From abstract performance-based regulatory requirements to verifiable engineering requirements: Some challenges

Software Certification Consortium meeting
May 15-16, 2025

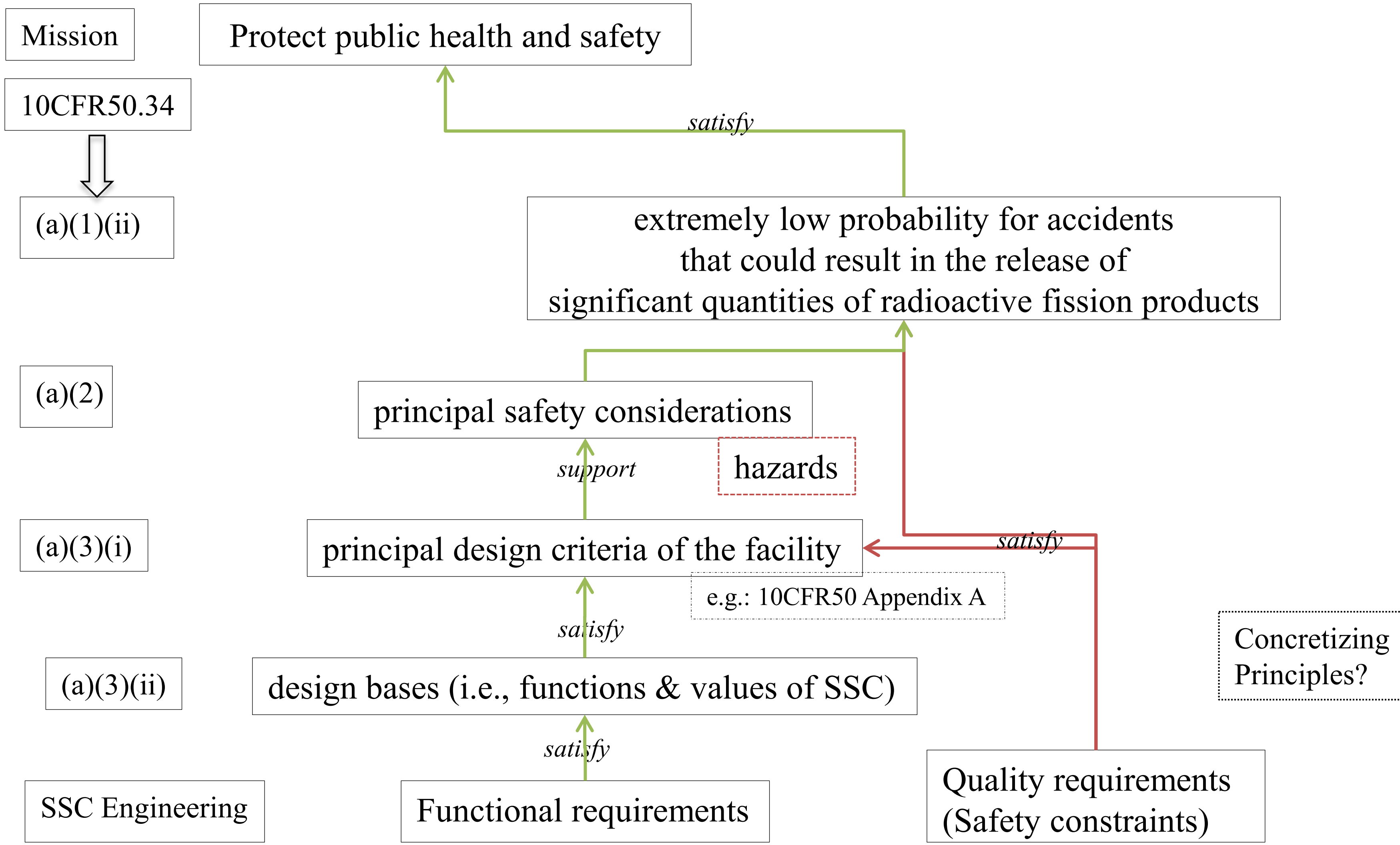
Sushil Birla, Office of Nuclear Regulatory Research
Norbert Carte, Office of Nuclear Reactor Regulation
U.S. Nuclear Regulatory Commission

"Performance-based": Multiple meanings

- Risk-Informed Performance-Based (RIPB) regulatory approval process
  - Industry desire.
  - NRC's long-standing direction.
- RI … reactor protection systems: From reactor safety analysis.
  - Based on assumptions with some validity challenges:
    - Independence across different reactor and plant level functions and systems:
      - Networks span across them.
      - Other shared resources.
- PB:
  - Confusion from different usages of the term "performance".
  - Approach oriented to the outcome[1] (in our context: SAFETY)
    - Flexibility in how the outcome is achieved.
    - Contrasted with prescriptive approach.
  - Even within this context, many differing views[2].

[1] National Academies, "Performance-based Safety Regulation"
[2] ANS Standards Board RP3C discussion, "Where is the PB in RIPB? …" R. Franovich, N.P. Kadambi, May 31, 2024

PB: Flow down from regulation



Example Selected for Illustration

Reactor type: Large light-water operating reactor

Salient safety-relevant characteristics:

- High consequence
- Low safety margin
- Changes in environment of reactor protection systems
  - Configuration control granularity may not be adequate.
  - Cumulative effects may not be understood or analyzed adequately.

SSC type: Digital I&C system for reactor protection (e.g., for trip function)

Salient safety-relevant characteristics:

- Safety function: Relatively simple combinatorial logic.
- Safety-criticality level: Highest. (Requires highest assurance level).
- Achieving this assurance level without prescriptive means is challenging:
  - Increased potential for hazards from systematic causes
    - Increased potential for unnecessary coupling.
    - Increased potential for unspecified interactions & emergent behavior.

Implication: Safety constraints must be specified very precisely.

Some principles to concretize safety constraints

The concretization leads to engineering requirements (safety constraints)

- that satisfy the safety goal.
- that are implementable, i.e.
  - design is feasible within the state of the art.
- that can be consistently verified & validated
  - validation and verification (V&V) can be performed within the state of the art.
- that do not constrain the design space unnecessarily.
- that do not ...
- …

Some implications:

1. Concretions will be more restrictive than the theoretically possible.
2. The solution space will expand as the state-of-the-art advances.
3. Different organizations will make different choices in their solution space.
4. Therefore, regulatory organizations would have to expand their capabilities.

Example hazard to derive quality requirements

Reactor level hazard contributor	Reaction out of control	Excess Heat	Significant amount of radioactive fission products released
Type of hazard control	Prevention		
Type of hazard controlling system	Reactor trip system: (RTS)		

RTS-internal hazard	Trip not executed correctly, timely
RTS-internal Hazard contributor	Anything that prevents correct timely execution of the trip function

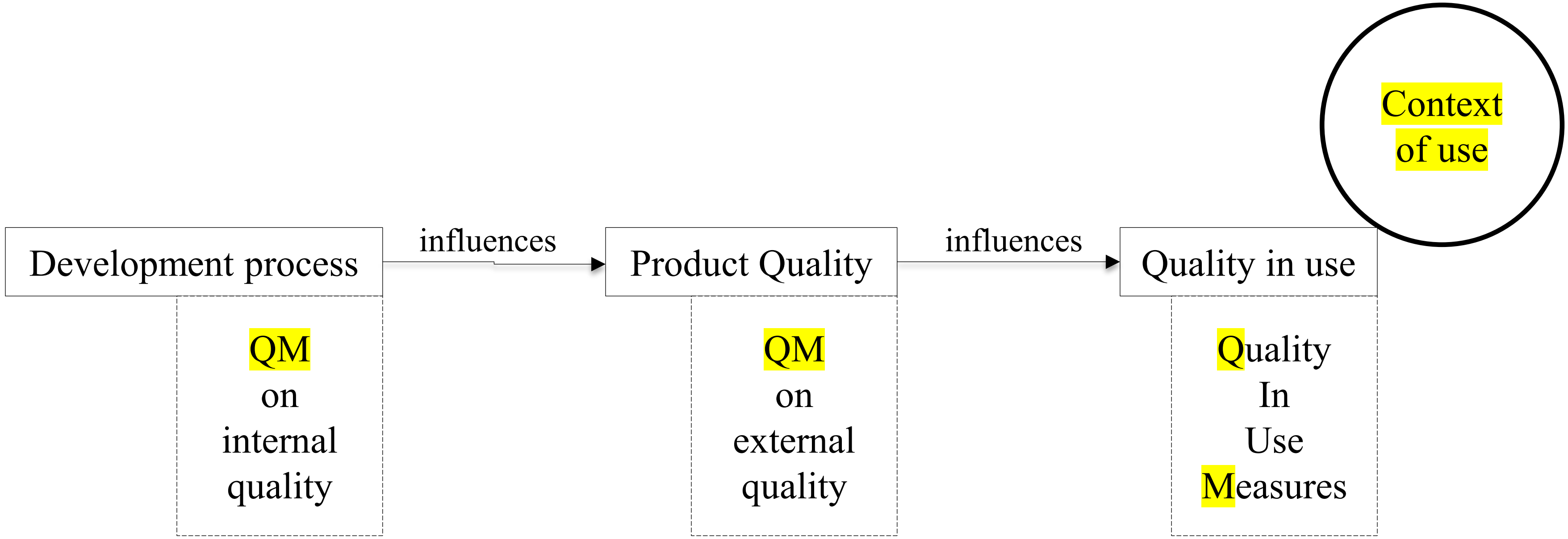
PB concretion criteria for identifying hazards

Hazard Identification (HI) is propagated to whatever the safety function depends on (until conclusive evidence of control is obtained), e.g.:

- Some other function
- Some other system
- Information or data
  - Its value
  - Its timely availability (incl. freshness)
- Component within the system
  - Its state (is it in the state needed)
  - Its behavior
- Processes
  - Engineering (Requirements; Design) ← competence ← culture
  - V&V ← (Test specification; oracle) ← Independence
    ← competence ← culture
  - Integrity (e.g.: configuration management; version control; safety assurance)
    ← Independence

See RIL-1101

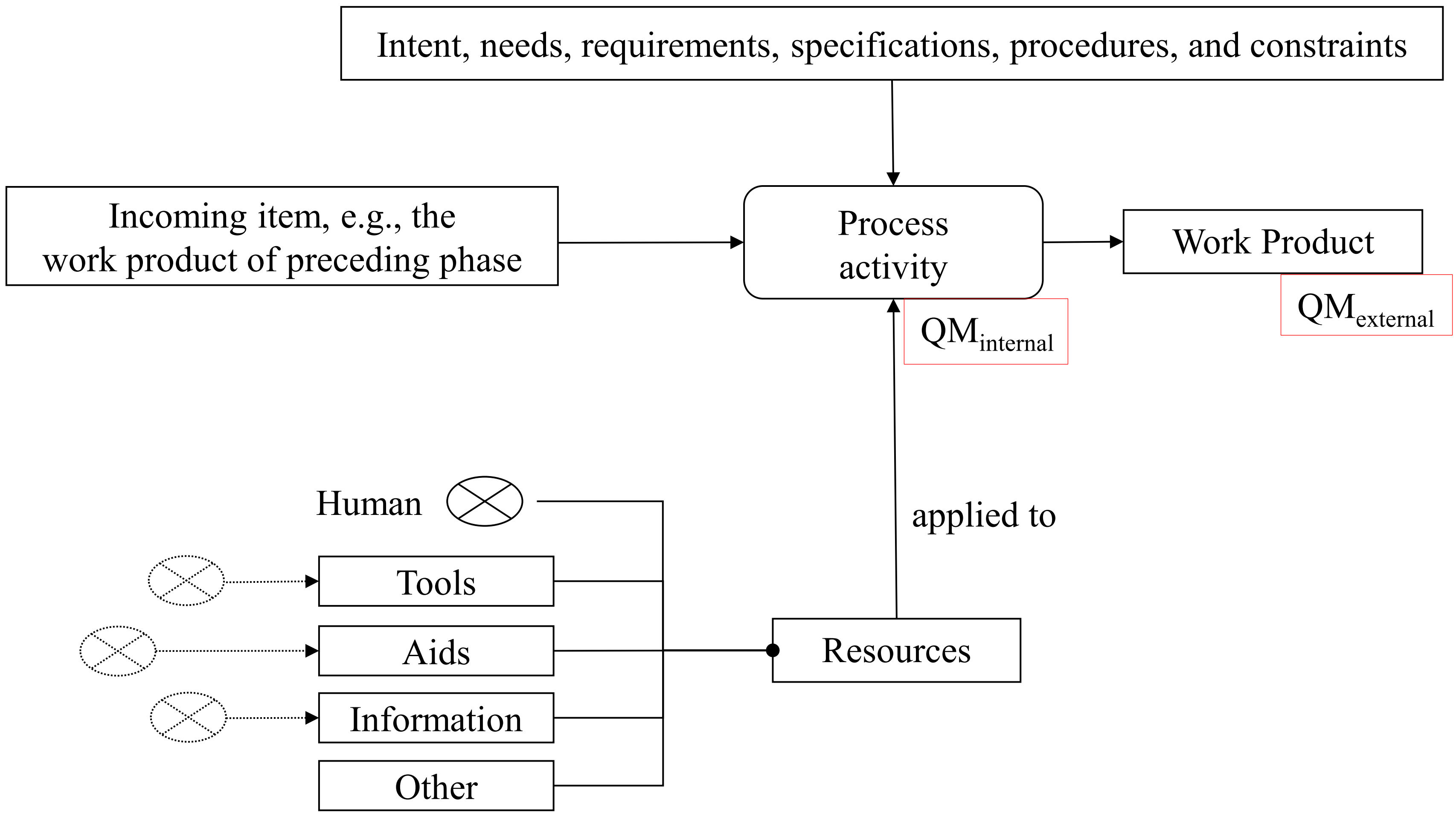
Quality Measures (QM) ─ performing hazard-free



Source: ISO/IEC 250xx series
Systems and software engineering
─ Systems and software Quality Requirements and Evaluation (SQuaRE)

See HPR-382:C4.3

Process-dependent sources of hazards: Examples



PB safety evaluation endpoint

All hazards and hazard contributors are identified

Not feasible for any arbitrary design!

HWR-1288

All hazards and hazard contributors are controlled

No uncontrolled hazards can be found

This does not mean none exist!

Preference order of hazard control approaches

1) Prevention

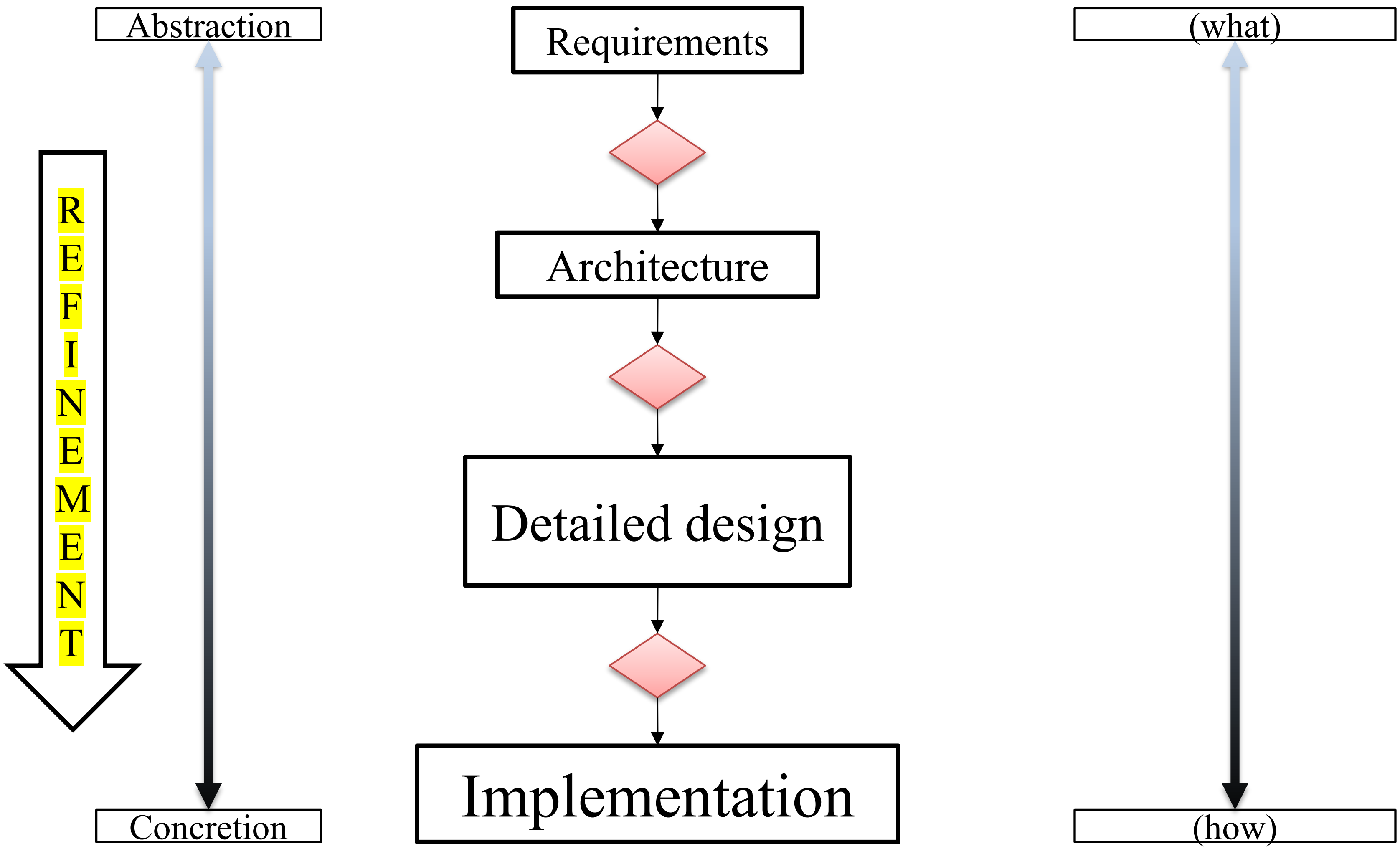
2) Containment (prevention of propagation)

3) Mitigation internal to the safety system
(e.g., monitoring, fault detection, intervention)

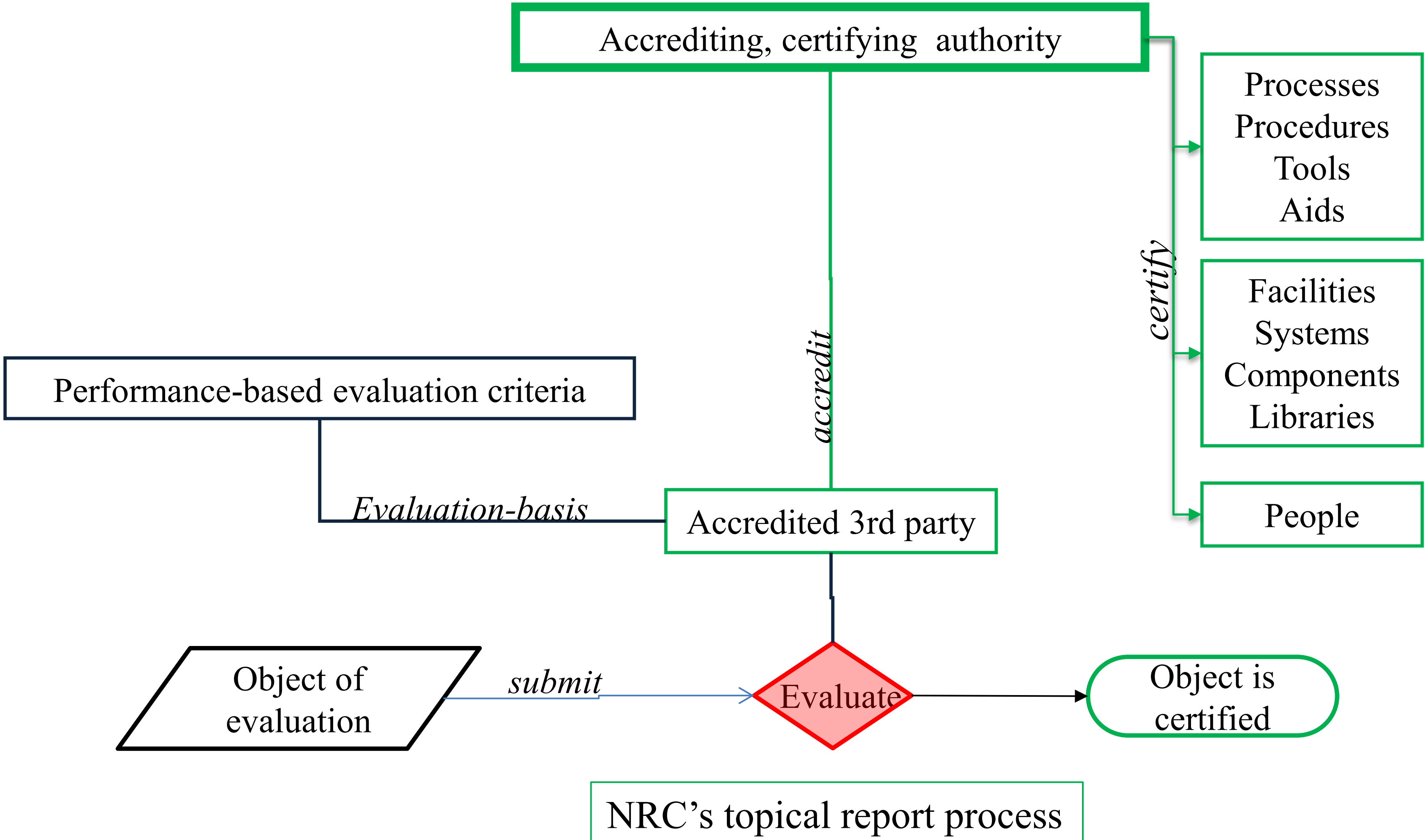
4) Mitigation external to the safety system
(e.g., a diverse actuation system)

Note:
1. This preference order is aligned with the "shift to the left" movement in the US Dept of Defense.
2. It also supports the US industry desire to avoid diverse systems for achieving the target assurance.

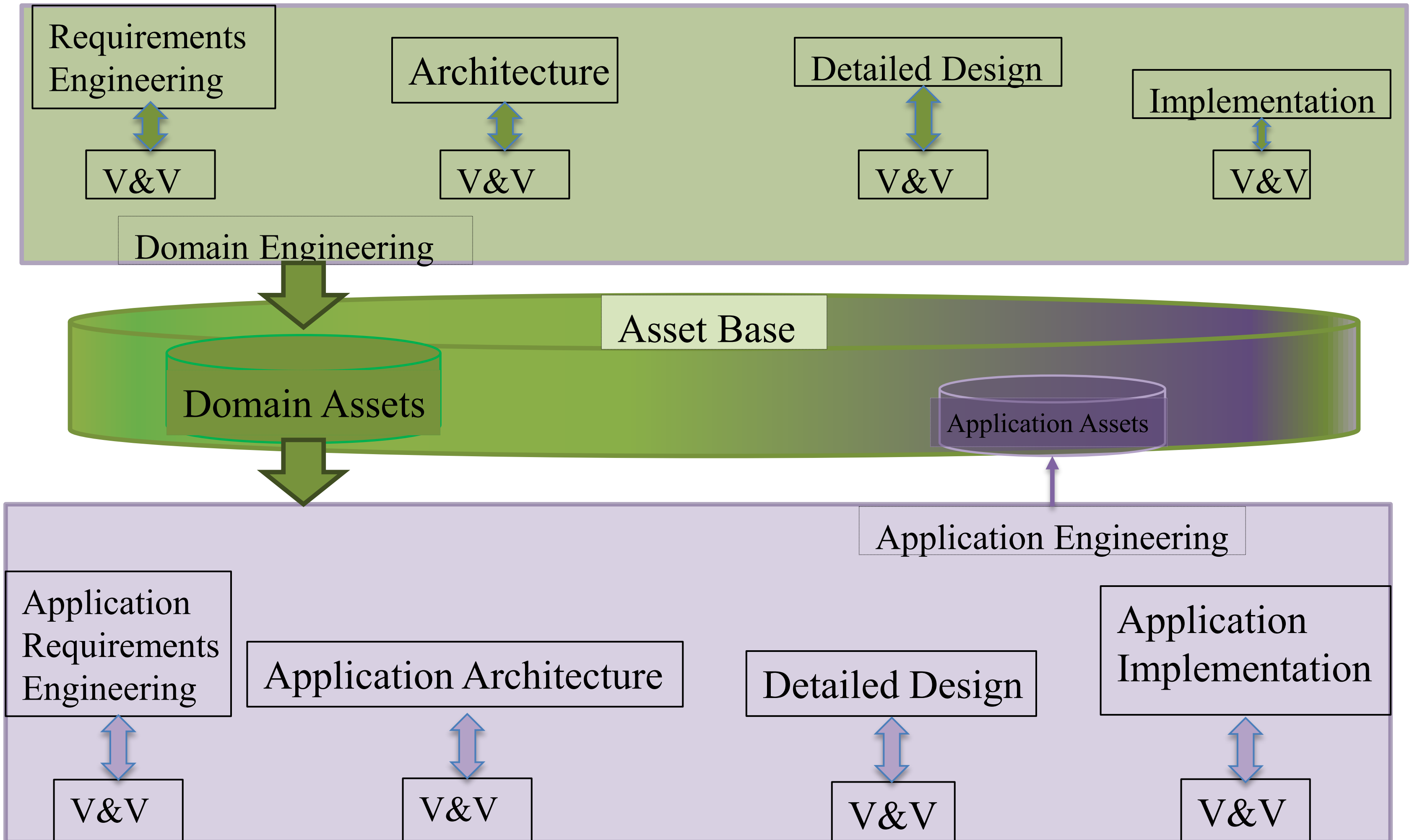
Prevent hazard-contributing conditions



Process to pre-certify building blocks



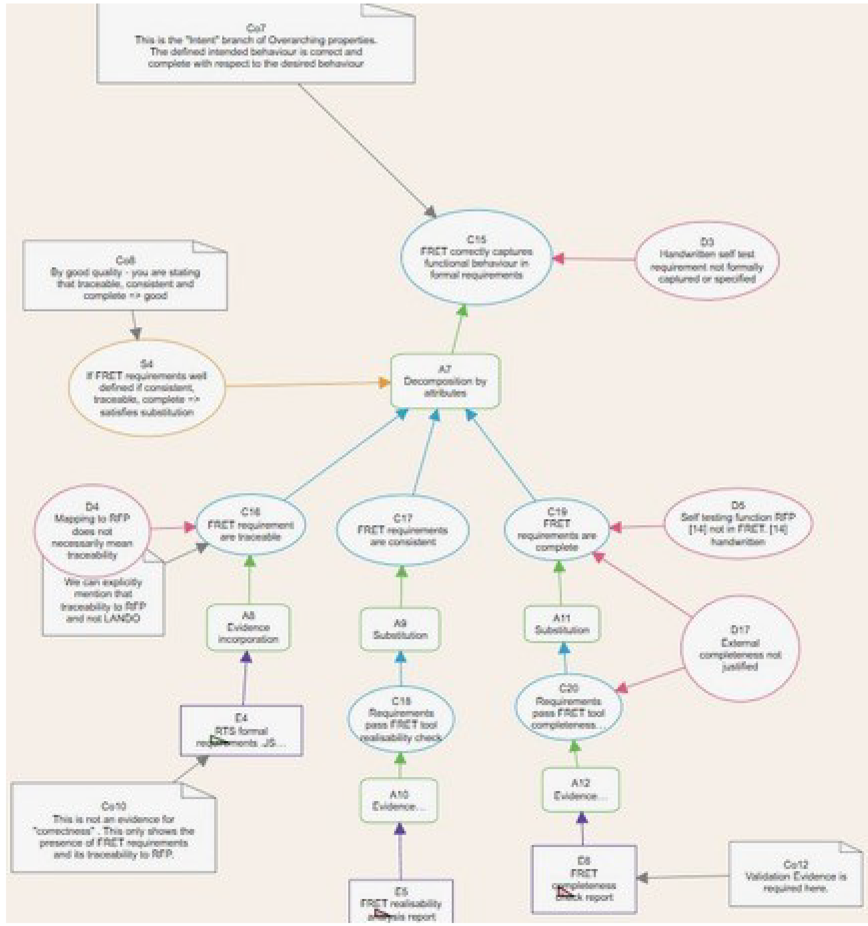
Reuse of pre-certified assets



Source: ISO/IEC 26550:2015(E)

Organizing Evidence in the Form of an Assurance Case

Example:
Assurance Case
fragment
for independent review



From the evidence at the leaf nodes to the top-level safety claim, all the relationships must be explicit.

ADELARD part of nccgroup

Challenges 1/2

- Ambiguity in requirements
- Abstraction:
  - Making the right abstractions
  - Ability to think in terms of abstractions
  - Teachability; Learnability.
- Composability
- Context of use (ODD; ConOps). Unknowns.
- Lack of mature, certifiable high-integrity development environments
  - Languages
  - Tools
  - Libraries
  - IDE

Challenges 2/2

- Competence.
  - Lack of competence models
  - Lack of training curricula and institutions
  - Lack of qualifying/certifying agents for people
  - How can Knowledge Engineering help?
- Lack of third party certification infrastructure.
- Economics: Little economy of scale.
- Local optimization rather than global.
- Barrier to entry
- Culture
- Decision-makers do not understand:
  - Multitude of ways in which software-based systems can be impaired
  - What would be “reasonable assurance”
  - What it takes to get reasonable assurance