Evaluating hazards in critical software-dependent systems

Sushil Birla
U.S. Nuclear Regulatory Commission
ASQ Section 509, December 15, 2015

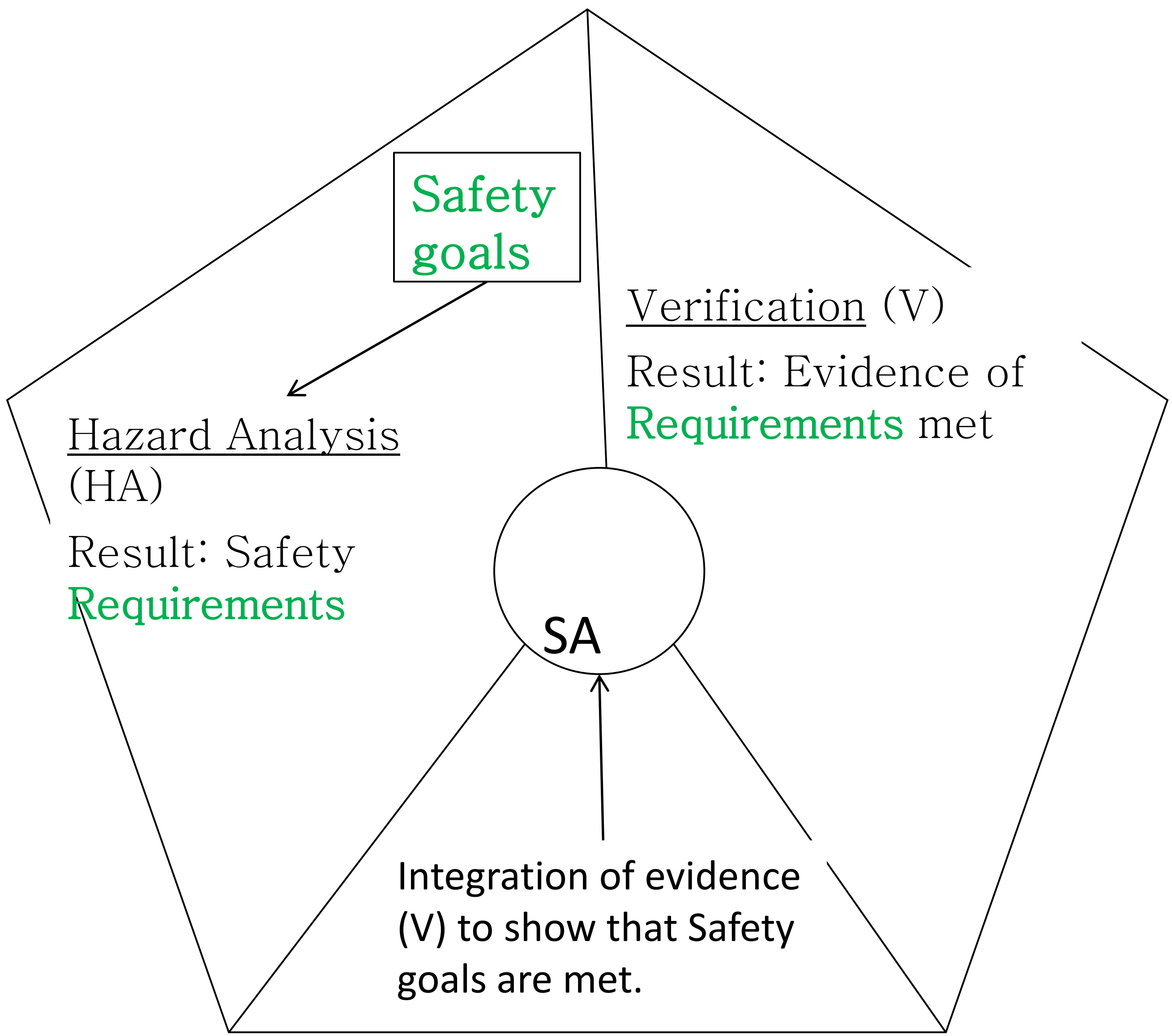
Outline

- The problem in focus
- Software quality – knowledge evolution
- Quality models – framework
- Streamlined quality model to evaluate safety
- Derivation of constraints
- Related R&D activities
- Reasoning structure
- Future directions

Focus

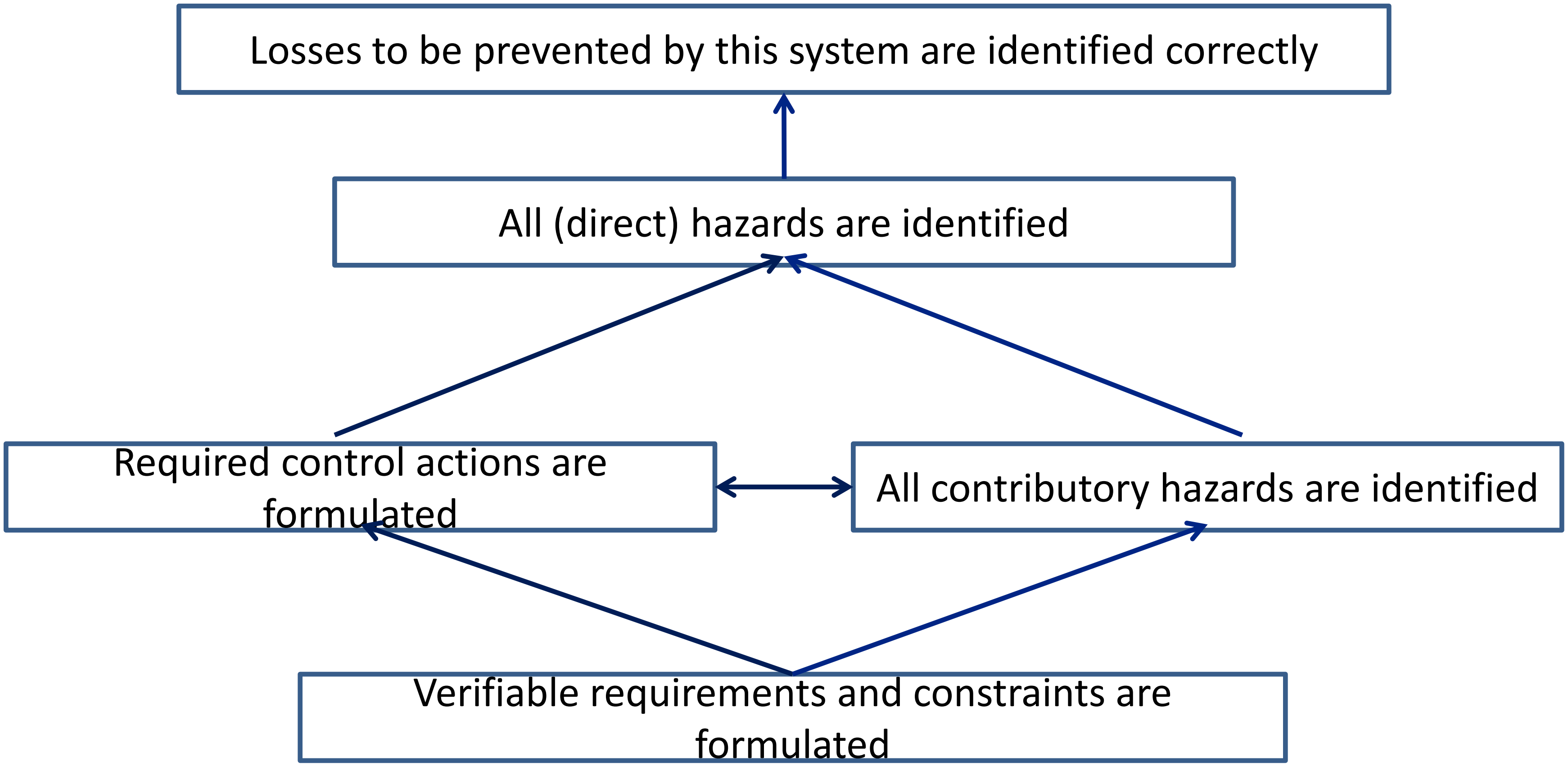
- Hazards in a critical cyber physical system
  - contributed through systemic causes, e.g., engineering deficiencies.
- Integrated approach to:
  - systems engineering
  - software engineering
  - safety engineering

Hazard analysis in relation to Safety Analysis (SA)



U.S.NRC
UNITED STATES NUCLEAR REGULATORY COMMISSION
Protecting People and the Environment

Identify safety goals, requirements & constraints through Hazard Analysis



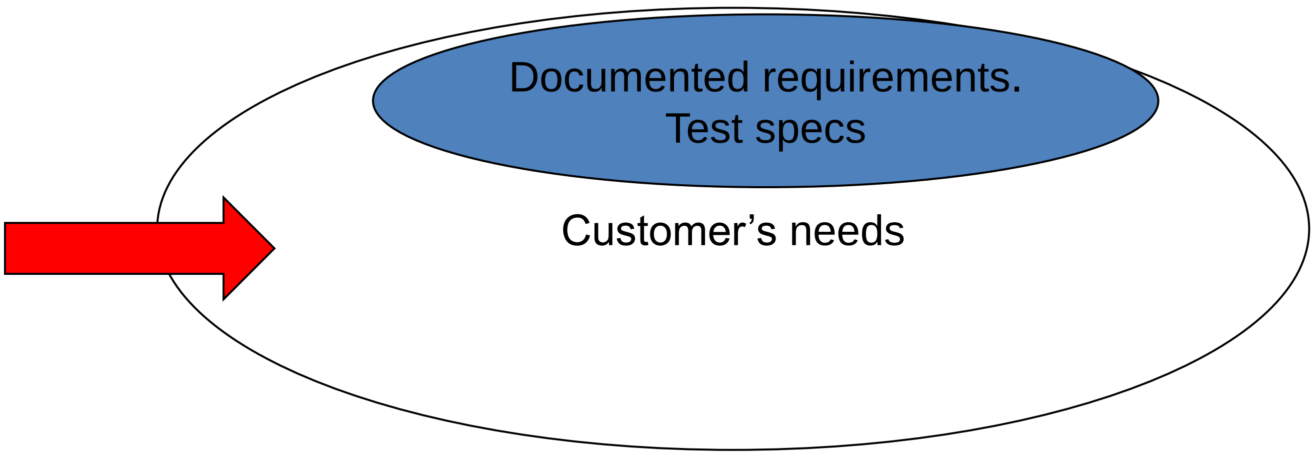
Root of mishaps: Requirements engineering

Mishaps in software-dependent critical systems

even after

- The system was verified to provide all the needed functions
- All constituents of the system met their respective requirement specifications
- None, by itself, “failed.”

Gaps



A weak link in requirements engineering

V&V

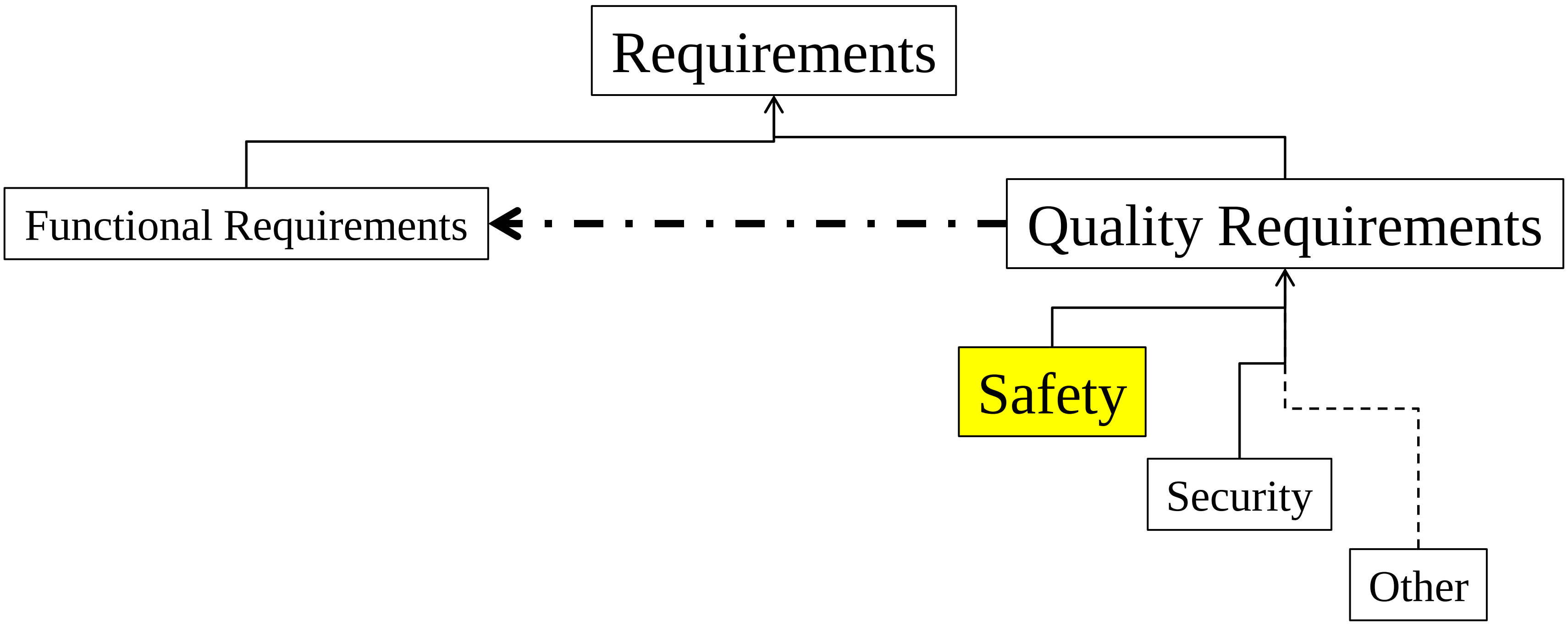
Verification
"Have I met the requirements?"

100% Verification will never identify inadequate requirements

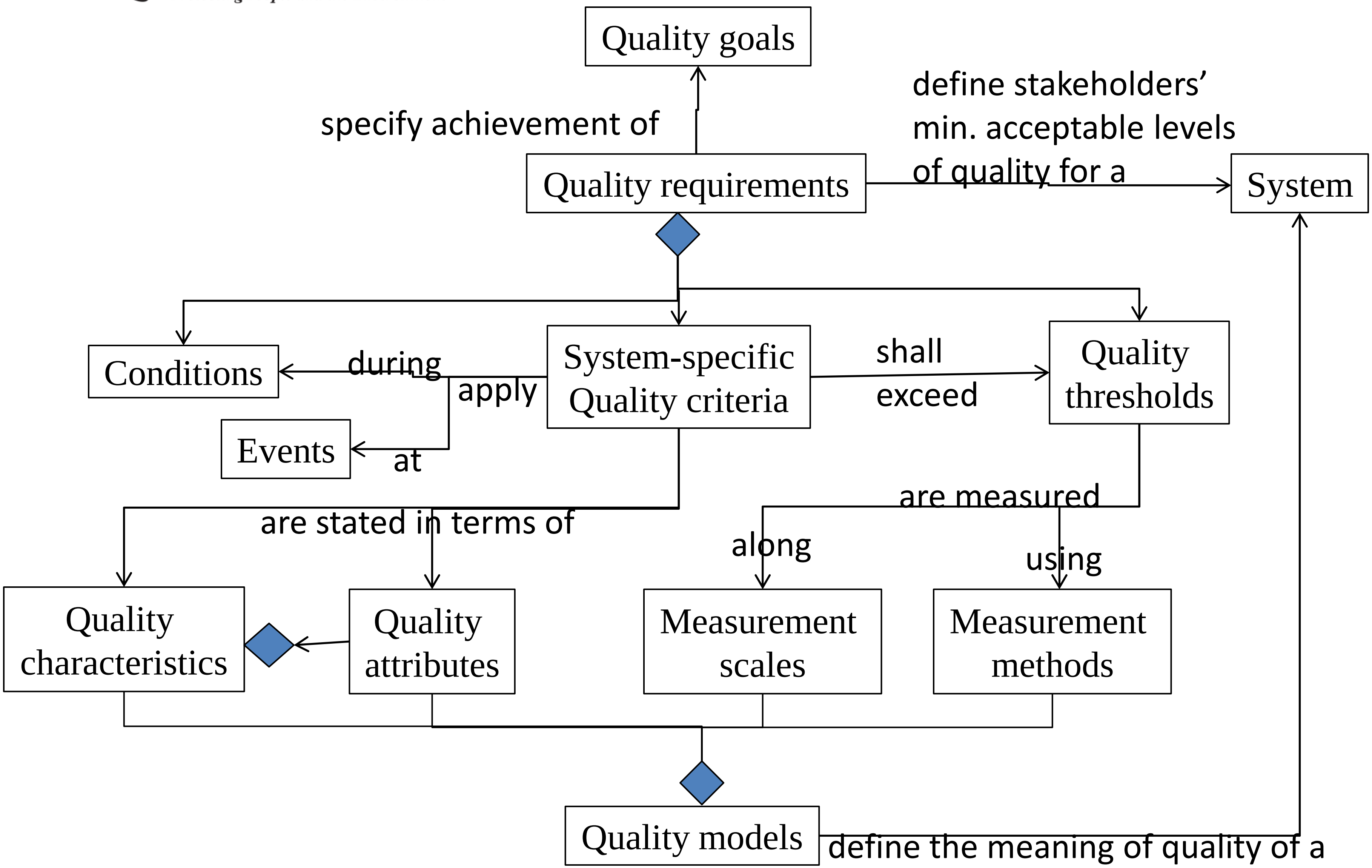
Validation
"Do I have the right requirements?"

Weakest Link is establishing valid requirements to drive system design

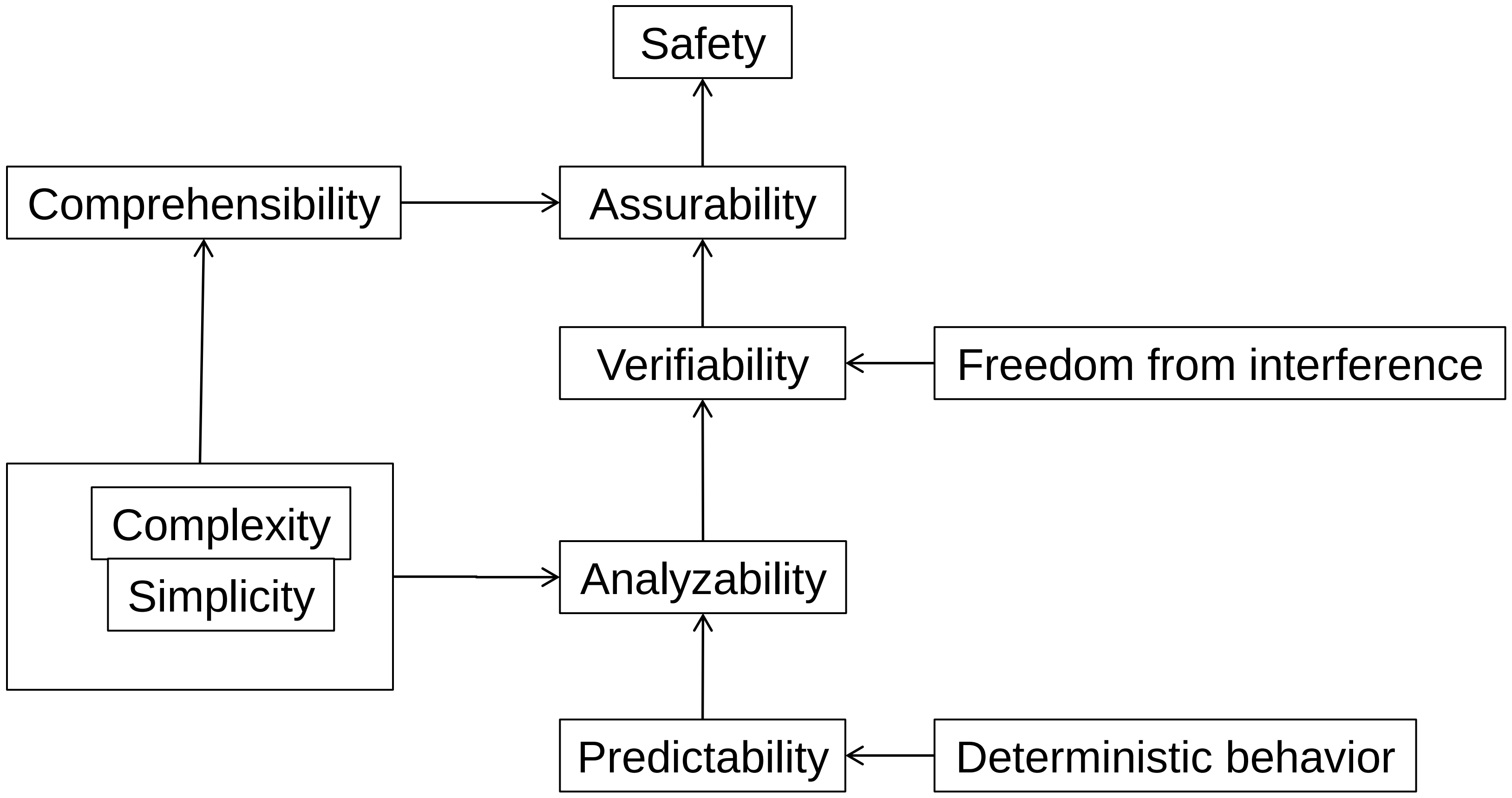
Aspect of the gap in focus



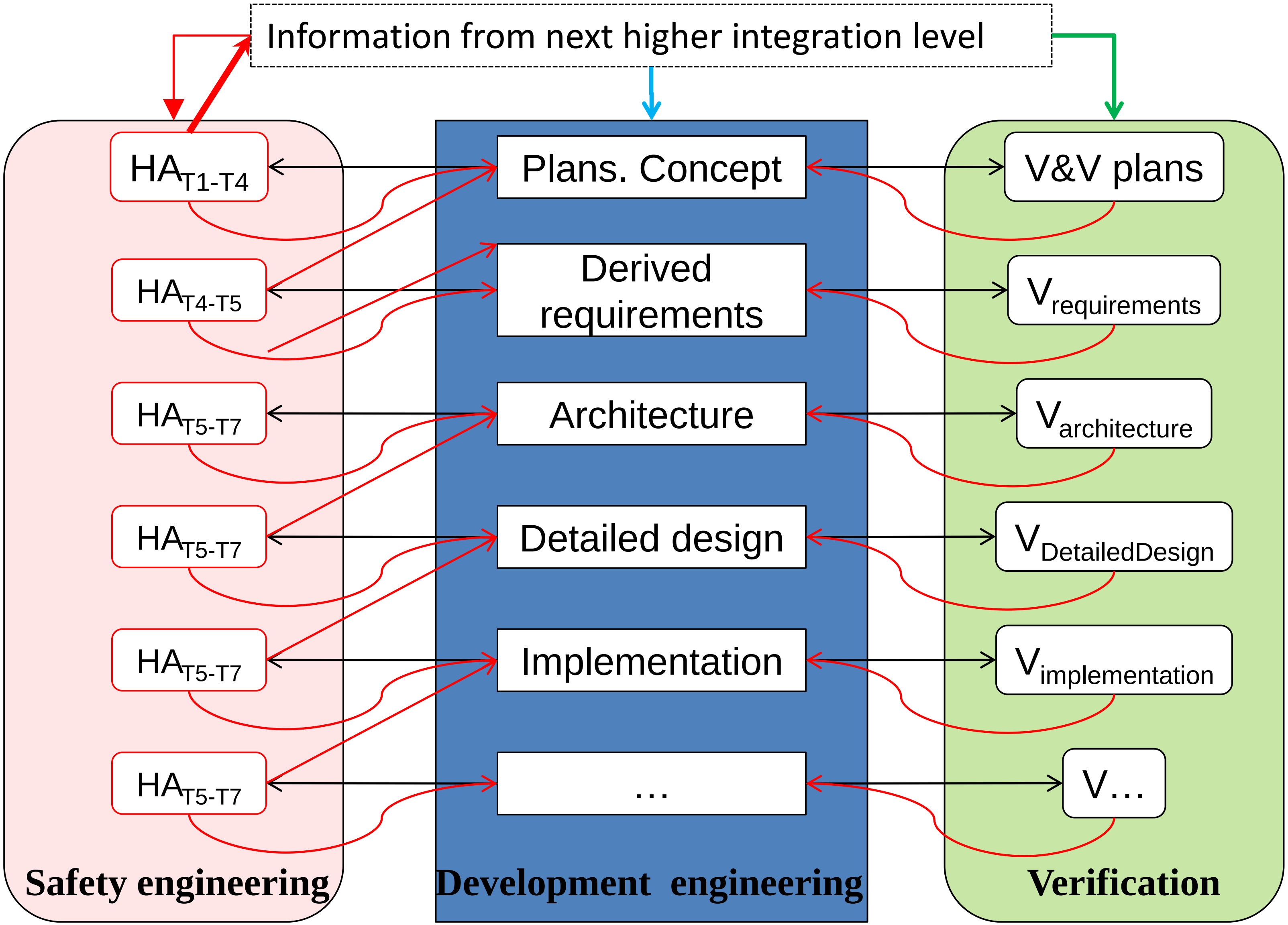
Quality models: framework



Streamlined Quality model to evaluate safety



HA: Source of validated safety requirements



Current State & Trends

Trends

- Interconnections ↑
- Feedback paths↑

→ Complexity ↑

- Comprehensibility ↓
- Verifiability ↓
- Analyzability ↓
- Deterministic behavior ↓

Side effects

Unwanted interactions↑

→

- Hidden dependencies ↑
- Independence ↓
- Common cause ↑
- Redundancy ↓
- Diversity ↓
- Defense in depth ↓
- Safety margins ↓

Consequence

Traditional HA techniques (FTA; DFMEA) ineffective
[RIL-1001; RIL-1002; NUREG/IA-0254; RIL-1101]

U.S.NRC
UNITED STATES NUCLEAR REGULATORY COMMISSION
Protecting People and the Environment

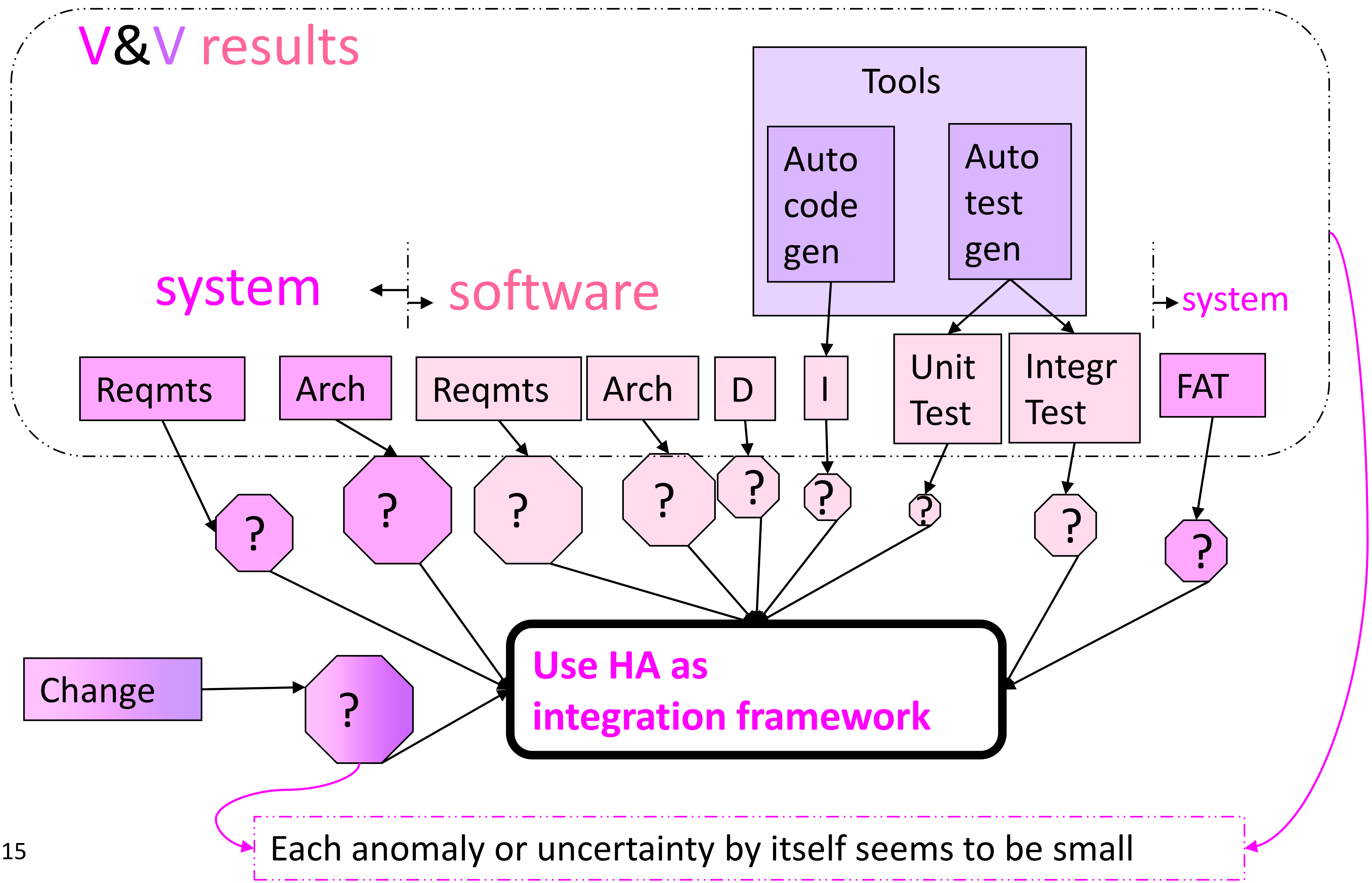
Software quality: knowledge evolution

- Since 1976: R&D on quality frameworks, models:
  - McCall; Boehm; U.S. Air Force
- 1979: F. Roberts – Measurement theory …
- 1978-1985: : ISO/IEC JTC1 efforts
  - Developed consensus for common standard
- 1994: QM-QA Vocab superseded by ISO 9000
- 2001: ISO/IEC 9126
  - State of the art inadequate.
  - Quality characteristics from ISO 84021.
    - Used broadly; Starts with the user's needs.
- 2005-2011: ISO/IEC 25000 series SQuaRE
- 2011: ISO/IEC/IEEE 29148 - RE

Some related R&D activities

- Boehm, USC:
  - Tradeoffs across competing quality attributes
- Software Engineering Institute
  - Quality attributes → Architecture
  - Integrated assurance & development environment
    - DARPA HACMS Rockwell Collins project
- NRC RIL-1101 http://cps-vo.org/node/8758.

Contribution of uncertainties



Architecture - definition

Structure or structures of the **system**, which comprise **elements**, the **externally visible properties** of those elements, and the **relationships** among them and with the **environment**

WHERE:

System: combination of interacting elements organized to achieve one or more stated purposes. Systems can comprise of systems. A system with only software elements is also a system.

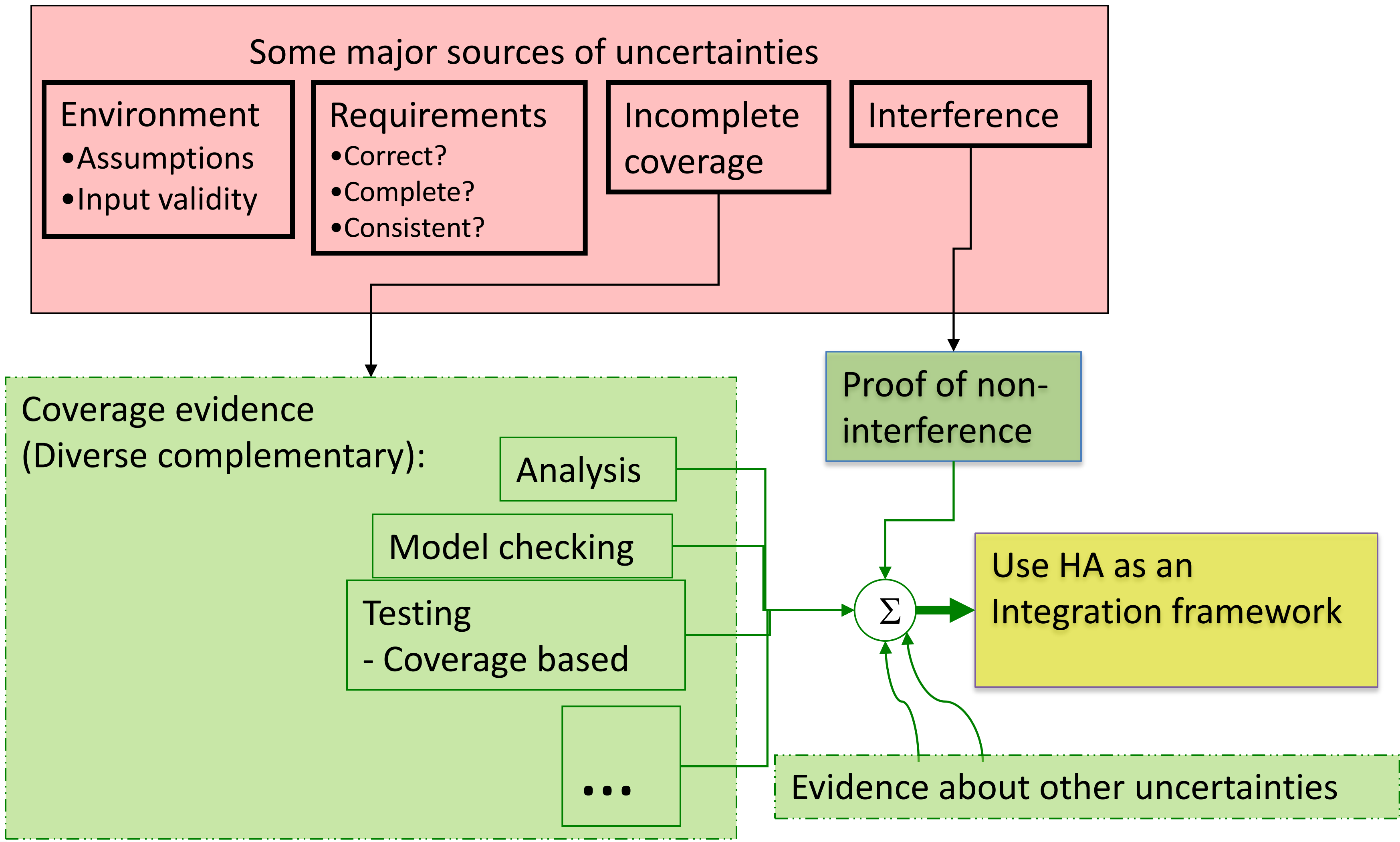
Environment: includes the combination of systems and elements external to this system, human elements interacting directly with the system and the commensurate manual procedures.

Element: a discrete part of a system that can be implemented to fulfill specified requirements. Examples: hardware, software, data (structure), human, process (e.g., process for providing service to users), procedure (e.g., operator instructions), facility, materials, or any combination.

Externally visible properties: include behavior – normal, as well as abnormal.

Relationships: include interactions and interconnections (communication paths).

Framework to integrate evidence



Hazard categories (direct; contributory): examples

Loss of concern due to:

- Unrecognized inter-dependencies in the system
- Unrecognized inter-dependencies in processes:
  - Technical
  - Supporting
  - Management
- Disruption in or emissions from the environment
- Emissions/outputs from or behavior of the system
- Unrecognized anomaly in the state of the process (reality ≠ perception)
- Ill-understood nature of change in monitored phenomenon
- Unrecognized anomaly in the state of the instrumentation
- Unrecognized anomaly in state of some element in environment
- Ill-defined boundary of the system
- Unrecognized hazards from interactions of the system with its environment
  - e.g., effects of invalid assumptions,

Verifiability : Hazards → Constraints

Hazard scenario contributed through non-verifiability:

Appropriate criteria are not formulated at beginning; therefore, corresponding architectural constraints are not formalized and checked. By the time work products are available for testing, it is discovered that adequate testing is not feasible.

Examples of derived constraints:

- Verifiability is required at all levels in the system integration
  - down to smallest element
- Unnecessary complexity is avoided
- Behavior is unambiguously specified
- System behavior is composition of behaviors of its elements.
  - No unpredictable behavior emerges.
  - Then, if all elements are verified individually, their composition is also considered verified.

- Interactions precluding complete verification are not

Interference: Hazards → Constraints

Examples of hazards contributed through interference:

- Connection across lines of defense blurs the lines
- Connection across redundant elements compromises redundancy
- Summing of signals at voting logic compromises their sources
- A monitor is compromised by the monitored element
- A safety function is compromised by a non-safety element

Examples of derived constraints:

- Freedom from interference is assured provably across:
  - Lines of defense
  - Redundant divisions of system
  - Degrees of safety qualification
  - Monitoring & monitored elements of system
- Interactions limited provably to those required for safety functions
- Interactions and interconnections that preclude complete verification are avoided, eliminated, or prevented

Comprehensibility : Hazards → Constraints

Hazard scenario contributed through non-comprehension:
System behavior is not understood in the same way by its community of users…

Examples of derived constraints:

- Behavior is understood completely, correctly, consistently, and unambiguously by different users interacting with the system
- The allocation of requirements to some function and that function to some element of the system is bi-directionally traceable.
- The behavior specification avoids mode confusion, esp. when functionality is nested
- The architecture is specified in a manner (e.g., language; structure) that is unambiguously comprehensible to the community of its users, e.g. reviewers, architects, designers, implementers, etc., i.e. the people and the tools they use.

From characteristics to constraints Flow-down - 1/5

Contributory hazard	Conditions that reduce the hazard space
Characteristic	**Derived constraint**
Verifiability	Required property, flowing down to the finest-grained element. Verifiability checked at every phase of the development lifecycle, at every level of integration, before proceeding further. Examples of conditions for verifiability: • Ability to create a test (or verification) case; • Observability; • Ability to constrain the environment. Evidence of verifiability in verification plan.

From characteristics to constraints 2/5

Characteristic	Derived constraint
Analyzability	Avoidance of unnecessary complexity. Behaviour unambiguously specified Flow-down is correct; 1. Allocated behaviors satisfy behavior specified at next higher level of integration; 2. Unspecified behavior does not occur. System behaviour = ©{behaviours of its elements} Development follows a refinement process.

U.S.NRC
UNITED STATES NUCLEAR REGULATORY COMMISSION
Protecting People and the Environment

From characteristics to constraints 3/5

Characteristic	Derived constraint
Unanalyzed or un-analyzable conditions	System is statically analyzable. 1. All states, including fault conditions, are known. 2. All fault states, leading to failure modes, are known. 3. Safe state space of the system is known.

From characteristics to constraints 4/5

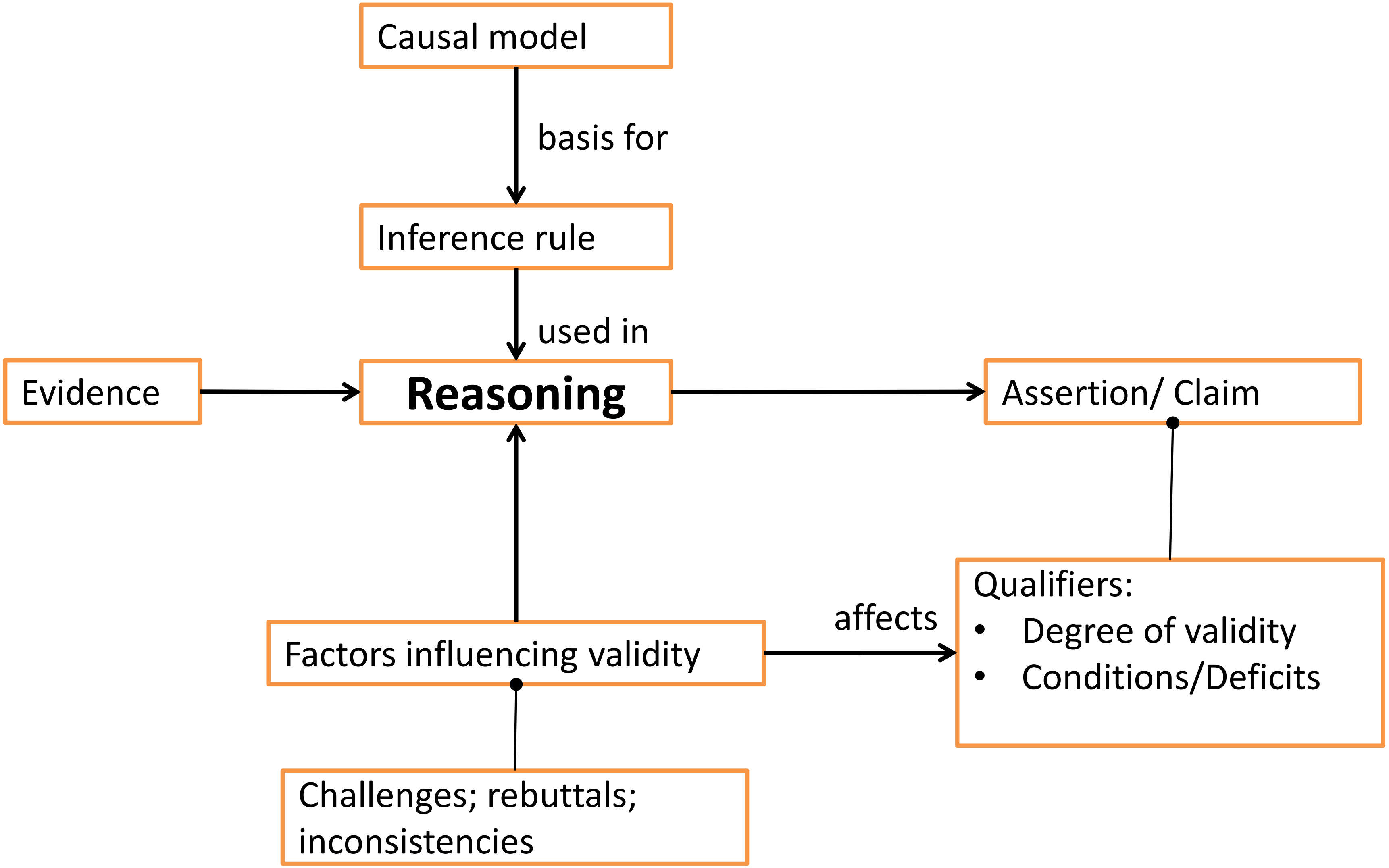
Characteristic	**Derived constraint**
System behaviour deterministic	System has a defined initial state. System is always in a known configuration. System is in a known state at all times.
System behaviour predictable	Each state transition is specified and known, including transitions corresponding to unexpected conditions. A hazardous condition can be detected in time to maintain the system in a safe state.

U.S.NRC
UNITED STATES NUCLEAR REGULATORY COMMISSION
Protecting People and the Environment

From characteristics to constraints 5/5

Characteristic	Derived constraint
Comprehensibility	Behaviour is completely and explicitly specified. Behaviour is completely understandable. Behaviour is understood completely, correctly, consistently, and unambiguously by different users interacting with the system The allocation of requirements to some function and that function to some element of the system is bi-directionally traceable. No mode confusion, esp. when functionality is nested. Architecture spec. is unambiguously comprehensible to the community of its users - people and the tools they use.

Reasoning structure



Questions/Discussion

U.S.NRC

United States Nuclear Regulatory Commission

Protecting People and the Environment

Other Information for Reference

U.S.NRC
UNITED STATES NUCLEAR REGULATORY COMMISSION
Protecting People and the Environment

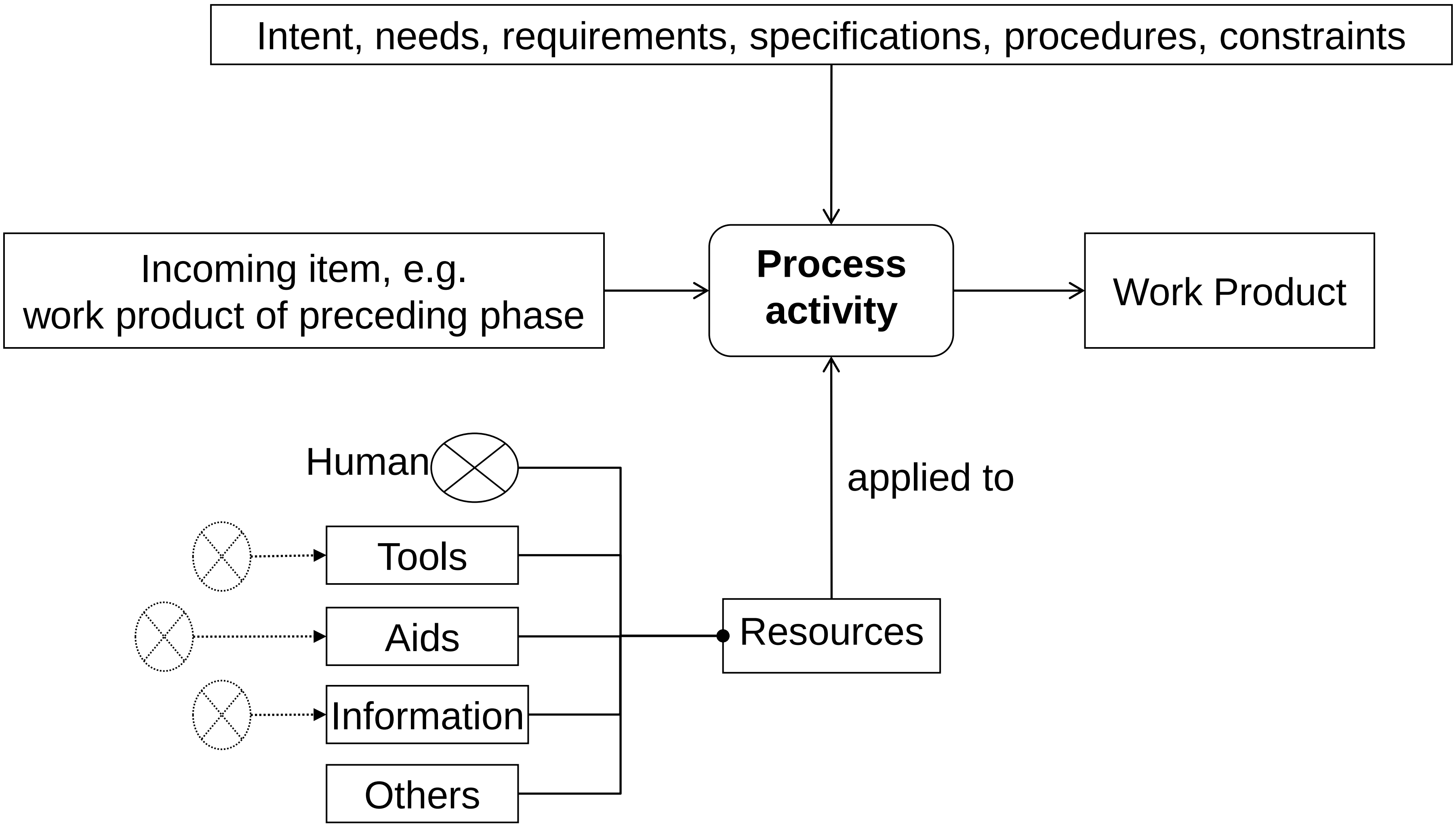
Acronyms 1/2

DARPA	Defense Advanced Research Projects Agency
DI&C	Digital Instrumentation and Control
EPRI	Electrical Power Research Institute
HACMS	High Assurance Cyber Military Systems
I&C	Instrumentation and Control
IEC	International Electrotechnical Commission
IEEE	International Electrical & Electronics Engineering Society
ISO	International Standards Organization
JTC1	Joint Technical Committee 1
min.	minimum
NPP	Nuclear Power Plant
NRC	Nuclear Regulatory Commission

Acronyms 2/2

QA	Quality Assurance
QM	Quality Management
R&D	Research and Development
RE	Requirements Engineering
RES	NRC Office of Nuclear Regulatory Research
RIL	Research Information Letter
spec.	specification
SQuaRE	Software Quality and Requirements Engineering
SW	Software
USC	University of Southern California
V&V	Verification and Validation
Vocab	Vocabulary

Dependency on a Process Activity



Key factors affecting HA Quality