DRAFT

Roadmap: Assurance capability

Cyber-Physical Systems, e.g., for Safety of Nuclear Power Plants

Draft Document Date of last editing: 2015-04-07 Last editor[1]: Sushil Birla[2]

1. Purpose

Facilitate inter-agency directional coordination in long-term R&D and transition of the share-able common-core (see Figure 1) technologies to realize "designed-in assurance" [1] of cyber-physical systems (CPS); examples: a digital safety system for a nuclear power plant (NPP); a life-critical medical device such a pacemaker; an insulin infusion pump.

This draft is a proposal for NRC to explore inter-agency coordination in "Designed-in Assurance R&D" for Cyber Physical Systems (CPS):

- This roadmap is a response to the joint OSTP-OMB memo [2] on the subject of science & technology priorities for the FY 2016 budget.
- This roadmap is intended to support the NRC strategic plan [3], with respect to interactions with other Federal agencies for improving its regulatory infrastructure.
- This inter-agency coordination roadmap is intended to support the development of NRC's long-range Safety Assurance R&D roadmap for NPP digital safety systems.
- NRC's long-range Safety Assurance R&D roadmap for NPP digital safety systems is intended to lead into NRC's 2015-2019 "instrumentation and control" (I&C) research plan.

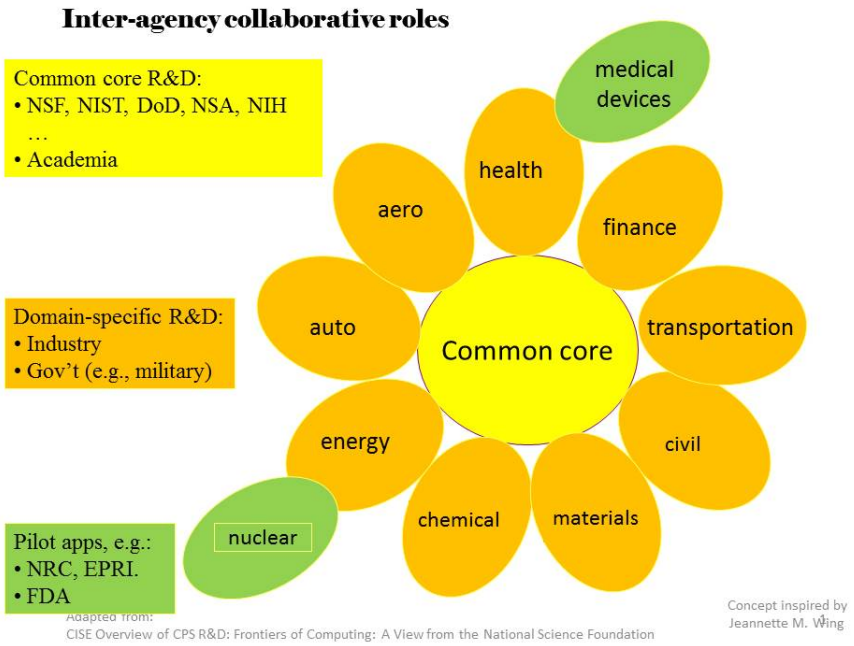

Figure 1: Common core R&D base supports many application domains

[1] This draft assimilates ideas generated in discussions with the SEI on August 5-6, 2014 and subsequent reviewers.

[2] The content does not reflect a position of the NRC, but is the author's draft proposal for review by others.

Purpose and focus of this version of the draft proposal: Facilitate inter-agency communication to evolve this roadmap to the next actionable state, e.g.: commitment for coordinated progress, consistent with best practices for federal R&D [4]. Information in [5] is a suggested starting point for coordination, for which NRC-FDA discussions have already started.

2. Scope

Coordination across different federal agencies supports the following objectives:

1. Identify common problems, issues, and challenges, reframed to enable coordination.
2. Avoid duplication of effort.
3. Share a common core technology base, which an agency can reuse or leverage to perform additional R&D specific to the application domain of its interest. See Figure 1.
4. Learn from each other to pursue agency-specific R&D more effectively and efficiently.

Scope of "Assurance" considers "losses of concern"[3] to the stakeholders of the system being assured. For example:

1. "Nuclear safety" is the primary concern in the nuclear industry.
2. To the extent that security is also a hazard to nuclear safety, it is a growing concern.
3. To the extent that other losses of concern to an NPP operator, such as "loss of production" and "damage to equipment," may conflict with nuclear safety, this influence also concerns the regulator.

Transition of the technologies to realize "designed-in assurance" of a CPS means maturation of the capability to an economically self-sustaining level, e.g.:

1. Maturation of the technological infrastructure needed for self-sustenance, e.g.:
   1.1. Pre-certified processes, methods, techniques, and procedures.
   1.2. Pre-certified tools.
   1.3. Pre-certified facilities.
   1.4. Pre-certified reusable assets.
   1.5. Third party certification services for all elements requiring certification.
   1.6. Accreditation infrastructure to accredit third party certification services.
   1.7. Standards and guidelines for items to be certified.
   1.8. Measurable criteria contributing to assurance.
2. Maturation of a participating organization's capability needed for self-sustenance.
3. Maturation of a complement of human competence [6] needed for self-sustenance.
4. An intrinsically self-sustaining economic cycle, based on:
   4.1. "Doing it right the first time" (aka "correct by construction"; "designed-in assurance).
   4.2. Competitive lifecycle economics.
   4.3. Competitive time to market.
   4.4. Growth of the acquirer-provider community and marketplace past a critical mass.
   4.5. Effective, efficient R&D and Transition through inter-agency coordination.

[3] Any condition leading to a loss of concern is treated as a hazard.

3. Organization of the roadmap

This roadmap identifies a plausible sequence of transitions to grow capability from the current state to an envisioned goal state. Figure 2 shows a generic template.

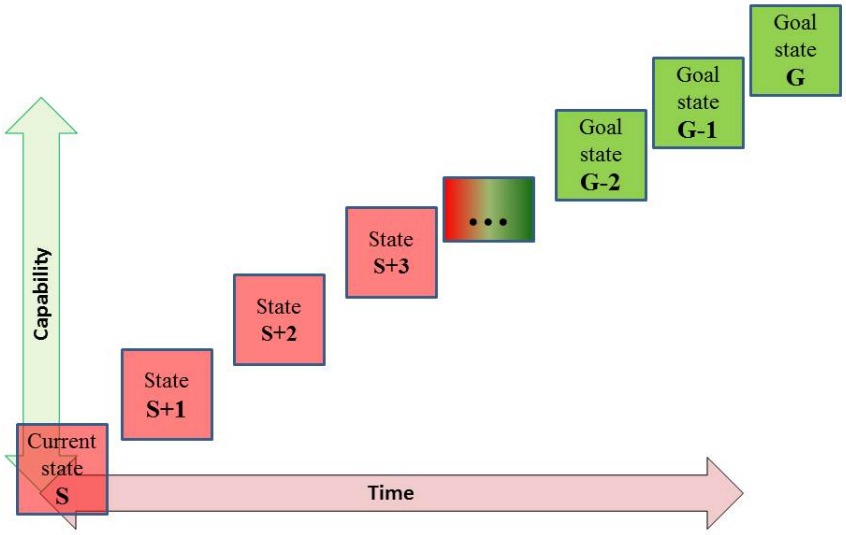

Figure 2: Incrementally Evolving Assurance Capability – template

See Appendix B for a broad-brush characterization of the envisioned goal state, the current state, and the immediate next state. See Appendix D for considerations in designing other intermediate states.

4. Intent of identifying intermediate capability-states

The intent of the road-mapping approach depicted in Figure 2, identifying intermediate capability states, is to:

1. Demonstrate measurable, visible benefit or incentive to progress further.
   1.1. Shake out the tool-supported environment[4].
2. Evaluate[5] the results. Revise the roadmap accordingly. (Built-in learning cycles).
3. Characterize later (closer-to-goal) states to the degree of specificity needed for establishing plausibility.
4. Characterize the next few (closer-to-current) states to the degree of specificity needed for establishing feasibility, forecasting resource demand, reserving critical resources, and defining the next immediate state.

[4] You only get one chance!
[5] It includes evaluation for decision to terminate, continue, or modify the plan.

5. Characterize the next immediate state to the degree of specificity needed for creation of a work plan or statement of work.

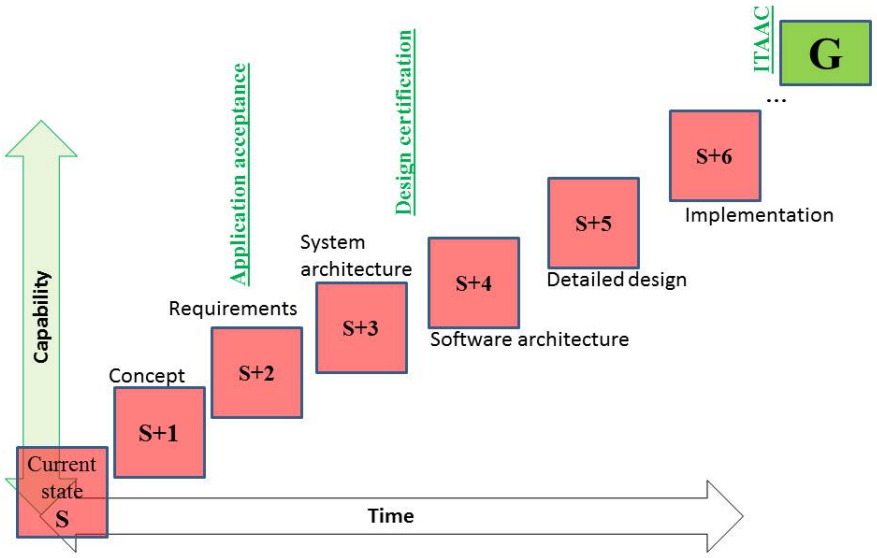

Figure 3: Evolve Assurance Capability Incrementally - NPP Safety System Case

Figure 3 illustrates the intent of the road-mapping activity for the case of an NPP safety system. Strategy of characterizing states S+1 and S+3 is explained below.

State S+1: Hazard analysis (HA) driven conceptual engineering capability is demonstrated to show benefits (e.g.: reduced number of meetings to reach convergence on the concept of diversity and defense in depth; reduced "regulatory uncertainty") to the applicant and to the regulator in supporting the informal, pre-application meetings. Also see Appendix B.

State S+3: Capability up to the stage of engineering the System Architecture includes verifiable requirements and constraints, which, when satisfied, assure that no unintended behavior can occur and the specified behavior will be realized correctly. Demonstrated benefits to applicant and regulator: Shorter time and reduced cost to get design certification; insignificant "regulatory uncertainty".

5. Holistic approach to develop assurance capability

Capability development is characterized along five dimensions or tracks as follows:

1. Infrastructure for conformity assessment
2. Integrated development and assurance environment
   2.1. Tools
   2.2. Reusable Assets
   2.3. Repository
3. Business case development

4. Community development
5. Workforce development

6. Infrastructure for conformity assessment

Examples of elements in an infrastructure for conformity assessment include the following:
1. Standards and guidelines for items to be certified.
2. Measurable criteria contributing to assurance.
3. Test-beds.
4. Third party certification services for all elements requiring certification.
5. Accreditation infrastructure to accredit third party certification services.
6. Support to pre-certify processes, methods, techniques, and procedures.
7. Support to pre-certify tools.
8. Support to pre-certify facilities.
9. Support to pre-certify reusable assets.

Table 1 characterizes the current state and envisioned goal state of some relevant standards and guides. A commensurate infrastructure is needed to evaluate and validate the standards, guides, and criteria as these evolve, and, eventually, to assess conformity to the standards.

Table 1: Standards and guidance: Current state and envisioned goal state

Category	Current state	Envisioned Goal State
Concepts. Vocabulary. Terminology. See Annex A	Inconsistent. IEEE STD {100; 610} ISO/IEC/IEEE STD 24765 ISO/IEC Guide 2 ISO/IEC Guide 51 ISO/IEC Guide 99 IEEE Std 982.1 ISO/IEC 2382- ISO/IEC 25000: ISO/IEC 25001: ISO/IEC 25020: ISO/IEC 25051: ITU-T Recommendation Z. 100 Specification and Description Language (SDL)	Consistent across CPS domains. Support Learnability, Usability.
	Confusing. Unclear context dependencies.	Clear for each context and across contexts.
	Ambiguous. Multiple meanings. Unclear semantic relationships across different terms.	Unambiguous semantics, incl. • Compositions • Other relationships. Readily understood, e.g.: • Supporting ontologies. • Visualize-able.
Assurance case (e.g.: Safety case)	• Labor-intensive: Not integrated with development processes. • Not easily maintained; gets	Framework for goal-driven assurance of a critical CPS: • Specialize-able for an NPP safety system.

Category	Current state	Envisioned Goal State
	inconsistent over time. • Not used much[6]. • Safety analysis report (SAR) is based on clause-by-clause compliance with standards / guides. • Not integrated with plans (e.g. IEEE Std 1228) • Relationship from a clause in a standard to the safety goal: No explicit reasoning path. • ISO/IEC 15026-3 ties assurance to safety integrity levels, which are based on risk-quantification – method does not suit safety-critical CPS. • ISO/IEC 15026-2 provides a structure for an assurance case and defines supporting terms: Claim Evidence Argument (reasoning) Assumption Uncertainty; etc.	• Linked to supporting standards. • Case built efficiently. • Case helps manage change. • Case is compose-able. • Team reasoning[7] support. • Automated reasoning support. • Integrated with development: "Designed-in / Built-in Assurance." • Includes qualifiers[8] and deficits. Available: • Complement of consistent standards: o Common core. o Agency specializations. • Exemplars. • Infrastructure for 3rd party certification / accreditation: o Pre-certif. processes o Pre-certif. methods o Pre-certif. techniques o Pre-certif. procedures o Pre-certif. facilities o Pre-certif. tools o Pre-certif. libraries o Pre-certif. architectures o …. o Pre-certif. people
Hazard Analysis (HA)	• IEEE STD 1012 [16] requires HA at every phase in the development lifecycle. • Inconsistent, confusing scope and definitions across standards. • Not integrated with development processes. • Little guidance to evaluate HA. • HA not performed well. o Inadequate guidance: process-oriented. o Inconsistent results.	• Performed with consistency. • Evaluated with consistency. • Supports early evaluation of supply chain. • Integrated in assurance framework – see above. • Information integrated with development processes; drives development. • Facilitates team reasoning. • Automation-supported.

[6] Little use of safety cases outside the UK.
[7] Systematic use of judgment, with repeatable, consistent outcome.
[8] Support team-reasoning, using appropriate measurement scales.

Category	Current state	Envisioned Goal State

Validation	• IEEE STD 1012 [16] requires validation at every phase in the development lifecycle.
	• Validation of initial requirements is weak.
	• Quality[9] requirements: Very weak.
	• Confused with verification.
	• Dependent on implicit knowledge.
		• Performed with consistency.
		• Evaluated with consistency.
		• Integrated with HA.
		• Integrated with requirements elicitation & analysis.
		• Evaluation of completeness:
		  ○ Formalized.
		  ○ Based on explicit evidence[10].

Verification	• Testing oriented: Too late in the process.
	• Weakened by weak requirements, e.g.:
	  ○ Unintended behavior.
	• No uniform criteria for early lifecycle use.
	• Weak learning cycle.
	• Existing relevant standards are process oriented (lack product-evaluation criteria); examples:
	  ○ IEEE Std 1012 [16]
	  ○ IEEE Std 1008
	  ○ IEEE Std 1028-
	  ○ IEEE Std 1044-
	  ○ IEEE Std 1061-
	  ○ IEEE Std 829-
	  ○ ISO/IEC 14598-
	  ○ ISO/IEC 15939
		• Verification specs & cases are reasoned derivatives of requirements.
		• “Correct by construction”; built into development.
		• Combination of analysis and testing: Reasoning-based proof of coverage.
		• Compose-ability.
		• Effort is shifted upstream.
		• Reduced downstream costs, time.
		• Drives learning.

7. Integrated Development & Assurance Environment

“Integrated Assurance & Development Environment” includes” support to perform all process activities, methods, techniques, procedures, and work flows needed to produce a certified work product. Examples of support include:

1. Tools, integrated in a compatible suite:
   1.1. Model creation.
   1.2. Analysis.
   1.3. Code generation.
   1.4. Repository[11] for models.
   1.5. Repository for other artifacts.
   1.6. Coverage for entire lifecycle.
   1.7. etc.

[9] So-called non-functional requirements
[10] Not dependent on intuition, implicit knowledge, etc.
[11] Example functions: Storage, search, retrieval; Configuration management; Revision control; Fault tolerance.

2. Library of reusable assets:
   2.1. Includes everything needed to instantiate in the work product of each phase.
   2.2. Library supports applicant in selecting the asset needed.
   2.3. Library supports tracking "where used" and usage experience for each asset.
   2.4. Examples of elements in the library to support conceptual phase activities:
      2.4.1. Sensors.
      2.4.2. Actuators.
      2.4.3. HMIs.
      2.4.4. Usage contexts.
      2.4.5. Plans; processes; procedures.
3. Support to create and validate an Assurance Case incrementally from planning to lifecycle maintenance.
4. Support to perform hazard analysis (HA) from planning to lifecycle maintenance.
   4.1. Includes hazard logging and tracking.
5. Support Learnability.
6. Support Usability [43].[12]
7. Support migration from current state to envisioned goal state.

Table 2 in [5] characterizes the quality (richness) of information content in the current common practice (current capability), relative to best practice (state of R&D results with demonstrated effectiveness at least through a pilot), and the state of the art (R&D results whose effectiveness has not been piloted at an industrial scale) [5]. This information supports gap analysis and characterization of intermediate capability states to bridge the gaps.

7.1. Known gaps in integrated development & assurance environment

These gaps were identified in the context of reaching State S1 for a nuclear safety application domain. More gaps would be identified in broader application contexts.

1. Repository:
   1.1. How to organize; how to search.
   1.2. Support shared usage by a team; includes concurrent use.
   1.3. Capability of managing models:
      1.3.1. Version control.
      1.3.2. Configuration management.
      1.3.3. Consistency management.
      1.3.4. Release management: from one team to another, or externally.
      1.3.5. Configurable (adaptable) workflow management.
      1.3.6. Having a distributed model repository.

[12] Usability & learnability of technology and tools to create and assure digital safety systems.

1.3.7. Standardized model interchange formats.
1.3.8. Support shared usage by a team; includes concurrent use.

2. Tools: Capability maturation, e.g.:
   2.1. Learnability: Remove barriers to adoption and learning before asking them to adopt/learn.
   2.2. Usability.
   2.3. Commercial support.
   2.4. Scalability.
3. Tools: Capability to implement (through stepwise refinement) on FPGAs / CPLDs.
4. Tools: Integration into consistent, compatible suite.
5. Library of reusable assets[13]:
   5.1. Ontologies; also see item 6.5.
   5.2. Support for refinement.
   5.3. Includes conditions of use for which an element is pre-certified/approved.
   5.4. Reusable architectures (including analysis results and guaranteed properties).
   5.5. Library of assumptions on which development may be based:
      5.5.1. Ability to characterize assumptions.

[13] Language exists to support it; SAVI has something similar.

5.6. Each model element includes formalized underlying assumptions.

6. Support for requirements engineering:
   6.1. Improve throughout the lifecycle.
   6.2. Goal-oriented requirements organization.
   6.3. Better ways to formalize.
   6.4. More measurable.
   6.5. Ontologies with support of computational linguistics to convey the intended meaning.
   6.6. Identifying relationships across requirements (e.g., safety conflicts with performance).
   6.7. Stakeholder specific views/projections.
7. Support for strict stepwise refinement and composition:
   7.1. Rules of composition, including composition of quality attributes.

7.2. Current state of the development environment: Some "known"s

Table 2 characterizes common practice, best practice and the state of the art. The characterization is focused primarily on requirements, including safety and such other quality attributes, functional hazard analysis capability. Secondary focus is on architectural design. The characterization of capability for detailed design and implementation is limited to the context of a nuclear safety system. More information may be found in [7].

Table 2: Characterization of information richness in phase work products

Row ID	Work product (lifecycle phase-wise)	Common practice	State of the practice (best in class); examples	State of the art; examples
1	Requirements from next higher level of integration, e.g. from NPP-level safety analysis.	Textual narrative. No configuration-controlled vocabulary. "Flat list" organization (i.e., no explicit relationship across requirements is identified).	Restricted natural language with defined vocabulary and structure across elements of a statement [8].	Use case scenarios [9].
			SpecTRM-RL [10].	Framework for specification & analysis [11].
			Requirements engineering support in Naval Research Labs [12]. Requirements tables as used for Darlington NPP [13][14]. Models to support mechanized reasoning. Examples: SysML [15].	
2	Plans {Safety	Low level of detail;	V&V plan [16].	Integrated safety

Row ID	Work product (lifecycle phase-wise)	Common practice	State of the practice (best in class); examples	State of the art; examples
	plan; V&V plan; HA plan}.	relatively late in the lifecycle.	Safety plan [17]-[19].	and security plan.
3	Concept.	Combination of (a) block diagram without semantics on the symbols and (b) textual narrative.	Models to support mechanized reasoning [20] (See note 1). SysML [15]; AADL [21]; Extended EAST-ADL [22].	META [23].
4	Requirements of digital safety system.	See row 1.	See row 1.	See row 1.
5	Architecture of digital safety system.	See row 3.	See row 3.	META [23].
6	Requirements for software in digital safety system.	See row 1.	[20][24][25].	See row 1.
7	Architecture for software in digital safety system.	See row 3.	See row 3. MASCOT [25] ; AADL [21].	META [23].
8	Detailed design of software.	For application logic: Function block diagram [26]. For platform software: Combination of (a) block diagram without semantics on the symbols and (b) textual narrative.	SPARK [27][28].	META [23] Refinement from architectural specifications.
9	Implementation of software (code).	For platform software, including communication protocols: C programming language + processor-specific assembler language.	Concept of using safe subset of an implementation language: MISRA C [29][30]. Language for programming FPGAs [31].	Auto-generation from detailed design.

Notes:
1. The models should contain enough information to understand dependencies and propagation paths for contributory hazards.

8. Business Case Development

Develop a business model to reach a self-sustaining state. Learn from existing examples[14], e.g.:

- Eclipse Foundation
- Apache Foundation
- Linux

Develop an evolving business case, e.g.:

1. A model structure from previous studies in lifecycle economics of a similar transformation.
2. Initial analytical model, based on elimination of waste/cost, supported with existing data [7].
3. Revised model, based on learning from piloting[15] at each intermediate capability-state.
4. Extend model to facilitate cross-domain learning.
5. Improved validity of model from other pilots and subsequent applications[16].

Contributors to the business case - examples

1. Reduced cost through building it right the first time. Also see in Appendix B.2.1 item 2.
2. Value through finding the defects earlier in the development cycle.
3. Retention of consistency between the model used for engineering including analysis and the implementation (the real system placed in operation).
4. Easier compliance with regulation:
   4.1. Reduced effort and calendar time to prepare an application for design certification.
   4.2. Reduced effort and calendar time to prepare an application for license.
   4.3. Reduced effort and calendar time to get an application approved:
      4.3.1. Fewer rounds of “requests for additional information” (RAI) with regulator.

      Notes:

      - Different incentives for different stakeholders.
      - Detractors: Unintended disincentives.

5. Opportunities for relatively low increment of effort:
   5.1. Migrating existing plant modeling building blocks to the envisioned environment[17];
   5.2. Extending plant-level HA capability to include unintended behavior through interactions and feedback paths.

8.1. Current state of business case:

1. Intrinsic business case exists [7].
2. Opportune opening in transition from microprocessor-based platforms to FPGA-platforms.
3. Unclear understanding of lifecycle economics in a high-consequence industrial domain.
4. Unfamiliar business-case methodology based on future state (technology; conformity infrastructure; etc.)
   4.1. Not credible until effectiveness is demonstrated through an industrial pilot in the application domain.
5. Working example of business case methodology exists [32].

[14] Are there other examples known to any of the reviewers of this draft?
[15] Requires “design of experiment” and data gathering to learn from each pilot.
[16] Requires “design of experiment” and extended data gathering for continuing learning.
[17] Benefit derived through integrated model/information; consistency maintenance.

9. Community Development

1. Identify early adopters willing to try something new; learn from their experiences (across stakeholders).
2. Incentivize.
3. Engage in pilots.
4. Identify means/channels to communicate vision and technology, e.g.:

   4.1. Software Certification Consortium (SCC) [33].

   4.2. NIST CPS Program (Contact: Dr. Christopher Greer).
5. Spread awareness.
6. Grow critical mass for a self-sustaining marketplace (spanning the customer-supply chain spectrum).
7. Engage through different interactions with different stakeholders at different points in the roadmap.
8. Engage through standardization activities - see assurance framework above.
9. Extend to provider of a development environment for FPGA-based systems.
10. Extend to supply chain, e.g., developer of an FPGA-based component of a CPS.
11. Grow ways to crowd-source tools and reusable assets.

10. Stakeholders – examples

Government: NRC; FDA; DOD/ASD(R&E); U.S. Army; NRL; Naval Reactors; FAA; NASA; NIST; DOE-NE.

FFRDCs: SEI.

Academia: MIT; University of Kansas; McMaster Center for Software Certification at McMasters University; Vanderbilt University, Carnegie Mellon University, University of Virginia.

Industry: EPRI; INPO; Utilities / power plant operators; Westinghouse, GE; RADIY.

Mixture: Members of AVSI SAVI (NASA, FAA, Army, Airbus, Embraer, Boeing, universities, avionics suppliers such as Honeywell and Rockwell Collins).

11. Workforce Development

Needs to develop[18] the existing workforce include the following.

1. Identify the required competencies including the knowledge, skills and process abilities: – see Appendix C.
2. Create competency development model
   - 2.1. Create competency families
   - 2.2. Create graduated competency capabilities for each competency family:
     - 2.2.1. Useful to support training and development and staffing.
   - 2.3. Create competency resource profiles by competency at the individual and family level.
   - 2.4. Identify the "competency-based processes" used within each of the competencies.
   - 2.5. Identify competency based assets:
     - 2.5.1. Capture competency-based process.
     - 2.5.2. Capture asset use information – see under library of reusable assets item 2.3.
       - 2.5.2.1. Use information can support training and development activities.
   - 2.6. Capability to think and reason abstractly.
   - 2.7. Abstractions.
   - 2.8. Ability to perform & evaluate hazard analysis, esp. for effects of interactions:
     - 2.8.1. Broaden[19].
     - 2.8.2. Deepen.
   - 2.9. Ability to formulate constraints to control hazards, suitable for formalization.
   - 2.10. Ability to specify and evaluate architectures, including constraints.
   - 2.11. Architect's ability to communicate with diverse stakeholders across different perspectives / domains, and understand the implications of change.
   - 2.12. Improve communication quality for reasoning (e.g., based on the Toulmin model from which Figure 5 is adapted).

[18] Gaps in proven methods, techniques, curricula, tools, etc. to develop competence in the areas listed.
[19] Current state: Practice is based on FTA at level of the NPP system and FMEA, at the level of the I&C system.

2.13. Ability to elicit assumptions for unambiguous expression.

2.14. Ability to formulate assumptions for formalization.

2.15. Unlearning[20].

2.16. Cultural considerations: An enculturation process, including behavioral change processes considering the existing cultural diversity – typically, human adaptation and behavioral changes.

2.17. Human adaptation and behavioral change process of disorientation, dissociation, and reconnection. Process that people go through to effect change; SuZ Miller's work; analogous to dealing with a death (denial, negotiation,… acceptance).

2.17.1. Incentive structures (Figure 4).

2.17.2. Systematization of organizational change throughout the ecosystem (Figure 4).

2.17.3. Systemization to effect change throughout the ecosystem (Figure 4).

3. Develop a skilled work force; use the competency based model – see Appendix C.

3.1. Identify better ways to teach practitioners (explore the social sciences field), e.g., education-technology based development of working engineers [34][35].

3.1.1. Provide an experiential interactive learning environment that supports skill development across a range of competencies[21] that a team member should possess [34].

3.1.2. Teach proper actions, vocabulary and ways of thinking in a particular domain [34].

3.1.3. Use Kolb's concept of experiential learning for spiral of learning cycles.
- 3.1.3.1. Exposure to a concrete experience.
- 3.1.3.2. Reflection on that experience.
- 3.1.3.3. Generalization of the experience → formation of abstract concepts.
- 3.1.3.4. Application of these concepts to the concrete experience.

3.1.4. Use constructivist[22] instruction with the following goals:
- 3.1.4.1. Solving problems.
- 3.1.4.2. Reasoning.
- 3.1.4.3. Thinking critically.
- 3.1.4.4. Using knowledge actively, reflectively.

3.1.5. Meet conditions for effective constructivist learning:
- 3.1.5.1. Use of complex and realistic environments.
- 3.1.5.2. Use of social negotiation.
- 3.1.5.3. Multiple perspectives.
- 3.1.5.4. Learning modes for learners.
- 3.1.5.5. Encouragement for self-learning.
- 3.1.5.6. Self-awareness of knowledge construction.

[20] Prerequisite to paradigm shift

[21] Examples: System engineering; hazard analysis at each level of integration; architecture; embedded system engineering; timing analysis; hardware/software engineering; analysis for safety/security properties; integration; understanding integrated effects of software/computer hardware/system interactions.

[22] A theory that posits that learners construct their own knowledge rather than simply adopting pre-packaged knowledge.

3.1.6. Ask the right sequence of questions to obtain the correct root cause information.

3.1.7. Leverage domain engineering [36] (incl. library of reusable assets).

3.1.8. Engage subject matter experts, e.g., HA experts.

3.1.9. Leverage capability developed for HA of networked medical devices [37].

3.1.10. Leverage R&D and best practices elsewhere in the federal government [4].

3.2. How do we teach the work force the processes needed for designed-in assurance?

3.3. There are existing bodies of knowledge (BoK), such as architecture engineering and behavior analysis, that are not being applied; the gap isn't an R&D gap:

3.3.1. Organize this BoK for transfer to this domain. (How?)

3.3.2. Leverage the tool environment as the learning environment. (How?)

3.4. Include the supply chain.

4. Focus on knowledge used to drive the tool, not proficiency and speed with the tool.

Misc. notes on workforce development

1. The regulatory review role requires the same or greater competence as the third-party reviewer.

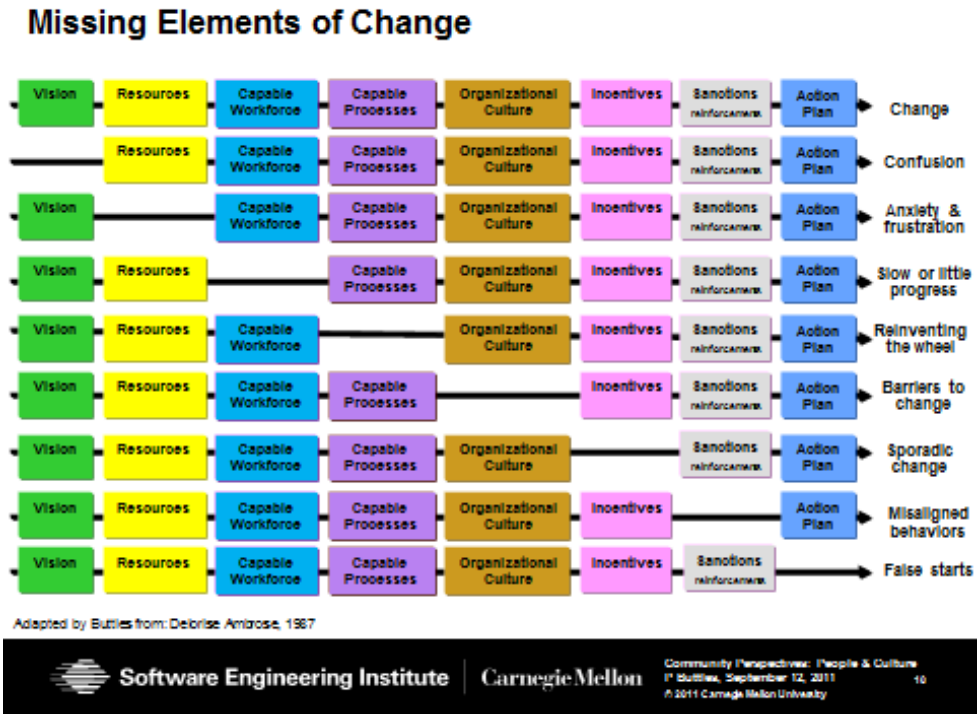

Figure 4: Systematize process for successful change

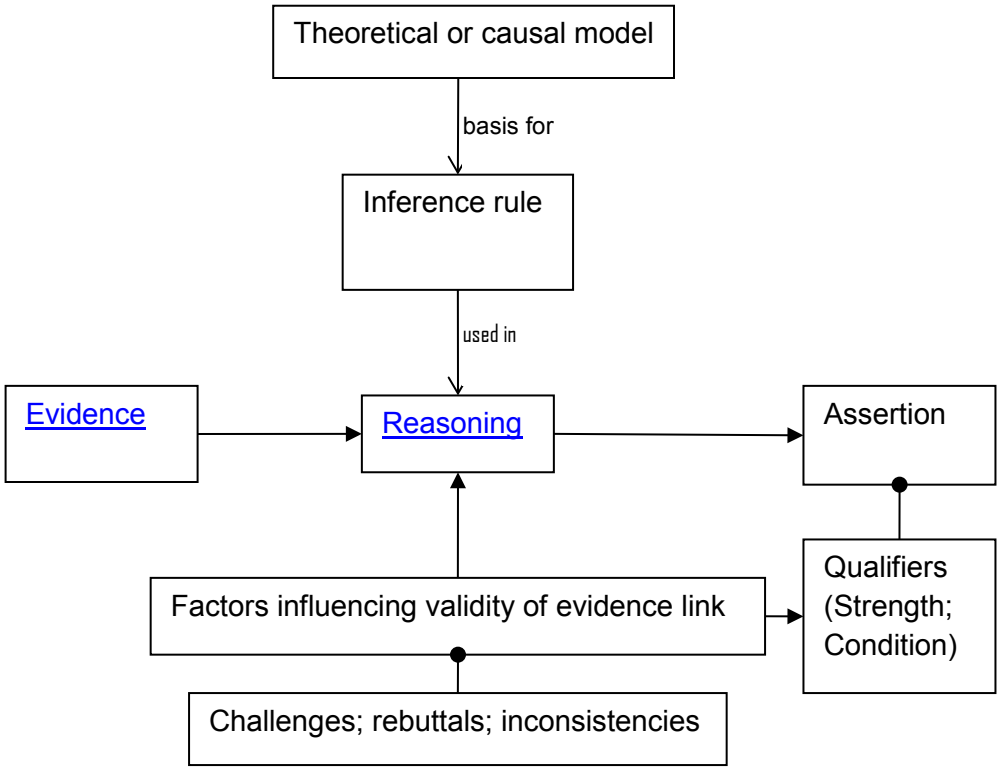

Figure 5: A framework for reasoning - adapted from the Toulmin model

12. References

[1] Trustworthy Cyberspace: Strategic Plan for Cyber-security R&D Programs. URL: http://www.whitehouse.gov/sites/default/files/microsites/ostp/fed_cybersecurity_rd_strategic_plan_2011.pdf.

[2] https://www.whitehouse.gov/sites/default/files/microsites/ostp/m-14-11.pdf .

[3] U.S. Nuclear Regulatory Commission (NRC) NUREG 1614 Vol. 6 "USNRC Strategic Plan Fiscal Years 2014-2018" August, 2014, Agencywide Documents Access and Management System (ADAMS) Accession No. ML14246A439.

[4] Best Practices for Federal Research and Development Partnership Facilities. URL: https://www.ida.org/~/media/Corporate/Files/Publications/STPIPubs/2014/ida-p-5148.ashx.

[5] Research Information Letter 1101,"Technical basis to review hazard analysis of digital safety systems" NRC. 2014. Also available at http://cps-vo.org/node/8758.

[6] Gilbert, T.F., "Human competence: Engineering worthy performance" John Wiley & Sons, 2007.

[7] http://cps-vo.org/node/17490 .

[8] Hinchey, A.G. et al, "Towards an automated development methodology for dependable systems with application to sensor networks" Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International, 2005.

[9] Allenby, K., Kelly, T., "Deriving Safety Requirements Using Scenarios," Proceedings of the Fifth International Symposium on Requirements Engineering, p.p.. 228-235, Toronto, Ont, Canada, August 7, 2002.

[10] SpecTRM-RL http://www.safeware-eng.com/software%20safety%20products/features.htm.

[11] Day, N.A., Joyce, J.A., "A framework for multi-notation requirements specification and analysis" Proceedings, ICRE 2000. URL http://ieeexplore.ieee.org/ielx5/6907/18574/00855551.pdf?tp=&arnumber=855551&isnumber=18574.

[12] Heitmeyer, et al, "The SCR method for formally specifying, verifying, and validating requirements: tool support" ICSE 1997. URL: http://ieeexplore.ieee.org/ielx3/4837/13372/00610430.pdf?tp=&arnumber=610430&isnumber=13372.

[13] Parnas D., Madey J., Functional Documents for Computer Programs. Science of Computer Programming, Vol. 25, No. 1, 1995.

[14] Galloway, A., Iwu, F., McDermid, J. A., Toyn, I., On the Formal Development of Safety Critical Software, In: Verified Software: Theories, Tools, Experiments, First IFIP TC 2/WG 2.3 Conference, VSTTE 2005, Zurich, Switzerland, October 10-13, 2005, Meyer, B., Woodcock, J. C. P. (eds.) pp 362-373.

[15] SysML, see: http://www.omgsysml.org/ (last accessed August 1st 2013).

[16] IEEE Standard 1012-2012, "IEEE standard for system and software verification and validation," March 29, 2012.

[17] ISO - International Organization for Standardization, BS ISO 26262-2: 2011, Road Vehicles – functional safety, Part 2: Management of functional safety.

[18] ISO - International Organization for Standardization, BS ISO 26262-3: 2011, Road Vehicles – functional safety, Part 3: Concept phase.

[19] ISO - International Organization for Standardization, BS ISO 26262-4: 2011, Road Vehicles – functional safety, Part 4: Product development at the system level.

[20] Despotou G., Alexander R., Kelly T.P., Addressing Challenges of Hazard Analysis in Systems of Systems, 2009, In proceedings of the 3rd Annual IEEE International Systems Conference (SysConf '09), Vancouver Canada, 23-26 March 2009.

[21] AADL, see; http://www.aadl.info/aadl/currentsite/ (last accessed August 1st 2013).

[22] Mader, R., Grießnig, G., Leitner, A., Kreiner, C., Bourrouilh, Q., Armengaud, E., Steger, C., Weiβ, R., " A Computer-Aided Approach to Preliminary Hazard Analysis for Embedded Systems," 18th IEEE International Conference and Workshops on Engineering of Computer-Based Systems, 2011.

[23] OpenMETA tool suite. URL: http://www.army-technology.com/news/newsvanderbilt-university-support-meta-tools-maturation-darpa-avm-programme.

[24] Miller, S.P., Tribble, A.,C., Extending the Four Variable Model to Bridge the System-Software Gap, in Proc. 20th Digital Avionics System Conference, DSAC01, Daytona Beach Florida, October 2001.

[25] Simpson H.R., The MASCOT method. Software Engineering Journal, 1(3):103–120, March 1986.

[26] International Electrotechnical Commission, "Programmable controllers – Part 3: Programming languages" IEC 61131-3, ed3.0, 2013.

[27] Barnes J.G.P., High Integrity Software: The SPARK Approach to Safety and Security, Addison Wesley, 2003.

[28] SPARK Pro toolset, see: https://www.adacore.com/sparkpro/ (last accessed August 2nd 2013).

[29] MISRA C, see: http://www.misra.org.uk/MISRAHome/MISRAC2012/tabid/196/Default.aspx (last accessed August 1st 2013).

[30] LDRA MISRA C toolset, see: http://www.ldra.com/en/solutions/by-standard-adherence/misra (last accessed August 2nd 2013).

[31] Conmy P.M., Pygott C., Bate I.J., VHDL Guidance for Safe and Certifiable FPGA Design, IET System Safety Conference, October 2010.

[32] INL/EXT-14-33129, "Light water reactor sustainability program advanced instrumentation, information, and control technologies Digital Technology Business Case Methodology Guide" US Dept. of Energy Office of Nuclear Energy, September 2014.

[33] http://cps-vo.org/group/scc .

[34] Douglas A. Bodner, et al, "Designing an Experiential Learning Environment for Logistics and Systems Engineering."

[35] Douglas A. Bodner, et al, "Simulation-Based Decision Support for Systems Engineering Experience Acceleration."

[36] Jo Ann Lane and Richard Turner, "Improving Development Visibility and Flow in Large Operational Organizations."

[37] John Hatcliff, et al, “Certifiably safe software-dependent systems: challenges and directions” in Proceedings of the Future of Software Engineering FOSE 2014, pages 182-200, ISBN: 978-1-4503-2865-4.

[38] ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.

[39] ISO/IEC 15939:2007(E) Systems and software engineering – Measurement process.

[40] ISO/IEC 25000: 2005(E) Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE.

[41] ISO/IEC/IEEE 24765:2010, Systems and software engineering — Vocabulary.

[42] ISO/IEC 25030:2007, Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements.

[43] ISO/IEC TR 25060, Systems and software engineering — Systems and software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability: General framework for usability-related information.

[44] Roberts, F. Measurement Theory with Applications to Decision Making, Utility, and the Social Sciences, Addison-Wesley, 1979.

[45] ISO/IEC 15288:2008 (IEEE STD 15288-2008), Systems and software engineering — System life cycle processes.

[46] ISO/IEC 17000 Conformity Assessment—Vocabulary and general principles, 2004-11-01.

[47] NRC, “Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition – Instrumentation and Controls,” NUREG-0800, Chapter 7 ADAMS Accession No. ML070550074.

13. Glossary

This glossary also applies to the appendices.

For definitions of terms not defined below, see Appendix A in NRC RIL-1101 [5].

For terms not defined in [5], please see [38]-[43].

Attribute (of quality**)**
Inherent property or characteristic of a system or its element that can be distinguished quantitatively or qualitatively. (Adapted from 2.2 in [39]).

Notes:
1. The means of distinction may be manual or automated.
2. Also see "Quality measure" and "Scale" [44].
3. ISO 9000 distinguishes two types of attributes: a permanent characteristic existing inherently in something; and an assigned characteristic of a product, process or system (e.g. the price of a product, the owner of a product). The assigned characteristic is not an inherent quality characteristic of that product, process or system. An inherent property may be:
    3.1. Functional property: It determines what the software (in general, item) is able to do.
    3.2. Quality property: It determines how well the software (in general, item) performs. It is defined as "measurable component of quality" in [40] §4.3.9.
4. **"(Software) quality characteristic":** Category of software quality attributes that bears on software quality (§4.4.10 in [40]). Note: A quality characteristic can be refined into multiple levels of sub-characteristics and finally into software quality attributes.
    4.1. Other authorities have not distinguished across meanings of these terms {property; characteristic; attribute} in the same manner. For this specific usage context, a quality model should clarify the meanings and inter-relationships.

Certification
Third-party attestation related to products, processes, systems or persons (§5.5 in [46]).

Notes:
1. Certification of a management system is sometimes also called registration.
2. Certification is applicable to all objects of conformity assessment except for conformity assessment bodies themselves, to which accreditation is applicable.
3. Attestation: issue of a statement, based on a decision following review, that fulfilment of specified requirements has been demonstrated (§5.2 in [46]).
4. Review: verification of the suitability, adequacy and effectiveness of selection and determination activities, and the results of these activities, with regard to fulfilment of specified requirements by an object of conformity assessment (§5.1 in [46]).
5. Specified requirement: Need or expectation that is stated (§3.1 in [46]).
6. Conformity assessment: demonstration that specified requirements relating to a product, process, system, person or body are fulfilled (§2.1 in [46]).
7. Product: result of a process (§3.3 in [46]).

8. Third-party conformity assessment activity: conformity assessment activity that is performed by a person or body that is independent of the person or organization that provides the object, and of user interests in that object (§2.4 in [46]).
    8.1. Criteria for the independence of conformity assessment bodies and accreditation bodies are provided in the International Standards and Guides applicable to their activities and listed in the bibliography of [46].
    8.2. The first-, second- and third-party descriptors used to characterize conformity assessment activities with respect to a given object are not to be confused with the legal identification of the relevant parties to a contract.
9. Accreditation: third-party attestation related to a conformity assessment body conveying formal demonstration of its competence to carry out specific conformity assessment tasks (§5.6 in [46]).
10. For other supporting information, see [46].

Quality
Capability of product[23] to satisfy stated and implied needs when used under specified conditions. (Adapted from 4.51 in [40])

Notes

1. This definition differs from the ISO 9000:2000 quality definition; it refers to the satisfaction of stated and implied needs, while the ISO 9000 quality definition refers to the satisfaction of requirements.
2. The term “implied needs” means “needs that may not have been stated explicitly (e.g., a need that is considered to be evident or obvious; a need implied by another stated need).”
3. **Quality model**: Defined set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality. (Adapted from 4.44 in [40])
4. **Quality measure**: An attribute of quality to which a value is assigned. Also see scale.
5. **Quality in use**: Capability of the product to enable specific users to achieve specific goals in specific contexts of use. The expression “in use” refers to the expectations of the end user.
    5.1. Actual quality in use may be different from quality in use measured in a test environment earlier in the product lifecycle, because the actual needs of users may not be the same as those reflected in the test cases or in the requirements specifications.
    5.2. Quality in use requirements contribute to identification and definition of external software quality requirements.
    5.3. Example of quality in use: Safety (freedom from harm).
6. **Measurement of external quality** refers to measurement from an external view of the product, where targets are derived from the expected “quality in use” and are used for technical verification and validation. For example, external software quality would be measured in terms of its capability to enable the behavior of the system to satisfy its quality in use requirements, such as safety.

[23] Or service

7. Measurement of internal quality refers to measurements during the developmental phases of the product lifecycle. Targets are derived from targets for measurement of external quality.
8. **(Software) Quality requirement** is defined **as** requirement that a software (in general, item) quality attribute be present in software (in general, the item).

System element
A discrete constituent of a system that can be implemented to fulfill specified requirements (adapted from note under §4.32 in [45]).

Notes:

1. The definition §4.32 in [45] is circular; it defines element using the word elements. The term "constituent" is substituted for the word "part" used in the note under §4.32 in [45]. Reason: Avoid confusion with other meanings of "part" in the context of software.
2. The word "discrete" implies that the constituent has a distinct boundary (i.e., interface) with its environment, and an intrinsic, immutable, unique identity.
3. Examples:
   3.1. Hardware element
   3.2. Software element
   3.3. Human element
   3.4. Data element
   3.5. Process
   3.6. Procedure (e.g., operating instructions)
4. An element may have other elements in it (e.g., a subsystem).
5. A system may itself be an element of a larger system.
6. The term "item" is used to refer to any of the above.

Scale (for a quality measure)
Ordered set of values, continuous or discrete, or a set of categories to which an attribute is mapped. (Adapted from 2.35 in [39])

Notes

1. The type of scale depends on the nature of the relationship between values on the scale [39].
2. Four types[24] of scale are commonly defined [39]:
   2.1. Nominal: The measurement values are categorical
   2.2. Ordinal: The measurement values are rankings
   2.3. Interval: The measurement values are equi-spaced
   2.4. Ratio: The measurement values are equi-spaced, where the value 0 (zero) is not mapped to any attribute.
3. The valid value space is predetermined.
4. The mapping of the magnitude of the measured attribute to a value on the scale is predetermined.

[24] For other types of scales, see [44].

State-of-the-Art
In an engineering field, “state of the art”, is a collection of the most advanced publicly available, scientifically validated knowledge, the utility of which is demonstrated through realization of a novel product or process, beneficial to society.

State-of-the-practice
In an engineering field, “state-of-the-practice”, is the most advanced documented knowledge in that field, in actual use on a regular basis.

Usability
Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use (4.2.4 in [38]).

14. Acronyms & Abbreviations

Symbol	Expansion	Informal supplemental information
AADL	Architecture Analysis and Design Language	See SAE Standard AS5506: http://standards.sae.org/as5506/
aka	Also known as	
AVSI	Aerospace Vehicle Systems Institute	http://www.avsi.aero/
CPS	Cyber physical system	
DOD	U.S. Department of Defense	
DOD/ASD(R&E)	U.S. Department of Defense Research & Engineering	ASD: Assistant Secretary of Defense. Systems engineering: ODASD(SE): http://www.acq.osd.mil/se/
DOE-NE	U.S. Department of Energy – Office of Nuclear Energy	
EPRI	Electric Power Research Institute	
FAA	Federal Aviation Administration	
FDA	U.S. Food and Drug Administration	
FPGA	Field programmable gate array	
FY	Fiscal Year	
GE	General Electric	
HA	Hazard Analysis	
I&C	Instrumentation & Control	
INPO	Institute of Nuclear Power Operations	

MIT	Massachusetts Institute of Technology	
NASA	National Aeronautics and Space Administration	
NIST	National Institute of Standards and Technology	Relevant R&D at NIST: CPS: http://www.nist.gov/cps/index.cfm Measurement: http://www.nist.gov/pml/div688/timing-031915.cfm ; http://www.nist.gov/itl/tis/mma.cfm
NPP	Nuclear Power Plant	
NRC	U.S. Nuclear Regulatory Commission	
NRL	U.S. Naval Research Laboratory	
OMB	Office of Management and Budget	
OSTP	Office of Science and Technology Policy	
R&D	Research and Development	
RAI	Request for Additional Information	At the NRC, when the reviewer of an SAR cannot find reasonable assurance of adequate safety because of inadequate information, the reviewer may formulate an RAI.
SEI	Software Engineering Institute	
SAR	Safety analysis report	For example, an NPP operator includes in it the analysis to demonstrate that the system is safe.
SAVI	The System Architecture Virtual Integration Program	http://savi.avsi.aero/
SER	Safety evaluation report	At the NRC, a product of NRC safety review for licensing.
V&V	Verification and Validation	

A. Concepts, Vocabulary, Terminology[25]

Examples of terms not used with consistent meanings in different usages

Analysis

Architecture

Assure
- Assurance
  - Safety Assurance
  - Security Assurance
  - Software Assurance
  - ….

Attribute (of quality)

Cause
- Systemic cause

Common cause

Common mode

Dependency
- Independent

Diverse
- **Diversity**[26]

Defect

Mistake

Error

Environment

Evidence

Failure
- Failure Analysis
- Common cause failure
- Common mode failure

Fault
- Faulty
- Fault Analysis
- Fault Mode
- Fault Modes and Effects Analysis (FMEA)
- Fault Tree Analysis (FTA)

Hazard
- Contribute
- Contributory hazard
- Safety hazard[27]

---

[25] Usage in this document is as defined in RIL-1101 [5].
[26] What do we need to know? What analyses support reasoning about diversity at the conceptual phase?
[27] 10CFR50 uses this term, but does not define “safety” and “hazard.”

- Security hazard
- Internal hazard (contrast with external hazard)
- Hazard Analysis
- Hazard Identification

Indicate

Intended

Process

Product

Quality
- Quality in use
- External quality
- Internal quality

Requirement
- Functional requirement (contrast with “Design”)
- Quality requirement

Reason
- Reasoning
- Reasonable

Reliability (symbol : $R(t_1, t_2)$)

Risk
- Risk estimation
- Risk evaluation
- Risk analysis
- Risk management
- Risk-informed; risk-informing

Safety
- Safety-significant

Scale (for a quality measure)

Software

State
- State space
- Hazard space
- Event
- State Transition

System
- (System) Element

Systemic

Validation

Verification

B. Current state, next state, and envisioned goal state

Referring to Figure 3 and Figure 4, the broad-brush characterizations of the current state, next state (capability to engineer concept), and an envisioned goal state. The characterization is limited to the purpose of identifying similarities with other high-consequence application domains (e.g., safety-critical medical devices) and thus, help factor out the common core capabilities, as in Figure 1.

B.1 Broad-brush characterization of current State (S)

While Figure 3 depicts a template to evolve capability in iterative cycles of growth, Figure 4 illustrates this concept in the context of NPP digital safety systems, as explained below. Similar situations exist in other critical application sectors.

1. **Current regulatory environment**: An electricity producer, intending to create a new NPP, applies to the regulator (e.g., NRC) for a license or certification of a reusable design under 10 CFR Part 52. An electricity producer, intending to modify an existing NPP, applies to the regulator for an amendment to its existing license under 10 CFR Part 50. The applicant provides the regulator with a safety analysis report (SAR), which the regulator reviews, producing a safety evaluation report (SER).

    1.1. Whereas two years is the estimated duration between the initial submittal of an SAR and the completion of an SER, it has taken over five years in some cases. All involved parties consider it an unacceptable state.

    1.2. When the regulator does not see sufficient information for a safety determination, it initiates a request for additional information (RAI). Often, the applicant's response does not resolve the issue.
        1.2.1. Each round adds significant delay.
        1.2.2. Typically, many rounds of RAIs and applicant's responses occur, adding delays.
        1.2.3. All involved parties consider it an unacceptable state.

    1.3. The regulator encourages pre-application informal dialogs to provide clarifications and avoid post-application delays. However, in some cases, even after 40 meetings, involved parties are not confident about the acceptability of their intended application.

    1.4. Both, the regulator and the applicant recognize that the current state is not satisfactory and desire improvements.

    1.5. Industry perceives regulatory uncertainty as a significant business risk.

2. **Safety Analysis Report (SAR) – current state**:

    2.1. The SAR is organized to show clause-by-clause compliance with regulatory guidance (RG) and industry consensus standards referenced therein, typically following [47].

    2.2. The SAR is a voluminous collection of narratives and graphics.

    2.3. The SAR does not support mechanized reasoning for review or confirmatory analysis.

    2.4. Evaluation of the SAR is manual.

    2.5. It is difficult to discover hazards contributed through unwanted/unintended interactions.

        2.5.1. This hazard space is increasing with increasing electronic intrusions and infiltrations being experienced in NPPs.

3. **Regulatory Guidance** (RG)**, including standards – current state**:

3.1. It is oriented towards process characteristics rather than product (safety system).

3.2. It is not consistently understood by all parties (e.g.: regulator; applicant; vendor).

3.3. Revisions of standards and guides are not keeping up with technology changes.

3.4. The lag leaves much in judgment space.

3.5. The contribution relationship of a guidance item to a safety goal is not explicit.
- 3.5.1. Various guidance items have overlaps and inconsistencies.
- 3.5.2. Known gaps recognized above are covered by judgment.
- 3.5.3. Novelties in configurations and technologies may create unknown gaps.

4. **Information supporting SAR – current state**:

4.1. Plans (safety; HA; V&V) are not developed early enough with information for use in safety evaluation.

4.2. For new reactors, sufficient design information is not available early enough for efficient, effective safety evaluation.

5. Different regions in the world follow different legacy standards, obstructing international harmonization and sharing of experiential knowledge.

6. Standards referenced in RG are premised on adequate reader (user) competence.

6.1. Their interpretation differs across practitioners with different backgrounds.

6.2. The contribution of an item in a standard to the safety goal is not explicit, i.e., in many cases, its safety significance is not easy to determine, leaving much to expert judgment.

7. **Competence – current state**:

7.1. No explicit competence requirements or criteria, suitable for consistent evaluation.

7.2. No certification process to ensure that competence is adequate for the assigned task.

7.3. These inadequacies cascade down the integration hierarchy from NPP-wide I&C systems to digital safety systems to digital platforms and kernels.

7.4. These inadequacies cascade down the supply chain.

7.5. The applicant (the electricity producer), often, does not have intramural self-sufficiency to perform the system engineering and integration.

- 7.5.1. The applicant may use technical services from third parties.
- 7.5.2. Vendors offer turn-key systems engineering but may not have the same perspective and understanding of the environment as the applicant.
- 7.5.3. Vendors may not have the same interests as the applicant.
- 7.5.4. This skill-dependence exposes the applicant to significant business risk.

8. **Hazard analysis** (HA) **– current state**:

8.1. The HA-part of the NPP-level safety analysis is based on Event Tree Analysis (ETA) and Fault Tree Analysis (FTA) techniques[28] which may not be adequate in the presence of interactions and feedback paths, enabled when digital systems are networked.

[28] Computerized tool assistance (tree-based) is being used.

8.2. The digital safety system HA is based on failure modes and effects analysis (FMEA) techniques which may not be adequate in the presence of interactions and feedback paths.

8.3. NPP-level HA is not integrated well with digital safety system HA.

8.4. HA is not well integrated with mainstream engineering to develop the system. Over time, inconsistencies creep in.

8.5. The NPP community perceives that newer hazard analysis techniques are not mature enough.

9. **Requirements – current state**:

9.1. Mostly text in natural language:
9.1.1. Lack standardized vocabulary.
9.1.2. Ambiguous. Different users may interpret different meanings.
9.1.3. Not amenable to reasoning about completeness, consistency, and correctness.

9.2. Not amenable to mechanized reasoning.

9.3. Lack explicit relationships across requirements.

9.4. Lack well-defined quality attributes.

10. **Design (architecture level) – current state**:

10.1. Not explicitly and systematically driven by quality attributes.

10.2. Architecture is documented through block diagrams and narratives:
10.2.1. Not suitable for mathematical analysis or consistent reasoning.

11. **Implementation and integrated testing – current state**:

11.1. Controversy over adequacy of testing.

11.2. Limited by inadequacies in requirements.

B.2 Broad-brush characterization of envisioned Goal State (G)

Efficiency and effectiveness goals motivate the vision to make “Assurance” a non-issue in terms of cost, delay, conflict and uncertainty. Treating this vision as a virtual goal helps find paths from the current state in a manner that drives out cost, delay, conflict and uncertainty in successive stages.

Figure 6 depicts an envisioned assurance process intended to support this goal.

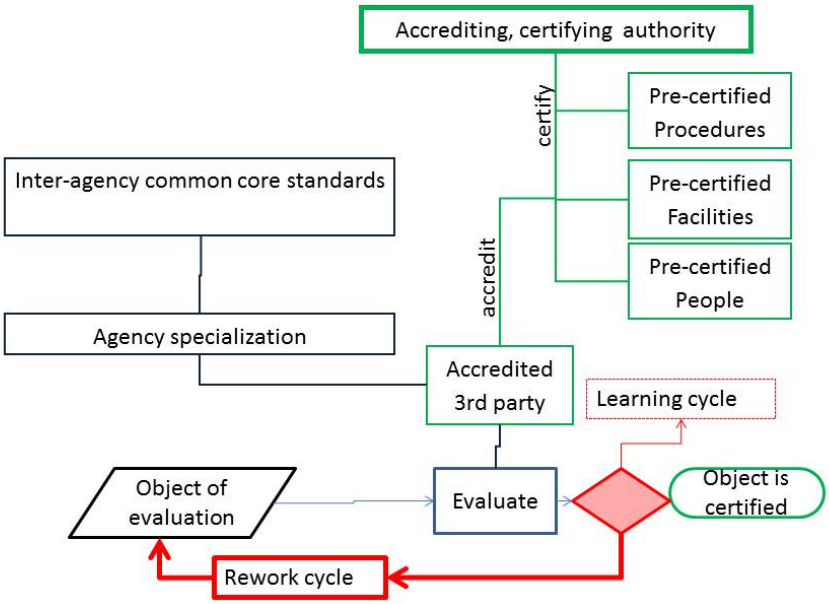

Figure 6: Envisioned assurance process

Figure 7 depicts envisioned precertification activities to support the system assurance process.

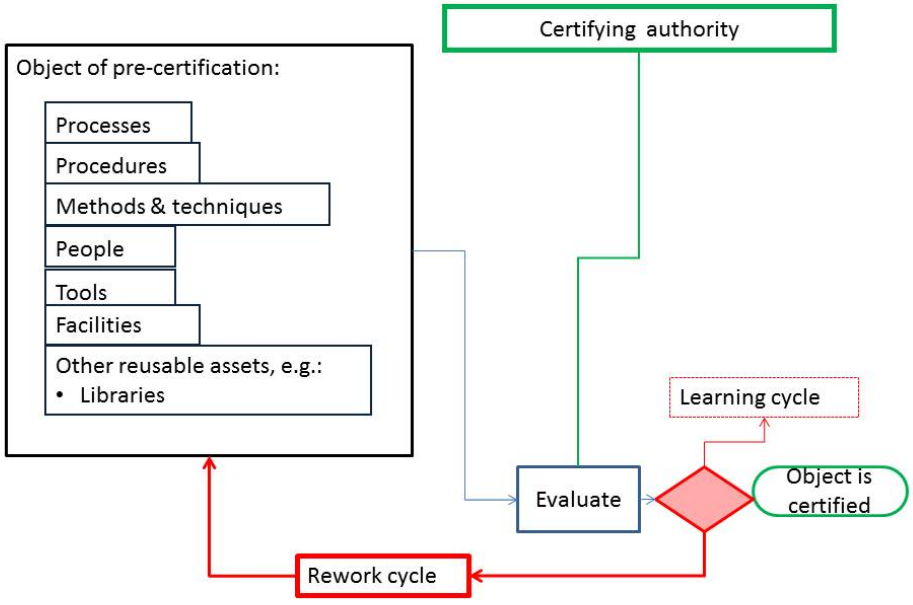

Figure 7: Envisioned precertification process

B.2.1 Other characteristics of envisioned goal state (G)

1. Certification of an NPP digital safety system becomes as routine as certification of an NPP safety valve. It is no more a source of regulatory uncertainty or conflict.
2. Assurance does not add cost or delay, but reduces system development lifecycle costs and duration in comparison to current practice. Examples:
   - 2.1. Rework to correct deficiencies is eliminated.
     - 2.1.1. Plans, processes, and procedures prevent defects.
     - 2.1.2. Prevention is built in from the beginning of the development lifecycle.
   - 2.2. Effort exclusively to perform verification activities, e.g., testing logic, is eliminated.
   - 2.3. A component in a system can be replaced safely[29] without requiring re-certification of the whole system.
   - 2.4. Pre-certified components are integrated into a system without requiring re-verification:
     - 2.4.1. No hazardous condition can be introduced.
     - 2.4.2. Evidence is provably compose-able.
     - 2.4.3. Assurance case is provably compose-able[30].
   - 2.5. Pre-certified components are reused with provable safety[31] without requiring recertification:
     - 2.5.1. No hazardous condition can be introduced.
     - 2.5.2. Evidence is provably compose-able, including quality properties.
   - 2.6. Domain specific pre-certified assets[32] are available, with unambiguous specifications including their conditions of use, such that any application satisfying the conditions can plug them in:
     - 2.6.1. Pre-certified, provably compose-able assets are available in the marketplace to any applicant or any one in an applicant's supply chain.
     - 2.6.2. Third-party certifiers[33] are available in the marketplace for various levels of integration.
   - 2.7. Asset library facilitates correct reuse.
   - 2.8. Routine activities are automated. Examples:
     - 2.8.1. Supporting evidence from verification is auto-generated and formal.
       - 2.8.1.1. At all levels of integration.
       - 2.8.1.2. For all levels of abstraction-concretion (development phases).
3. Industry is able to use the capability to advantage in other economically important digital systems that are not safety critical.
4. Standards: There is a common core (shared across different application domains), supplemented with domain specific specializations as needed.
   - 4.1. There is a clear and smooth process to propose changes to the criteria and methods.

---

[29] Enablers: Compose-ability; Compositionality.
[30] Assurance case framework facilitates reasoning-based integration of verification evidence.
[31] Enablers: Compose-ability; Compositionality.
[32] Examples: Components; Models; Plans; Processes; Procedures; Chains of reasoning.
[33] Example: Accredited by the regulator.

5. Lifetime maintenance of an assurance framework, including standards and a family of tools and language subsets and test-beds. It could help develop third-party certification capability, including evaluation, assessment of systems, their development environments and elements in these environments.
6. Hazard analysis is integrated with requirements engineering:
   6.1. Derived from the safety goal rather than compliance with regulatory guidance.
   6.2. Quality[34] requirements (requirements for quality attributes) are correctly incorporated.
   6.3. Assurability is a constraint, specified, flowed-down and satisfied from the beginning
   6.4. Recognition that requirements evolve throughout the lifecycle.
   6.5. Requirements and specifications are semantically related across levels of abstraction, with automation support for search, navigation and reasoning.

B.2.2 Stakeholder perspectives and roles in envisioned Goal State

B.2.2.1 Applicant

1. Certification of a digital safety system is as routine as obtaining a certified safety valve.
   1.1. No contention with regulator.
   1.2. No regulatory uncertainty.
   1.3. The need for pre-application dialog goes away[35].
2. Focus of cognitive work shifts to:
   2.1. Perform better hazard analysis and requirements engineering.
   2.2. Validate the requirements.
   2.3. Evaluate the supply chain.
   2.4. Evaluate the facilities used in creation of the safety system.
   2.5. Realize systems that are inherently invulnerable.

B.2.2.2 Regulator

1. No need to do any direct evaluation of an applicant's safety analysis.
2. Review third-party evaluation results & confirm.
3. Check that evaluation has come from an accredited third party.
4. Evaluate results to learn from the process and identify future improvements needed in the regulatory process, e.g. criteria.
5. Focus of cognitive work shifts to evaluating whether facilities, tools, assets and supporting resources support the assurance objective.

B.2.2.3 Third-party Certifier

1. Apply regulator-approved domain-specific criteria to evaluate applications.
2. Utilize pre-certified resources[36] in the process.
3. The process is economically self-sustaining.

B.2.2.4 Community developing standards and guides

1. Develop agreed upon criteria & goals (general + domain specific).

---

[34] Commonly known as non-functional
[35] The privilege does not go away.
[36] Eventually to be fee based, but may require underwriting by the government

2. Evolve criteria in response to feedback and experience.

B.2.2.5 Tool Vendor

1. Self-sustaining[37] economics business model, including support for evolution.
2. Consensus-based approach/method to evolve supporting infrastructure.

B.2.2.6 ***Supply Chain***

Practice a process that leverages the common core and domain-specific extensions.

B.3 Broad-brush characterization of State S + 1: Capability to engineer concept

The capability to evolve and engineer a concept correctly is crucial, because most cost and issues are embedded in the concept, even though the effects may surface later in the development lifecycle. This capability affects the pre-application activities of the applicant, including pre-application meetings with the regulator.

B.3.1 Applicant's perspective

1. Substantially reduce application preparation effort and duration.
2. Eliminate regulatory uncertainty later in the lifecycle.
3. Leverage reusable assets and automation to develop concept.
4. Perform preliminary hazard analysis (PHA) at the concept level with tool support.
5. Transform concept phase work products to subsequent phase work products efficiently.
6. Exercise a range of concepts and corresponding hazard analysis:
   - 6.1. Perform comparative evaluation;
   - 6.2. Have productive, efficient pre-application discussions with the regulator;
     - 6.2.1. Pre-application discussions include the environment of the system.
   - 6.3. Demonstrate how the applicant will satisfy the safety goal, diversity, defense in depth and redundancy requirements.
7. Build confidence to commit engineering resources for the subsequent engineering.

B.3.2 Capability to be developed to support concept phase engineering

The following examples provide a broad-brush characterization.

1. Ability to describe interactions between system (e.g. a top-level function model of a reactor protection system and its interactions with its environment[38]. {functional requirements; quality attributes; assumptions}:
   - 1.1. Language support.
   - 1.2. Asset library: Support modeling of dependencies of various types. (Extensible).
2. Control system view: It may not interact, but one has to be aware of it.
3. Sufficient information to do hazard analysis at conceptual level:
   - 3.1. Identify hazard.

[37] Creation may require infusion of capital.
[38] Hazard scenario: interactions between the system and element in its environment

3.2. Specify hazard control.

4. Support for engineering and evaluating concepts of separation and partitioning on FPGA targeted architecture.
5. Proven partitioning or containment techniques to prevent unnecessary dependencies.

B.3.3 Regulator perspective

1. If applicant requests a pre-application meeting, applicant brings information of the quality and completeness needed to support efficient, effective discussion.
2. Increased likelihood of a good application.
3. Reduced effort.
4. Early awareness of novelties (conditions not encountered before) in the application.

B.3.4 Some activities to transition from one state to the next

1. Set up the learning cycle for iterative, incremental growth:

   1.1. Add to the library of reusable assets (building blocks; model library elements).

   1.2. Extend the domain model.

   1.3. Create other learning exercises.

   1.4. Exercise.
2. Have learner apply the acquired resources (including knowledge and skills) to selected subsets of information, e.g., problems[39] encountered in the past.
3. Extend the model and exercises to deepen understanding of the technology and resources supporting HA. For example sources of ideas, see [5].

B.4 Some characteristics of State S+2

1. Capability to develop a requirements model[40] of the system, including constraints, inter-relationships, esp. dependencies, and assumptions; capability to perform HA on it.
2. Knowledge-base[41] of contributors to hazards through deficiencies in requirements.
3. Specifications of constraints to avoid these hazards.

B.5 Some characteristics of State S+3

1. Capability to develop a model of the system at the next level decomposition, e.g., the four divisions and the voting logic.
2. Knowledge-base of contributors to hazards.
3. Specifications of system constraints to avoid these hazards.

[39] Example: Sharing of neutron-flux density data sensed in each division with other divisions
[40] Include quality model, which includes quality (or quality of service) attributes and sub-characteristics.
[41] Examples: Contribution paths through causality and other dependencies.

C. Competence

Inadequate replenishment of requisite competence: The CPS engineering workforce is changing and so is the environment from which the workforce is being replenished. With the decline in the U.S. manufacturing industry, there has been corresponding decline in its industrial automation development base. Education and training concerning software are driven more by consumer products and information technology (IT) industries than by high-consequence automation. “Development of CPS systems for the highest level of safety” is a very small, niche in the market.

Competence is a critical factor; competence to perform HA of an NPP digital safety system includes a complement of the following - not necessarily in one person (in [5] Appendix C):

1. Proven self-learning[42] ability, assimilating needed new knowledge in a scientifically sound framework:
   - 1.1. Education equivalent to a master’s degree level knowledge of safety critical industrial automation systems engineering;
   - 1.2. Ability to recognize the knowledge needed and limitations of one’s knowledge.
   - 1.3. Ability to fill one’s knowledge gaps through self-study, supplemental training, and consultation with experts.
2. Reasoning capability (see Figure 5):
   - 2.1. Objectivity. (Also see item 9).
   - 2.2. Ability to abstract and generalize from one context and apply to another.
   - 2.3. Ability to recognize fallacies in some chain of reasoning.
3. Continuing, self-driven update of professional knowledge; examples:
   - 3.1. Application domain: How an NPP works
     - 3.1.1. Energy conversion from fuel to power on the grid.
     - 3.1.2. Heat exchange.
     - 3.1.3. Critical functional elements, processes and process state variables in an NPP.
     - 3.1.4. Their inter-dependencies.
     - 3.1.5. Associated (contributory) hazards
     - 3.1.6. Study of operating experience (event reports; root cause analysis reports).
   - 3.2. Industrial automation domain:
     - 3.2.1. Physical elements: sensing; actuation; power; communication; storage.
     - 3.2.2. Logic elements: computation; control logic; communication; software/firmware.
     - 3.2.3. Associated (contributory) hazards.
     - 3.2.4. Study of operating experience (event reports; root cause analysis reports).
   - 3.3. Science and engineering of distributed systems, including:
     - 3.3.1. Systems engineering
       - 3.3.1.1. Embedded systems engineering.
       - 3.3.1.2. Specification and analysis for properties such as safety and security.
     - 3.3.2. Integration.
       - 3.3.2.1. Integrated effects of interactions (e.g., across software-hardware).
     - 3.3.3. Architecture.
     - 3.3.4. Computation.

[42] When the object being analyzed entails some characteristic, which the analyst has not encountered in past experience, as is often the case in digital safety systems, corresponding learning is needed.

3.3.5. Communication.
3.3.6. Hardware engineering.
3.3.7. Software engineering.
3.3.8. Specification and analysis for timing and order of execution.

3.4. Hazard and safety analysis and assurance methods and techniques for such systems at each level of integration and abstraction-concretion.

4. Experience in analysis of systems similar in criticality, functionality, and configuration:
   4.1. Good performance under the guidance of an expert in hazard analysis.
   4.2. Good performance independently.
5. Strongly safety conscious. See in [5] Appendix F.1 and F.3.
6. Communication skills in group activities (see Appendix F.4 in [5]) – examples:
   6.1. Ability to communicate effectively, objectively with stakeholders.
      6.1.1. Succinctness.
   6.2. Ability to listen actively for understanding and learning from others.
   6.3. Ability to elicit information needed.
   6.4. Ability to explain one's reasoning to others (see Sections C.3.3 and C.4 in Appendix C of [5]).
   6.5. Ability to express and explain to others insights from deep knowledge.
   6.6. Ability to develop collective communicative competence. See Appendix F.4.3 in [5].
7. Other interpersonal skills and characteristics, supportive of teamwork (see Appendix F.4 in [5]), e.g.:
   7.1. Willingness to recognize and accept weakness in own reasoning.
   7.2. Willingness to explain own reasoning (clearly; succinctly) in the face of opposition.
   7.3. Assistive rather than competitive behavior.
   7.4. Ability to evoke minority viewpoints (concerns or reservations).
   7.5. Ability to understand other team members' reference-frames.
   7.6. Ability to assimilate differences, neutralizing biases.
   7.7. Ability to converge[43] towards objectivity (see Sections C.3.3 and C.4 in Appendix C of [5]). See "collective mindfulness" in Appendix F of [5].
   7.8. Other constructive group interaction skills.
8. The complement of competence in the HA team includes breadth and depth.
   8.1. Depth: Individuals having mastery over the respective engineering disciplines, technologies, products or components, and processes, involved in each phase of the system development lifecycle (possibly involving phase-wise changes in team-membership) and respective dependencies.
      8.1.1. Knowledge of respective operating experience (what can go wrong).
      8.1.2. Track record of learning from it (how to prevent what went wrong).

[43] Through ability to articulate premises & qualifications of claims; how those derive from given contexts.

8.2. Breadth[44]: Individuals are able to understand how their respective roles fit into the overall HA, including the associated inter-dependencies.
  8.2.1. Knowledge of the environment[45] of the safety system and its development.
  8.2.2. Experience in analysis of hazard groups such as those identified in RIL-1101.
  8.2.3. Experience in deriving requirements.
  8.2.4. Experience in deriving to avoid or eliminate contributory hazards.
  8.2.5. Experience commensurate to the functionality and configuration of the system.

9. The HA-team has cultural diversity[46] - supportive of safety.

D. Quality requirements for the road-mapped R&D

These quality requirements apply to the road-mapped R&D proposed for inter-agency coordination.

Notes:

1. In the interest of brevity and to avoid duplication, links are used extensively. Uncommon terms are explained in the Glossary and Abbreviations & Acronyms.

D.1 Properties to be satisfied

The relationships of these properties to NRC's strategic plan are identified through links to the strategic plan items.

While the properties are characterized generally in [38], NPP domain-significant dimensions and criteria are itemized under each property below.

Notes

1. In this context, these properties are not independent dimensions, but have significant inter-relationships.
2. Their combination drives the design of the sequence of iterative, evolutionary R&D cycles.
3. Design of the sequence should demonstrate feasibility of reaching the envisioned goal state.
4. Design of the immediate next cycle should be specific enough to implement in a work plan with specific deliverables, duration and effort.
5. Design of a few cycles following the immediate next cycle should be specific enough to support resource forecasting, reservation of critical resources, and other related planning.

D.2 Scalability

1. Ability to adjust the scope to the resources (e.g., funding[47]) available.
2. Rate of scale-up when additional[48] resources become available.

---

[44] Provide continuity to the HA-team across lifecycle phases.
[45] Also see Section 10.
[46] See reference-frames in item 7.5; examples: belief systems, values, thought processes, paradigms, customs, conventions, language.
[47] Example: Funding for baseline planning may be $200,000 per year for NRC-external resources.
[48] Example: Although confirmation date & amount are not certain, forecasting is feasible.

D.3 Effectiveness

Effectiveness in transforming the state of safety assurance capability is characterized as follows.

1. Ability to operationalize the new capability (e.g., knowledge; methods & techniques; tools; and such safety certified resources) - sooner the better.
   1.1. Demonstrated improvement in effectiveness of and efficiency of pre-application meetings between the regulator and the applicant, through the informal discussion of concept alternatives under the applicant's consideration.
   1.2. Demonstrated improvement in effectiveness of application-reviews.
   1.3. Demonstrated improvement in efficiency of application-reviews.
2. Integration and utilization of state-of-the-art knowledge from the best known sources.

D.4 Efficiency

Efficiency of transforming the state of the assurance capability is characterized as follows.

1. "Distance" from the fastest rate feasible - the lesser the better (more efficient).
2. Resource-usage corresponding to "distance" from the fastest rate feasible - the lesser the better.
3. Absence of waste (e.g., effort that could not have led to the desired outcome).
4. Develop and leverage reusable assets, e.g., domain model.
   4.1. For example starters, see Annex to this appendix.
5. Leverage existing knowledge - see item 2 under Effectiveness:
   5.1. Avoid "reinventing the wheel."
      5.1.1. Reduced effort to achieve the required outcome.
      5.1.2. Shorter duration to achieve the required outcome.

D.5 Satisfaction of user[49] community

1. Motivation. Example of a scale to measure motivation: {Excited, inspired, and enthused to learn more; Inquisitive; Open to change; Reluctant to change; Closed to change; Scared; Disgusted; Repelled}.

   Note: For the purpose of designing the sequence of iterative evolutionary cycles, the scope may be limited to individuals who express an interest in personal development and commitment, e.g., through an "expression-of-interest" questionnaire.

2. Confidence in applying the new capability (e.g., knowledge; methods & techniques; tools).

D.6 Learnability

1. Learning rate of an individual in relation to his/her potential.
   1.1. Measurability of progress [39] [44].

   Note: The sequence of iterative evolutionary cycles may be based on self[50]-declared potential, e.g., through a self-declaration questionnaire.

2. Understanding the barriers to reaching the potential.

---

[49] Example: In a regulatory organization, reviewers of applications for design certification or license.
[50] Scope: Those who have expressed interest.

2.1. Measurability [39][44].

3. Capability to dissolve the barriers (e.g., process; skills; supporting tools).
4. Evolvability of individual capability:
   4.1. Future learning builds on prior learning in an effective, efficient sequence.
      4.1.1. Growth of the individual's conceptual framework is consistent with the conceptual dependencies.
      4.1.2. Foundational concepts and relationships are ingrained before introducing their compositions.
   4.2. Learning in small increments: Iterative learning cycles.
      4.2.1. Satisfaction in each cycle accelerates the individual's learning.
   4.3. Self-paced learning.
5. Capability to "spread[51]" the learning.
6. Also see Appendix C.

D.7 Sustainability

1. Ability to learn by oneself.
2. Maintainability of the supporting resources (e.g., processes; methods & techniques; tools; reusable assets).
3. Portability of the supporting resources, as underlying implementations change.
4. Evolvability of the supporting resources, esp. reusable assets. Iteratively grow the domain model.
5. Third-party-supported sustenance.

[51] Examples: Train the trainer. Pair learners with complementary capabilities.