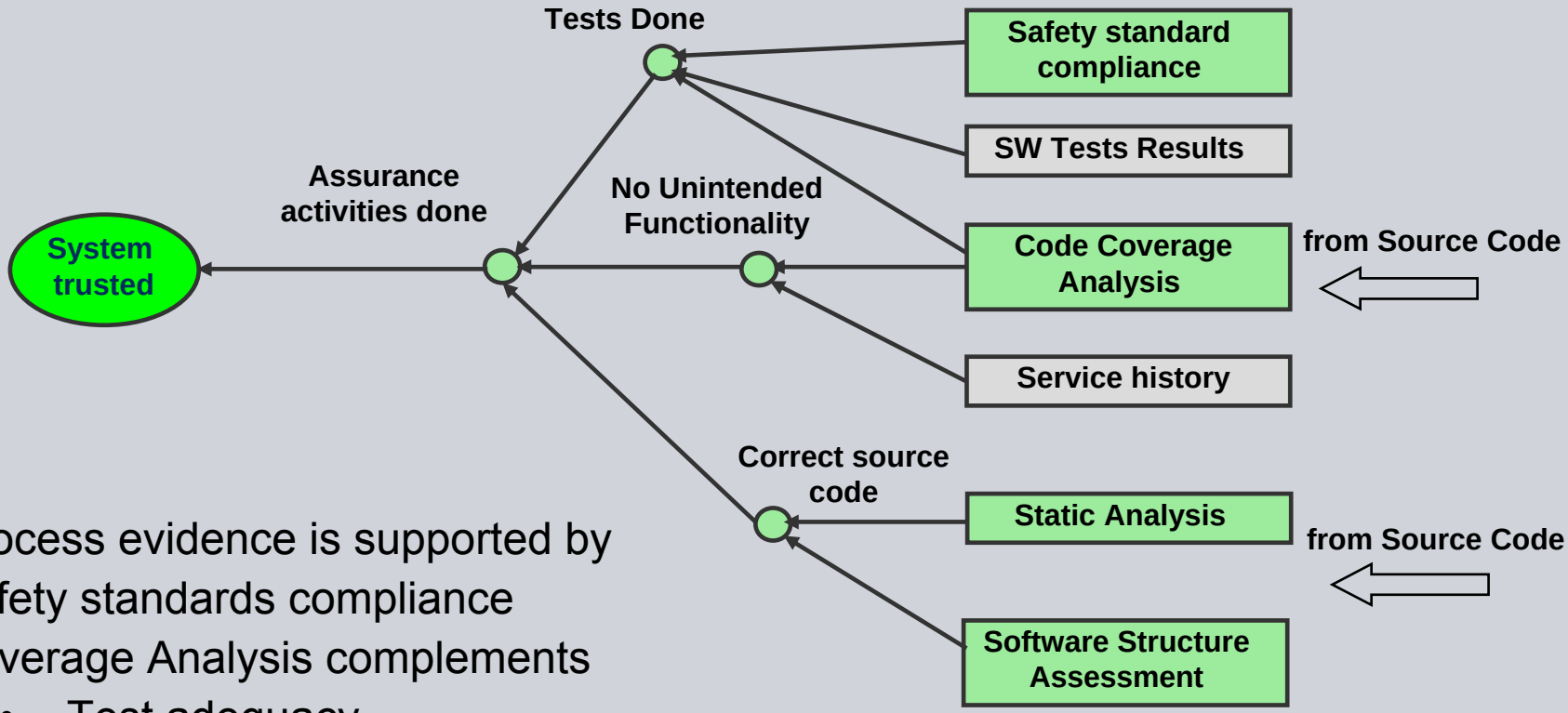
SIEMENS

Lessons learned

- Coverage analysis (MC/DC) is the recommended mean to support certification materials
- Coverage analysis is a rigorous testing technique that helps:
  - Eliminate gaps in a test suite
  - Most in the absence of a detailed, up-to-date requirements specification
- Multiple condition/decision coverage is very useful general-purpose measure for C, C++, and Java
- Setting right target of coverage can increase testing productivity
- Programmers have to think more about testability of their code

SIEMENS

Our experience makes the case …



- Process evidence is supported by safety standards compliance
- Coverage Analysis complements
  - Test adequacy
  - Service history
- Software Structure Assessment provides additional confidence in source code thoroughness

We continue removing remaining uncertainty required for certification…

SIEMENS

Conclusions

- Assurance of mission-critical software quality required by **certification** cannot be provided by only one technique

- Still important to have trusted engineering processes to produce software artifacts which:
  - can be measured
  - analyzed during the lifecycle

- Application of V&V techniques complementing each other like
  - static code analysis supported by software structure assessment
  - coverage analysis assuring quality of set of tests

- Outlook
  - Choice of methods has to depend on the required level of dependability inline with the target domain
  - We are looking forward for application of promising techniques such as model checking, advanced static code analysis, in-depth testing for software safety

THANK YOU!

QUESTIONS?