DRAFT RESEARCH INFORMATION LETTER 1101:
Technical basis to review hazard analysis of digital safety systems

1 Executive Summary

This research information letter (RIL) provides the US Nuclear Regulatory Commission (NRC)'s licensing staff the technical basis to support their review of hazard analysis (HA) performed on a digital safety system by an applicant seeking a license, amendment to a license, or design certification.

The RIL is prepared in response to a user need request from the Office of New Reactors (NRO), dated December 8, 2011, asking the Office of Nuclear Regulatory Research (RES) for technical assistance to develop the technical basis to support regulatory review of an applicant's HA relevant to digital instrumentation and control (DI&C) safety systems in nuclear power plants (NPPs). Since NRC does not have any relevant explicit guidance on review of HA, NRO intends to use this RIL to develop and support review guidance, specific to a new project applying new technology in a digital safety system for a small modular reactor. This application will also serve as a learning cycle, from which NRC expects to identify needs for future improvements in its review guidance, regulatory guidance, and the underlying technical basis.

The RIL has been focused on issues encountered in NRO's recent licensing reviews – particularly hazards, which are rooted in systemic causes and are contributed through the development of a digital safety system; these hazards are called contributory hazards in the RIL. The technical basis is focused on evaluation of an applicant's HA rather than performing HA. Furthermore, content selection is based on the premise that NRC's existing guidance for reviewers and applicants is being utilized fully. Thus, the RIL is not intended to be a self-contained, comprehensive, and complete technical reference for reviewing HA of digital safety systems in NPPs.

Digital safety systems and the factors contributing to their safety-worthiness are becoming more difficult to analyze, due to many factors, such as increasing inter-connectivity, rapid changes in the nature of systems and the underlying technologies, resulting shortening of accumulated experience for a certain class of systems and the associated technologies, declining supply and replenishment of requisite competence, longer less-track able supply chains, inadequate quality of cross-organizational cross-disciplinary communications, etc. Examples of associated contributory hazards include the following: Inadequate definition of the boundary of the digital safety system being analyzed; incorrect decomposition and allocation of NPP-level safety functions into NPP-wide I&C architecture and then to the digital safety system being analyzed; inadequate identification of safety requirements and associated quality properties and their flow-down into constraints on the architecture of the system and then the architecture of the software or other forms of logic; inadequate flow-down to identify requirements and constraints on technical processes, supporting processes, and organizational processes.

RIL-1101 reflects the state of the art. First, information was synthesized from a variety of publications. However, RES found very little published information organized specifically to support HA reviews for the environment characterized above. Therefore, RES engaged diverse external subject matter experts to acquire knowledge from their respective experiences, and refined the RIL accordingly.

Contents

Figures

Tables

2 Introduction

This research information letter (RIL) provides the US Nuclear Regulatory Commission (NRC)'s licensing staff the technical basis to support their review of hazard analysis (HA) performed on a digital safety system by an applicant seeking a license, amendment to a license, or design certification. Section 2.1 provides a brief background on HA, supported with elaboration in Appendix C. Section 2.2 elaborates on the purpose and intended audience.

The RIL is prepared in response to a user need request from the Office of New Reactors (NRO), dated December 8, 2011, asking the Office of Nuclear Regulatory Research (RES) for technical assistance to develop guidance for regulatory review of an applicant's HA relevant to digital instrumentation and control (I&C) safety systems in nuclear power plants (NPPs). NRC does not have any relevant explicit guidance on HA or review of HA of a digital safety system of the kind seen in recent licensing reviews.

The RIL has been focused on issues encountered in NRO's recent licensing reviews – particularly safety concerns rooted in the development of a digital safety system and systemic causes, called contributory hazards in the RIL. The technical basis is focused on evaluation of an applicant's HA rather than performing HA. Furthermore, content selection is based on the premise that NRC's existing guidance for reviewers and applicants is being utilized fully. Thus, the RIL is not intended to be a self-contained, comprehensive, and complete technical reference. Section 2.3 elaborates the scope.

Digital safety systems and the factors contributing to their safety worthiness are becoming more difficult to analyze, due to many factors, such as the following:

- Increasing inter-connectivity. (Section 3.4 H0-5)
- Rapid changes in the nature of systems and the underlying technologies. (H-OTproc-7)
- Resulting shortening of accumulated experience for a certain class of systems and the associated technologies. (H-OTproc-7)
- Declining supply and replenishment of requisite competence. (Section 3.1 H0-2).
- Longer less-track able supply chains. (Section 3.1 H0-9 explanation list item 2; H-SAE-4)
- Inadequate quality of cross-organizational cross-disciplinary communications, etc. (Section 3.2 H-culture-9)

Examples of associated contributory hazards include the following:

- Inadequate definition of the boundary of the digital safety system being analyzed.
- Incorrect decomposition and allocation of NPP-level safety functions into NPP-wide I&C architecture and then to the digital safety system being analyzed.
- Inadequate identification of safety requirements and associated quality properties and their flow-down into constraints on the architecture of the system and then the architecture of the software or other forms of logic.
- Inadequate flow-down to identify requirements and constraints on technical processes, supporting processes, organizational and processes.

For each "contributory hazard scenario" the RIL also provides examples of conditions that reduce the hazard space. To suit project-specific needs, NRC's licensing offices can select scenarios and corresponding conditions to reduce the respective hazard space, and transform these conditions into review criteria, such as is being done in NRO's mPower design specific

review standard (DSRS) Appendix A [1], review guidance for hazard analysis of a DI&C safety system.

Section 2.4 explains the organization of the RIL.

2.1 Background

A hazard, in general, is defined as "potential for harm." In RIL-1101, the scope of "harm" is limited to the loss of a safety function in an NPP.

Hazards analysis (HA), a systems engineering activity, is the application of systematic and replicable methods to identify hazards, their potential adverse effects, their causes, and the changes in system concept or safety requirements needed to meet the overall safety goals of the system. As part of a system hazard analysis, the applicant will identify the hazards of concern as well as the system requirements and constraints to eliminate, prevent, or control them.

Although the term "hazard analysis" has been used in many ways in various other publications, in RIL-1101 the scope of HA is limited to analysis such that:

1. All losses of concern are identified and mapped from the NPP-level to the I&C level correctly.
2. All hazards leading to the loss of allocated safety functions are identified.
3. Causes, including contributory causes are identified.
4. Commensurate requirements and constraints are identified.

2.2 Purpose and intended audience

The purpose of this research information letter (RIL) is to provide the technical basis to support NRC I&C staff in the exercise of judgment during licensing reviews they[1] perform on an applicant's hazard analysis (HA) of a DI&C system for safety functions in a nuclear power plant (NPP).

The RIL supports an integrative[2] review of an applicant's safety analysis of safety related I&C systems.

The RIL is not intended as an interim or surrogate regulatory guide to licensees or applicants. However, as a technical basis for the limited scope described in the next subsection, it may also be useful to stakeholders outside the NRC.

2.3 Scope

The RIL is a response to NRO's user need request for supporting a specific project. However, the content is sufficiently generic to evolve a successor for broader application, after learning from NRO's first experience [Section 2.3.1]. Content selection is focused on evaluation (rather than performance of HA [Section 2.3.2] for safety automation for NPPs [Sections 2.3.3-2.3.5].

[1] Or their agent or a third party

[2] In current practice, the review is performed through elements in different parts of NRC's standard review plan [1] chapter 7.

Content selection is focused on hazards contributed through systemic causes, especially the engineering development cycle [Section 2.3.6]. Content is focused on supporting a deterministic review process [2.3.7].

Evaluation of HA is not the same as evaluation of safety analysis – the relationship is explained in Section 2.3.8.

2.3.1 Immediate scope limited to learning cycles

Although the content provided in RIL-1101 is intended to be more broadly applicable, the adequacy for broader application has not been validated. These limitations are identified below.

2.3.1.1 Assumptions about areas not well understood

Within the scope described above, RIL-1101 focuses on areas that are not well understood or recognized (e.g., those rooted in systemic causes and contributed through system development activities). To quote from [2]:

> "Common underlying factors[3] involve organizational culture, safety culture, fatigue, other fitness for duty issues, training, experience, habit, habituation, dysfunctional schedule pressure, adverse ambient conditions, work-related distractions, and the like. Nevertheless, addressing ineffective hazard recognition instances, addressing the factors that resulted in them, and addressing their extents would be a highly cost-effective initiative."

Judgment used in the selection of coverage of the subject matter is based on assumptions about what is not well understood. These assumptions should be tested through learning cycles, before broader application of RIL-1101.

It is assumed that hazards internal to the DI&C system, contributed by hardware elements are well understood. Therefore, review of hardware-related HA is addressed in Section 3.7.1 only briefly[4].

2.3.1.2 Extrapolation from recent licensing experience

Subject matter[5] selection was also based on issues experienced by the licensing offices in the last several review projects, with the assumption that those issues were indicative of a trend. It is possible that new issues surface in upcoming reviews that were not explicitly addressed in RIL-1101. Its adequacy should be tested through several learning cycles, before broader application.

2.3.1.3 Support for application-specific customization of the SRP Chapter 7

Selection and extent of treatment of subject matter is further narrowed to support application[6]-specific customization of the SRP Chapter 7.

---

[3] RIL-1101 scope does not include all the quoted factors.

[4] Appendix C leads to more information through links to supporting references.

[5] Contributory hazards

[6] Example: mPower project

2.3.2 Focus on evaluation rather than performance of hazard analysis

RIL-1101 is focused on providing the technical basis for exercising judgment during licensing review activities. RIL-1101 is not intended as an interim or surrogate regulatory guide to licensees or applicants. RIL-1101 is not intended to provide guidance on how to perform HA.

Prevalent public standards and guides on HA elaborate on techniques to perform HA, but there is little information available on criteria for evaluating the results of HA, even though the concept of hazard analysis is over four decades old.

2.3.3 Focus on licensing reviews of safety automation

Although results from HA, in general, include requirements for aspects outside the initially commissioned DI&C safety system (e.g., training, maintenance, and operational and maintenance environments), RIL-1101 does not provide the technical basis to evaluate requirements concerning operation and maintenance and the people engaged therein.

Consistent with the scope of the SRP Chapter 7, the scope of RIL-1101 is limited to the safety automation. The human, the human-automation interface, and the associated control room are treated as part of the environment (Section 3.4.1) of the system in scope.

2.3.4 Focus on safety related systems for NPPs

Prevalent public standards [3] and guides [4], [5], and [6] on HA are oriented to the general case of a system implementing a variety of functions with varying degrees of criticality. In contrast, RIL-1101 focuses on safety related systems for NPPs, where the consequence of a mishap, unwanted release of radioactivity into the environment (known in HA terminology as the loss), is of the highest degree of severity. The scope includes a system realizing a safety function, as well as any system or element on which the correct timely performance of a safety function is dependent.

Review of analysis for hazards external to the DI&C system, in general, is covered in other parts of NRC's standard review plan [6]. RIL-1101 considers external hazards primarily from the perspective of issues with interfaces and interactions across different parts of the organization.

RIL-1101 does not elaborate on reviewing the analysis of hazards from the physical environment (Section 3.4.1; Appendices E.4 and E.5)), because the state of practice in these aspects of HA is relatively mature.

2.3.5 Types of systems intended in scope

RIL-1101 describes the evaluation of an applicant's HA associated with digital safety systems for new and advanced reactors. The scope of this RIL is limited to a system realizing a safety function or on which the correct timely performance of a safety function is dependent. Other elements interfacing with, interacting with or affecting the DI&C safety system are treated as parts of its environment; to that extent, such environment is also within the scope (see Section 3.4.1).

The scope treats any change to a previously analyzed DI&C safety system as a new hazard analysis review cycle.

2.3.6 Focus on contributory hazards rooted in systemic causes

The RIL is focused on hazards rooted in systemic causes and contributed through system development activities (elaborated in Sections 3.1-3.6 and 3.8-3.9).

Systemic causes are a special kind of common causes of failure[7] (CCF), such that, often, their propagation is pervasive; that is, there could be many propagation paths, and these are not easy to discover and analyze. (In contrast, the propagation path from a CCF due to the breakdown of a component in a hardware system is relatively easier to identify and analyze). In a system with complex logic[8], recognizing and understanding the cause-effect relationships or influence paths well enough requires explicit identification of a variety of dependencies. Some dependencies can be recognized in the analysis of the system itself (e.g., Sections 3.4.2, 3.6.1, 3.6.3, 3.7.1, 3.8.1, and 3.9.1). Some can be recognized through analyzing interactions of the system with its environment (e.g., Section 3.4.1). Many other dependencies occur through organizational processes (e.g., Section 3.2), technical processes (e.g., Sections 3.3, 3.4.3, 3.5, 3.6.2, 3.6.4, 3.7.2, 3.8.2, 3.9.2), and supporting or auxiliary processes. RIL-1101 does not enumerate all possible factors exhaustively, but identifies the need for a thorough understanding and recognition of the various forms of dependency, illustrated through a few examples.

Dependencies can arise in various ways[9]; for example:

- Due to an inadequate communication protocol
- Due to data (e.g., incorrect value; incorrect time of arrival)
- Due to maintenance (e.g., system returned to operation in incorrect configuration)

Dependencies on common sources of defects or deficiencies can render homogeneous redundancy ineffective, because the same defect can repeat in each redundant element; for example:

- Defect or deficiency[10] in a requirement
- Defect or deficiency in implementation of the application software
- Defect or deficiency in implementation or configuration of the system software

Dependencies can propagate effect of a defect or deficiency to independent and functionally different units; for example:

- Dependency on common internal information; for example:
  - Year 2000 “bug”
  - Count of cycles since the last reset
- Dependency on conditions, external to the units; for example:
  - Usage of resources dependent upon process transients

2.3.7 Scope excludes risk quantification

The scope excludes quantification[11] of severity of consequence and probability of occurrence[12].

---

[7] Meaning in this context: Loss of the top-level safety goal.

[8] For example, in the form of software.

[9] Often, unplanned and unexpected.

[10] Issue: If requirements are deficient, the terms “failure” and “defect” are not applicable; the CCF notion, applied to a specified system, does not serve well; failure analysis and defect analysis do not serve as adequate hazard analysis.

This exclusion is based on the following reasons:

1. The consequence of the failure of a safety function is treated at the highest level of severity.
2. Logic[13] leading to a safety function must execute correctly or the consequence is of the same level of severity as the DI&C safety system - no mitigation is possible.
3. A safety system in an NPP is an independent layer of defense; no credit for meeting the allocated safety requirements is assumed from another layer of defense[14].
4. Contributory hazards originating in the system development lifecycle or rooted in systemic causes[15] are pervasive in their effects. Therefore, their effects are not separable and analyzable as mutually exclusive cause-effect paths leading to a safety function. Thus, the technique of redundancy, as used in hardware elements, is also not helpful in addressing such contributory hazards.
5. Since design certification for DI&C platforms, tools, processes, etc. allows multiple future applications, the same elements could be replicated for different NPP functions, multiplying vulnerability to the same contributory hazard. This multiplication effect is not bounded.

2.3.8 Relation between hazard analysis and safety analysis

Figure **1** shows the relationship of HA[16], as treated in RIL-1101 to other activities contributing to the applicant's safety analysis report (SAR).

- The result of HA activities (depicted in the upper left sector of Figure **1**) is a set of safety requirements and constraints (included in the design bases), which are verifiable independently by a third party not involved in the development of the safety system. Also included are derived requirements and constraints on the design and implementation of the safety system. This set of requirements and constraints is intended to be a part of the licensing basis.
- Activities in the scope of inspections, tests, analyses, and acceptance criteria (ITAAC) (depicted in the upper right sector of Figure 1) would verify that these requirements and constraints have been satisfied. These activities are not a part of reviewing hazard analysis, as delineated in RIL-1101.
- Whereas each verification activity yields corresponding evidence (e.g., that a certain item, such as hardware or firmware or software has met the requirements and constraints allocated to it), overall verification includes the integration of all the various evidence items (depicted in the lower sector of Figure 1) in a way that demonstrates that the overall safety requirements of the system have been satisfied. These activities are also not a part of hazard analysis, as delineated in RIL-1101.

---

[11] Scope also excludes qualitative classification or gradation.

[12] Exception: Section 3.7.1 pertaining to hardware components.

[13] Example: Software

[14] An independent layer of defense protects against the unknowns and uncertainties in the other layers of defense.

[15] The focus of RIL-1101

[16] Figure 1 is a simplified depiction, See note.

Note: Figure **1**, simplified for illustrating the relationship with the overall safety analysis, omits the following complexity. HA is iterated at each phase in the development lifecycle of a system and the development lifecycle of each of its elements. Iteration at any phase may reveal that the phase has introduced a new hazard. The corrective action may simply be a revision within that phase or it may require a change in a preceding phase, invalidating the result of the preceding phase. The latter case may require multiple iterations and tradeoffs, making the process more complex.

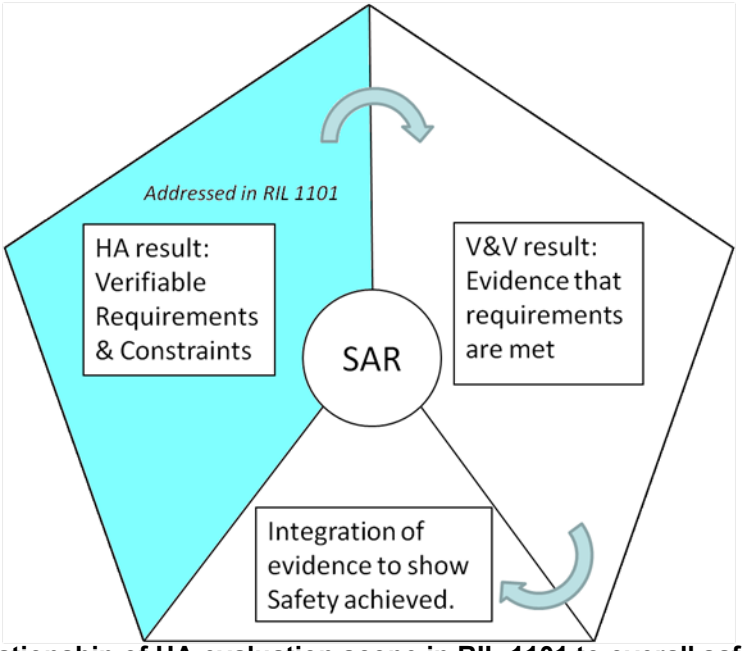

Figure 1: Relationship of HA-evaluation scope in RIL-1101 to overall safety analysis

2.4 Organization of RIL-1101

Section 3 provides the technical basis to support NRC I&C staff in the review of an applicant's HA. In accordance with NRO's (User's) request, supporting explanatory information in the appendices. For example, Appendix C, which is incorporated by reference in Section 3.1 Table 1 item H0-1G, provides state-of-the-art information on HA methods.

Section 3 is organized by groups of contributory hazards, as explained below.

1. Subsections 3.1 - 3.3 group contributory hazards that are applicable to all phases of the development lifecycle; typically, these are controlled before starting the development of a particular system.
2. Subsections 3.4 - 3.9 group contributory hazards from the perspectives of phases of the development lifecycle.
3. Within each subsection, contributory hazards are grouped in one or more tables. These groupings serve as different perspectives on intertwined issues, and are not intended to be mutually exclusive partitions.

4. While contributory hazards might manifest themselves or might be discovered in any of several phases of the development lifecycle or levels of integration of a digital safety system, the RIL attempts to place the item in a group corresponding to the earliest prevention opportunity.
5. For each group of contributory hazards, the table organizes related information as follows.
   5.1. The table title (explained in the narrative introducing it) bounds the scope and context of entries in the table. The context of a cell in the table is defined by the title of the table.
   5.2. In the left block of each table, the entry in a row is an example of a scenario or a category of contributory hazards.
   5.3. In each row, the right block associated with each contributory hazard provides an example of a condition that reduces the respective hazard space.
   5.4. Each contributory hazard is uniquely identified with a label of the type “H-alpha-<i>”
      5.4.1. The “H-alpha-” part of the label is in the column title, applicable to each row, but not repeated. Examples: H-0-; H-culture-; H-OTproc-.
      5.4.2. The <i> portion of the label is a numeric, unique to each contributory hazard item.
      5.4.3. For example, H-SAE-1 is a complete label for a contributory hazard item.
   5.5. A label of the type “H-alpha-<i>-G<j>” identifies a condition G<j> that reduces the H-alpha-<i> space[17].
      5.5.1. For example, H-SAE-1G1 is a condition: associated with H-SAE-1.
6. Although the RIL includes hyperlinks for ease of navigating across related items, it does not provide hyperlinks between all of the many-to-many relationships that exist; for example:
   6.1. between contributory hazards
   6.2. between contributory hazards and conditions reducing the respective hazard spaces
   6.3. between conditions reducing the various hazard spaces, and
   6.4. across hazard groups
7. A note structure, distinguished by indentation, font type and size provides a brief explanation or example for an “H-alpha-<i>” or “H-alpha-<i>-G<j>” paragraph (if needed).
8. A link to an item in an appendix leads to further elaboration and background.

Section 4 explains where HA-review fits in the regulatory framework.

Section 5 summarizes the contribution of RIL-1101 and Section 6 outlines the follow-on research and development (R&D) identified in the course of this work (e.g., unresolved review comments).

Where a word or expression is used in a meaning more specific than or different from the common usage defined in mainstream dictionaries, it is defined in Appendix A: Glossary. Its first occurrence is hyperlinked to that definition.

[17] Possible combinations of specific conditions, relevant to this scenario, that lead to potential loss of concern for the system being analyzed.

3 Considerations in evaluating Hazard Analysis

RIL-1101 addresses primarily factors contributing to the degradation of a safety function, originating in the system development lifecycle[18]. These factors are part of a network of causes or dependencies that result in some defect or deficiency in the system, which could lead to the degradation of a safety function. RIL-1101 refers to these factors as contributory hazards.

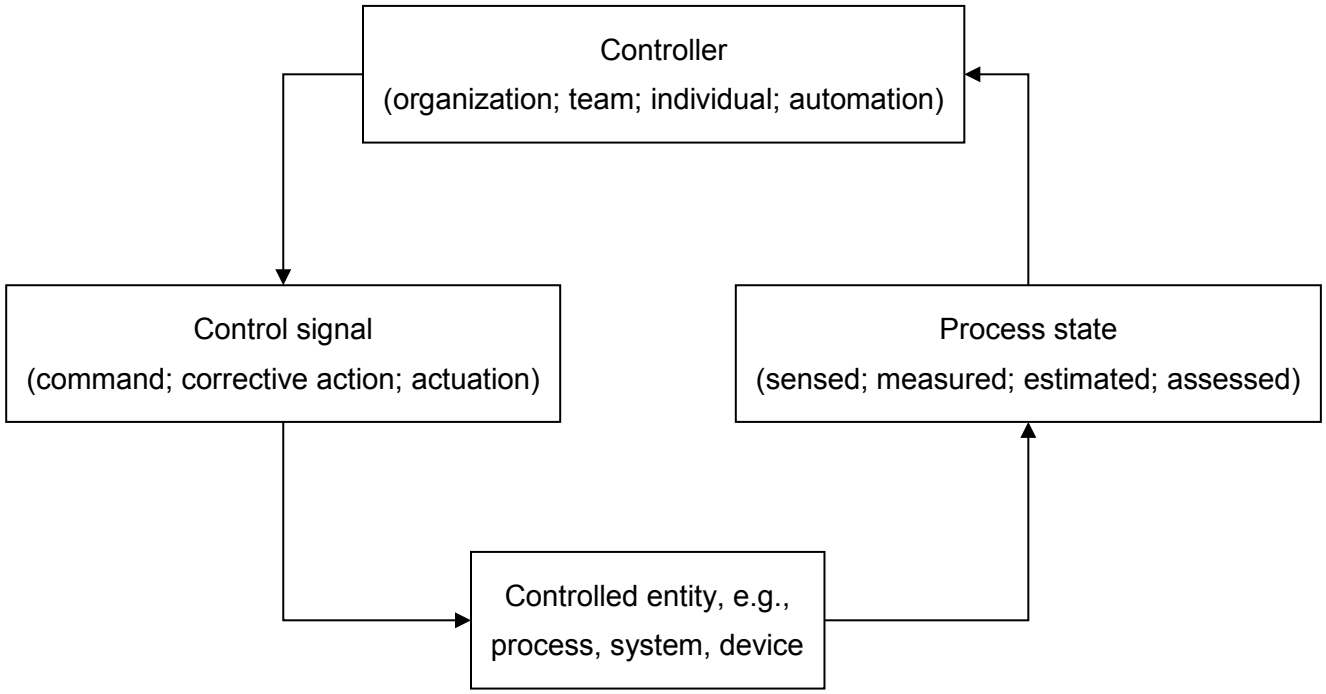

Figure 2: Example of a dependency structure (cyclic graph)

However, recent experience [7]-[9] has revealed that propagation paths of hazards are not always linear, and cause-effect relationships are not always direct chains. The indirect propagation of effects (e.g., degradation of a safety function) and interactions and consequences are not well understood. For example, [10] characterizes these as "*issues that transcend the functions of individual components and involve interactions between components within the system as well as the interaction of the system with the environment.*" Traditional techniques for hazard analysis,in their basic form, such as fault tree analysis (FTA) [11][12], and failure modes and effects analysis for design DFMEA [13][14], used in NPPs, do not support their discovery well.

RIL-1101 focuses on contributory hazards from such interactions and indirect couplings, and from more complex propagation paths (e.g., feedback paths), and transformation of a fault during propagation. For example, in a well-controlled system, such as one intended for a safety-critical environment, Figure 2 illustrates that the dependency structure is a cyclic graph. It is a well-known generic control structure, for which well-known analysis techniques exist. It can be applied to a safety-related system in its concept phase (Section 3.4) or to its element (Sections 3.7; 3.8-3.9). It can also be applied to the technical processes (Section 3.3) to develop a safety

[18] rather than random hardware failures during operation

related system or its element. It can also be applied to the organizational processes (Section 3.2) that influence the development processes.

Figure 4 is an example of a generic dependency structure, illustrating how the transformation of a work product depends upon the process activity and factors upon which it depends. A process dependency model can be applied to organize and understand the contribution of organizational processes (Section 3.2), as well as technical processes (Section 3.3). The model is also applicable to any other creative, but deterministic, activity, from which predictable, verifiable, analyzable results are needed. Each activity step is affected by the procedures and resources, such as competence (e.g., H-culture-6G2), information, tool, or other aid) employed in performing that activity. The quality of the work product depends upon the quality of the procedures, resources and their utilization, that is, any deficiency is a contributory hazard).

In addition to the factors directly in the causal paths, hazards can also be contributed from side effects such as interferences across activities and resources.

Experience with complex systems in general [15] and with digital systems for critical functions in diverse application sectors[19] has revealed that the current code of practice does not assure absence of conditions having the potential for functional degradation of safety system performance.

The difficulties NRO experienced (e.g., as reported to ACRS [16]) are examples of the more general trends of increasing system complexity and increasing contribution of systemic causes towards malfunctions. Generally accepted engineering standards[20] do not provide sufficiently specific guidance to ensure their technically consistent, efficient application to digital systems with such complexity. Such reviews require significant additional information [16] from the applicant, significant additional review effort and reliance on judgment, in order to address the gap in the existing review guidance. These gaps were identified in [18] as uncertainties in the assurance of digital safety systems. As depicted in Figure 3, RIL-1101 focuses on the challenges from these uncertainties, characterized as contributory hazards, and couples these uncertainties in the assurance of digital safety systems with conditions that reduce the respective hazard spaces.

[19] Also see case studies reviewed in Appendix G.

[20] mentioned in 10 CFR 50.34(a)(ii)(B); referenced in NRC's regulatory guides

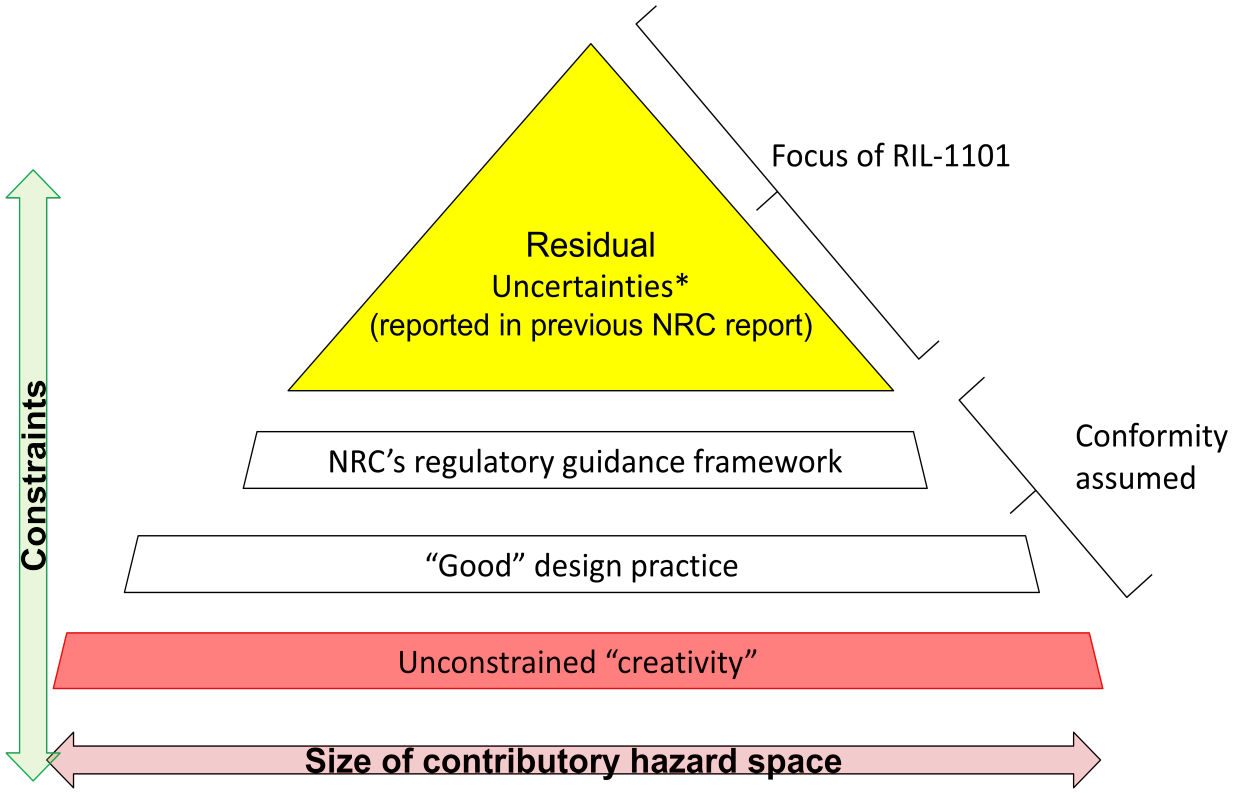

Figure 3: Contributory hazard space in focus

3.1 Evaluation of Overall Hazard Analysis

Table 1 outlines some overarching considerations in evaluating the HA of a DI&C system. These factors, shown as contributory hazards in Table 1, affect the quality of the HA results broadly. The scope of entries in Table 1 is limited to the performance of HA. An integrative explanation of the considered factors is given after the table.

Table 1: Considerations in broadly evaluating performance of hazard analysis

Contributory hazard		**Examples of conditions that reduce the hazard space**	
ID H-0-	**Description**	**ID H-0-**	**Description**
1	HA approach is not suitable to the system, element, intermediate-phase work product, process or activity being analyzed.	**1G**	The selected HA approach is well-matched to the system aspect, element, development phase, or work product being analyzed - see Appendix C.
2	Competence in performing HA is not adequate for the system being analyzed.	**2G0**	The HA is performed with the requisite complement[21] of competence. Also see Appendix C, [H0-{2G1.1, 2G1.2}], [H-culture-6G2]
		2G1	The competence includes breadth and depth.

[21] For a complex system involving multiple disciplines, HA is expected to be a team effort.

	(Also see **H-SRE-1**)	**2G1.1**	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. 1. Knowledge of respective operating experience (what can go wrong) 2. Track record of learning from it (how to prevent what went wrong) 3. Ability to express and explain to others insights from deep knowledge.
		2G1.2	Breadth[22]: Individuals are able to understand how their respective roles fit into the overall HA, including the associated inter-dependencies. 1. Knowledge of the environment[23] of the safety system and its development. 2. Experience in analysis of for hazard groups such as those identified in RIL-1101. 3. Experience in deriving requirements and constraints to avoid or eliminate contributory hazards. 4. Experience commensurate to the complexity of the system or item.
		2G2	The HA-team has the requisite team-work capabilities; for example: 1. A diversity of perspectives and backgrounds 2. Capability to elicit and evoke differing responses, esp. minority viewpoints 3. Capability to understand other team members' reference-frames 4. Capability to assimilate differences, neutralizing biases 5. Capability to converge towards objectivity 6. Other constructive group interaction skills
3	Validation is inadequate – impaired, because people in the developer's organization are unable to think independently. Intra-organizational reviews suffer from "GroupThink."	**3G1**	The HA, including elements upon which it is dependent (see H0-8, H0-9) and the resulting requirements and constraints, is validated (in [14] common position (CP) 2.1.3.2.6) independently[24]. 1. The HA-validation team has the requisite competence [links]. 2. The HA-validation team provides perspectives

[22] Provide continuity to the HA-team across lifecycle phases.

[23] Also see Section 3.4.1.

[24] The independent team may engage the HA-team in review and walks through its work products.

			and background different from the team performing the HA.
6	Hazard controls needed to satisfy system constraints (which prevent hazards) are inadequate	**6G1**	Hazard controls are identified and validated to be correct, complete, and consistent. [H0-7G1↓]
7	Flow down from the controls [H0-6-G1] to verifiable requirements and constraints is inadequate.	**7G1**	Requirements and constraints [H0-6G1↑] are formulated and validated to be correct, complete, and consistent in consideration of the preference[25] order 1-4 as follows: 1. Prevent hazard 2. Eliminate hazard 3. Contain hazard (prevent propagation) [H-SR-4G4↓] 4. Monitor, detect and mitigate[26] hazard 4.1. Monitor [H-SR-4G1↓] 4.2. Detect [H-SR-4G2↓] 4.3. Intervene [H-SR-4G3↓] 4.4. Notify (some independent agent)[27] [H-SR-4G5↓] 4.5. (Recipient[28] of the notification) Perform safety-supporting function 4.6. Confirm safe state
8	The analysis is not propagated to elements in an NPP on which the system being analyzed depends or the safety functions allocated to it depend. See in Table 4 H-ProcState-5	**8G1**	All dependencies are identified and analyzed, to assure that safety is not impaired.
9	The analysis is not propagated to processes and process activities on which the integrity of the system being analyzed	**9-G**	All dependencies are identified and analyzed, to assure that safety is not impaired, including organizational processes, management processes, supporting processes, and technical processes, .

[25] It is based on extent of reduction of hazard space, potential fault space, and uncertainty space.

[26] Maintain safe state

[27] e.g.: Operator; another automation device or system.

[28] e.g.: Human; other automation.

	depends or the safety functions allocated to it depend. See in Table 4 H-ProcState-6 H-ProcState-7 H-ProcState-8		
10	Propagated effect of changes introduces inconsistencies, invalidating previously performed HA.	**10G1**	Starting from the initial HA performed on the functional concept (in [14] CP 2.1.3.2.3) the HA is revised at every phase[29] in the development lifecycle, with change control management and configuration management. Examples of contributory hazards that may be discovered include: 1. Hardware faults 2. Unanalyzed conditions [H-S-1.1.1G1↑].
		10G2	The HA has been iterated until no new hazards are identified [H0_8G1↑]. 1. Added monitoring, detection, mitigation or other requirement has not introduced some new hazard. 2. Some complexity-increasing side effect from the change has not introduced some other, yet-unanalyzed hazard.
10.1	Hazard-introducing effect of iterations is not well understood.	**10.1G**	H0-9.1{G1 – G7}↑ H0-10-G1↑ H0-10-G2↑
11	Required hazard control action is degraded.	**11G1**	Each required control action is analyzed for ways it can lead to the hazard; for example: 1. Not provided; for example: 1.1. Data sent on a communication bus is not delivered. 2. Provided when not needed 3. Incorrect state transition (e.g., combination of 4-5 below). 4. Incorrect value provided; for example: 4.1. Invalid data 4.2. Stale input value is treated inconsistently. 4.3. Undefined type of data 4.4. Incorrect message format 4.5. Incorrect initialization

[29] Also apply these considerations to successive phases of the system development lifecycle.

			5. Provided at the wrong time or out of sequence 6. Provided for too long a duration (e.g., for continuous-control functions). 7. Provided for too short a duration; for example: 7.1. Signal is de-activated too early (e.g., for continuous-control functions). 8. Intermittent, when required to be steady; for example: 8.1. Chatter or flutter 8.2. Pulse; spike 8.3. Degradation is erratic 9. Interferes with another action; for example: 9.1. Deprives access to a needed resource; for example: 9.1.1. "Babbling idiot" 9.1.2. Locking up and not releasing resource 9.2. Corrupts needed information 10. Byzantine behavior
12	Hazards in modes of operation other than the "at power" normal mode, or in transition from one mode to another are not adequately understood or analyzed.	**12G1**	HA is performed for all modes of operation (in [14] CP 2.1.3.2.7) and corresponding requirements & constraints are derived (e.g., see checklist in Appendix H).

The following notes explain certain contributory hazards identified in Table 1

H0-1: An ill-matched technique can increase the difficulty of identifying a hazard. Although RIL-1101 does not recommend any particular technique for any aspect of HA, Appendix C provides a broad survey. Techniques suitable for identification of hazards contributed during the engineering lifecycle continue to evolve.

H0-2: Inadequate replenishment of requisite competence: The DI&C 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. Development and training of the workforce is driven more by consumer products and information technology (IT) industries than by high-consequence automation. "Development of DI&C systems for the highest level of safety" is a very small, niche in the market.

H0-{2, 3}: Hazard identification, especially at the conceptual phase of the system development lifecycle, is primarily a manual activity. Independent validation is also a manual activity. The required competence increases with the complexity of the system.

H0-{8-9}: The intent of reviewing for these factors is to check that the system on which HA is to be performed and its context (environment) are correctly identified, the

dependencies correctly understood, the primary hazards (external and internal) are identified, and the commensurate constraints are identified.

H0-{8-9}: Whereas "ineffective hazard recognition" has been recognized as a serious issue [6], unrecognized dependencies are an increasing contributor to this issue, as the complexity of organizations, processes, and systems is increasing. In addition to the lack of awareness, lapses could occur because of inability to track and maintain a consistent understanding of the dependencies[30].

H0-{6-7; 11}: These factors address the flow down from direct hazards to system constraints to required controls to verifiable requirements and constraints. The following sections provide further information.

H0-{8-9}: The extent of dependencies on processes, including the physical processes in the plant, may not be fully understood. From an NRC reviewer's perspective, a third party certification of the system could provide the requisite assurance that all dependencies have been identified and their effects analyzed. For example, Figure 4 depicts an abstraction of process-related direct dependencies. Following are examples, indicating less than adequate controls and thus less than adequate understanding of inter-dependencies across processes.

1. Organizational processes lack such controls; or
2. The organization does not apply such controls to the feeder processes or food chain or supply chain; or
3. The organization does not plan for such understanding at system concept phase of the lifecycle.

H0-9: The extent of dependencies in a system and its elements may not be fully understood or may not be understood in the same way across all parties engaged in developing the system or multiple changes might introduce obscurity. From an NRC reviewer's perspective, a third party certification of the system could provide the requisite assurance that all dependencies have been identified and their effects analyzed. Following are examples, indicating less than adequate controls and thus less than adequate understanding of inter-dependencies across systems or elements of a system.

1. The organizational processes lack such controls; or
2. The organization does not apply such controls down the supply chain; or
3. The organization does not plan for such understanding.

H0-10.1: When HA is performed at some stage in the development lifecycle of the system and its elements, additional safety requirements and constraints could be discovered. Inclusion of those requirements[31] could change the system concept or design, requiring another HA cycle to evaluate the impact of such changes. The cumulative and cascading effects of these iterations may not be well understood, introducing the potential to miss subtle implications of a change.

[30] The state of practice in representing and analyzing such dependencies is relatively weak, as discussed in Appendix C..

[31] Incorrect, incomplete, ambiguous or improperly implemented safety requirements can lead to hazards.

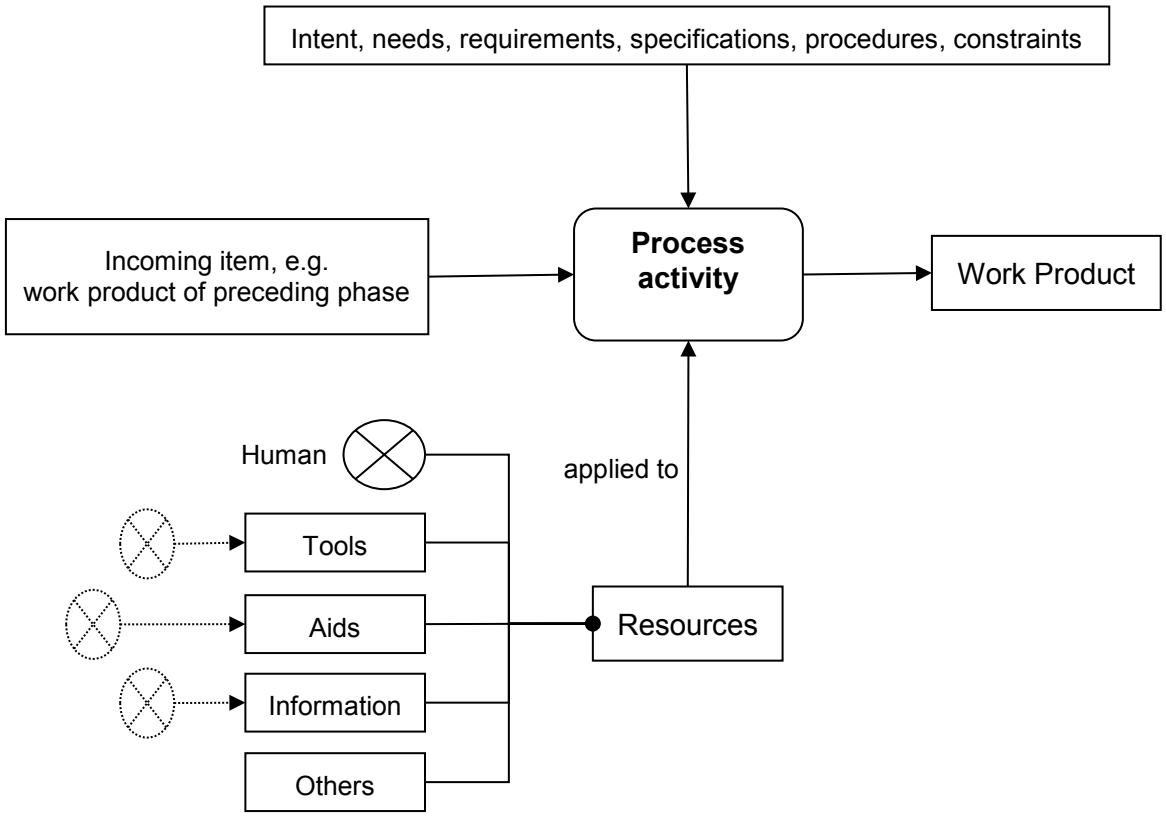

Figure 4: Factors influencing the work product of development

3.2 Contributory hazards from the organizational processes

Organizational processes include management processes, infrastructural processes, and other supporting processes. The term, “supporting processes” includes change impact analysis process and maintenance processes upon which the system design is predicated.

The culture of an organization with respect to safety engineering, the processes of managing and engineering safety (included within “organizational processes”) have pervasive effects, that is, their contribution cannot be analyzed[32] as causal events. Table 2 identifies some common concerns, adapted from Annex B in [14].

[32] This aspect of HA roughly corresponds to but is significantly broader than the HA mentioned in [4] Table 1a.

Table 2: Organization's culture: Examples of contributory hazards

Contributory hazard		Conditions that reduce the hazard space	
ID H-culture-	**Description**	**ID H-culture-**	**Description**
1	The reward system favors short-term goals, placing cost and schedule over safety and quality (sliding on a slippery slope, not fully cognizant of the cumulative effect of compromises).	1G1	The reward system supports and motivates the effective achievement of safety. Safety is the highest priority.
		1G2	The reward system penalizes those who take shortcuts that jeopardize safety or quality.
		1G3	The organization has integrity.
		1G3.1	The process[33] state is consistent between reality and its representation.
		1G4	Lifecycle economics supporting safety and quality drive the organization
2	Accountability (e.g., as illustrated in Figure 2 and Figure 4) is not traceable; achievement of safety cannot be assured	2G1	The process assures the accountability for effective achievement of safety
		2G1.1	Influencing factors are organized in an effective control structure (Figure 2).
3	Personnel assessing safety, quality, and their governing processes are influenced unduly by those responsible for execution [H-culture-1↑]	3G1	The processes for safety, quality, verification & validation, and configuration management are independent of the main development process.
4	Personnel feel pressure to conform: 1. "Stacking the deck" when forming review groups. 2. Dissenter is ostracized or labeled as "not a team player" 3. Dissent reflects negatively on performance reviews. 4. "Minority dissenter" is labeled or treated as "troublemaker" or "not a team player" or "whistleblower." 5. Concerned employees fear repercussion.	4G1	Such behavior is discouraged and penalized.
		4G2	The process uses diversity to advantage. 1. Intellectual diversity is sought, valued, and integrated in all processes. 2. "Speaking up" (raising safety concern) is rewarded.
		4G3	Supporting communication and decision-making channels exist and the management encourages their usage (e.g., individual can express safety concern directly to those ultimately responsible).

[33] Applicable to any activity in any process in the organization, influenced by its management.

	[H-culture-1↑]	4G4	1. Each hazard identified by anyone is recorded[34] for HA. 2. If it is decided that it is not a hazard, the reasoning is recorded, along with dissenting positions, if any 3. Each issue raised by anyone is recorded. 4. A resolution process ensures that the analysis, evaluation, resolution and disposition of the issue are performed in a timely and effective manner.
5	Management reacts only when there is a problem in the field.	5G1	Safety and quality issues are discovered and resolved from the earliest stage in the product lifecycle.
6	The required resources (quality; quantity) are not planned or allocated in a timely manner.	6G1	Resources required[35] are estimated with adequate accuracy[36] in a timely manner.
		6G2	The required resources are allocated in time.
		6G3	Skilled resources have the competence commensurate to the activity assigned. [H0-2G; H-SRE-1G{1,2,3}]
7	A critical cognitive task is interrupted to switch its assignee across multiple tasks; such interruptions could increase the potential of mistakes, thereby increasing the potential fault space or contributory hazard space.	7G1	Run critical cognitive tasks to completion (default practice of the organization). Interruption is allowed only when the task has progressed to a stable, well-understood state, such that the interruption does not increase the hazard space.
8	Processes do not produce deterministic, predictable results.	8G1	A defined, documented, disciplined process is followed in all dimensions at all levels, as needed for consistent achievement of safety; for example: 1. Management 2. Engineering 3. Procurement 4. Verification 5. Validation 6. Safety assessment 7. Safety audit

[34] Example of a record: Hazard log

[35] Example: Type of competence; degree or level of competence or proficiency; amount of effort time

[36] Implied constraint: Processes are adequately designed and controlled. [H0-9.1G1; H-culture-8G1; H-OTproc-1G]

9	When system lifecycle activities are distributed across multiple organizations or parts of the same organization, safety-relevant information[37] is not communicated efficiently, letting key items of information "fall through the cracks." [H-SRE-7↓]	9G1	Cross-organizational dependencies are identified and tracked explicitly.
		9G2	Safety goals drive open collaborative communications across boundaries.
		9G3	Decomposition of safety goals from NPP level analysis and allocation to safety related systems is complete, correct, and consistent and unambiguous.
10	Mistakes repeat.	10G1	Continuous improvement is integral to all processes.
11	Heavy dependence on testing[38] at the end of the product development cycle. By that stage: 1. It often becomes infeasible to correct the problem soundly. 2. Patches increase complexity and impair verifiability.	11G1	See H-culture-5G1
		11G2	Technical processes are designed to prevent safety and quality issues as early in the development lifecycle as possible.
		11G3	Processes for safety, quality, V&V and configuration control are planned[39] and designed to prevent and discover safety and quality issues as early in the development lifecycle as possible.
12	Dependence on implicit information, e.g. implicit assumptions. [H-ProcState-4↑] [H-OTproc-8↓]	12G1	All information upon which assurability of safety depends is explicit and configuration controlled.
Notes: A symbol of the form [H-culture-8↑] indicates that this item "supports" or is "derived from" the linked item A symbol of the form [H-S-1.1G1↓] indicates that this item "requires" the linked item (e.g., H-S-1.1G1) is required			
Explanatory notes for contributory factors identified in Table 2: **H-culture-9:** Cross-disciplinary, cross-organizational communications quality is affected by stretched lines of communication across the NPP operator (the utility-licensee), the supplier of the plant, the supplier of the DI&C system, and the supplier of components of the DI&C system.			

3.3 Contributory hazards from the organization's technical processes

Improperly designed or executed technical processes can lead to defects in a system. Examples of technical processes include, but are not limited to the following:

- Requirements engineering – also see Section 3.5

[37] Implied constraint: H0-9G

[38] It is unlikely that testing as the only means of verification will suffice.

[39] Examples of work products: Safety plan; quality plan; V&V plan, demonstrating completeness of coverage.

- Architecture engineering – also see Section 3.6
- Design, e.g. see Section 3.8
- Implementation, e.g. see Section 3.9
- Verification activities by those performing these development activities
- Third party verification
- Process assessment
- Process audit.

Examples of some general contributory hazards and conditions to reduce the respective hazard space are given in Table 3 (adapted from Appendix A.1 in [13]), premised on the satisfaction of constraints identified in Table 2.

Table 3: Technical processes: Examples of contributory hazards

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-OTproc-	**Description**	**ID H-OTproc-**	**Description**
1	Technical processes are not deterministic [H-culture-8↑], that is, correctness of results cannot be assured	1G	The organization's technical processes are defined to a level of detail such that for each work element involved, there is a specification of the competence, tools, information, and other resources required (see Figure 4) to execute that work element correctly and to integrate results of such work elements correctly. [H-culture-8G1↑]
2	Any process variable in any work element may contribute to some defect, if not adequately controlled. [H-OTproc-1↑]	2G	Each process variable in each work element is controlled and supported with commensurate methods, tools, and competence. [H-OTproc-1G↑] (Figure 2; Figure 4)
3	Cognitive load (or intellectual complexity) imposed by a specified work element exceeds the capability of assigned personnel. [H-culture-6↑]	3G1	The cognitive load imposed by a specified work element, including an integration activity, is assured to be well within the capability[40] of personnel available to perform that activity.
3.1	Difficulty of understanding the architecture is a contributor to the cognitive load.	3G2	The system architecture is analyzable and comprehensible. [H-OTProc-3G1↑]. [H-S-1.1G1↓; H-S-2G6↓]
4	Mistakes (leading to defects) occur[41]; however, technical processes are not designed with the commensurate robustness and resilience to	4G	The organization's technical processes include processes to detect and recover from mistakes (e.g., verification, audit).

[40] This may require certification of personnel through a standardized process.

[41] Perfection in human performance is not achievable – at least, not in a sustainable manner.

	protect from such mistakes.		
5	The organization believes incorrectly that its processes are adequate, exposing it to unknown sources of defects, for which it cannot identify the causes. [H0-9.1↑; H0-9.3↑]	5G1	The process is assessed and certified independently.
		5G2	Qualified resources are available to assess the process. [H-culture-6G1; H-culture-6G2]
6	The processes in real-life execution deviate from the designed processes, resulting in exposure to unknown sources of defects, for which it cannot identify the causes.		[H-culture-{1G3.1; 2G1}.1
		6G1	The process in execution is audited independently.
		6G2	Qualified resources are available to audit the process.
7	In comparison to previous generation systems, less accumulated experience and reusable results; for example, shorter lifecycles of implemented systems or configurations leading to • Less accumulated experience on the same item • Changing environments for the same item		H0-9G H-culture-{2G1.1; 8G1} H-OTproc-{1G; 2G}
		7G1	More rigorous analysis – see Table 1, Table 2. Commensurate conservatively derived requirements and constraints.
8	Engineering models lack adequate fidelity to reality, i.e. modeling abstractions are not sound.	8G1	Modeling abstractions are validated.

Notes:
A symbol of the form [H-culture-8↑] indicates that this item "supports" or is "derived from" the linked item
A symbol of the form [H-S-1.1G1↓] indicates that this item "requires" or spawns the linked item (e.g., H-S-1.1G1) is required

Explanatory notes:

H-OTproc-3: With increasing complexity [11] of systems, processes, and organizations, involving people from multiple organizations, multiple disciplines, multiple locations, and increasing content of software (or other implementation of logic), there is an increase in contribution of hazards rooted in engineering activities, relative to unmitigated effects of hardware failures; for example:

- Requirements engineering, addressed in Section 3.5
- Architecture engineering, addressed in Section 3.6
- Software engineering, addressed in Sections 3.6.4 and 3.8

3.4 Evaluation of Hazard Analysis - System Concept

The system concept, sometimes known as the functional concept (of the intended system), is described in terms of the initial requirements associated with it and its relationship with its environment, including the boundary and the assumptions on which these are based. Sometimes, the associated requirements are embodied in a "concept of operations" document. Sometimes HA[42] of a functional concept is called preliminary hazard analysis (PHA[43]) – (also see Appendix C-2.2.

In practice, the degree of specificity of a system concept varies over a wide range, such that sometimes the initial concept is so vague that it leads to misunderstandings, lapses, or inconsistencies, detracting from the effectiveness and utility of the system. Thus, ambiguities in the initial concept become contributory hazards, which should be avoided in NPP safety applications. Application and evaluation of HA (Section 3.1) is most effective in the concept phase of a system development lifecycle. Avoidance of the contributory hazards (see Table 1) requires much more rigorous description and control of the system concept and its relationship with its environment, as discussed in this section.

3.4.1 Hazards associated with the environment of the DI&C system

This Section focuses on the relationship of the conceived system with its environment; it also introduces hazard-avoiding system properties, which are applicable recursively to architecture inside the intended safety system. Section 3.4.2 elaborates on these properties. Section 3.4.1.3 focuses on hazards contributed through human-interaction aspect of the system's environment.

Hazards (including contributory hazards) may originate in the environment of the analyzed DI&C system, or may originate in the DI&C system, or may be the result of its interaction with its environment. See Appendix E.4 for hazards from the physical environment. See Appendix E.5 for ways in which a DI&C system may affect its environment adversely. Section 3.4.1.1 addresses hazards related to process monitoring. Section 3.2 addresses hazards contributed through the organizational processes and culture.

These considerations are applicable to architecture-related contributory hazards in every phase in the development lifecycle (from conception to implementation), to every level in the system architecture integration hierarchy, and to transformations from one level to another.

3.4.1.1 Hazards related to interaction with plant

Often, hazards arise from an inconsistency between the perceived process state and the real process state. Here, the term "process state" is used in the general sense, for example: the state of the nuclear reaction process, the state of some supporting physical process in the NPP, the state of control automation, the state of some instrument, or even the degradation process of some device. Hazards can also arise from unanalyzed conditions in the joint behavior of the plant (including equipment and processes) and the safety system. Table 4 shows examples of contributory hazards and conditions that reduce the respective hazard space.

[42] It roughly corresponds to but is significantly broader than the HA mentioned in [4] Table 1b.

[43] These are good candidates for discussion with the applicant before it submits the license application.

Table 4: Hazards related to interaction with plant: Examples

Contributory hazard		Conditions that reduce the hazard space	
ID H-Proc State -	**Description**	**ID H- Proc State -**	**Description**
1	The nature of change in some monitored physical phenomenon[44] monitoring the process of interest in the environment of the digital safety system is not well understood or not characterized correctly. Also see H-SR-23	1G1	The physical processes[45] in the monitored phenomenon are modeled and represented correctly; for example:
		1G1.1	Nature of variation over time
		1G1.2	Dependencies on other phenomena
		1G2	The perceived state matches reality with the fidelity required in value and time.
1.1	The temporal aspect of change in a continuously varying phenomenon is not well understood or not characterized correctly.	1.1G1	Temporal behavior of a continuously varying phenomenon is characterized correctly. such that timing requirements for monitoring it can be derived without loss of fidelity. This includes timing relationships across monitored phenomena,
		1.1G1.1	The physics of the phenomenon (e.g., dynamic behavior, including disturbances) is understood well and characterized mathematically.
1.2	The temporal aspect of change in a sporadic phenomenon is not well understood or not characterized correctly.	1.2G1	Requirements for reacting to sporadic events (e.g., sudden change) include the minimum inter-event arrival time, based on the physics of the event-generating process.
		1.2G2	Signal indicating event of interest is not filtered out.
		1.2G3	Signal indicating event of interest is not missed due to inadequate sampling, as determined through mathematical analysis.

[44] Examples: Pressure; temperature; flow; neutron flux density

[45] Examples: Energy-conversion; equipment degradation; component degradation

		1.2G4	ring event of interest does not disrupt any other action upon which a safety function depends.
2	Unanalyzed joint behavior of the safety system and the plant equipment and processes impairs a safety function.	2G1	Safety system and its environment, including the NPP equipment and processes are analyzed as a coupled system with sufficiently deep models of the behaviors (e.g., processes) to represent reality with fidelity.
3	Allocation of safety functions and properties from a system at a higher level of integration to one at a lower level, is not correct, complete or consistent, or is ambiguous.	3G1	Relationships with losses of concern identified at NPP level analysis and commensurate safety goals formulated in NPP level analysis are explicit.
		3G2	Decomposition of safety goals into required safety functions (design bases) is complete, correct, and consistent and unambiguous.
		3G3	Allocation of safety requirements to safety related systems[46]) is complete, correct, consistent and unambiguous. Also see Table 8.
		3G4	Allocation of safety properties, including corresponding decomposition or flow-down or derivation of constraints, is complete, correct, and consistent. See Section 3.5.1.1. Table 7.
		3G5	The boundary of the system being analyzed is well-defined with respect to its environment (in [10] CP 2.1.3.2.1).
		3G6	Interface to and interactions with the plant are specified and constrained in a manner that the system is assurable [H-S-1] (e.g., understandable [H-S-2↑], verifiable[47] [H-S-1.1], free from interference [H-S-3]). Examples of elements in the environment include interfaces to and interactions with: 1. Sensors 2. Actuators 3. Services needed; for example: 3.1. Electricity 3.2. Air flow

[46] If there are multiple levels of assembly (integration) this criterion applies to each level-pair.

[47] i.e., satisfaction of the constraint or specification is verifiable by analyzing the system concept.

			3.3. Compressed air 3.4. Water 4. Human-machine interfaces 4.1. Roles, responsibilities, functions. 4.2. Procedures specifying 4.1.
			Restrictions & constraints placed on the system are explicit; example constraints: 1. Compatibility with existing systems. 2. Physical and natural environment. 3. Protection against propagation of non-safety system faults and failures.
		3G7	Constraints on other elements in the environment of the system are explicit.
4	Interactions of the system with its environment, including effects of assumptions, are not well-understood. [H-ProcState-3↑] (In [13] Appendix A.3 item 3). [H-culture-12↓]	4G1	Constraints on other elements in the environment of the system are explicit.
		4G1.1	[H-culture-12G1↓] The organizational processes (Section 3.2) include explicit tasks or activities to validate each assumption in time to avoid adverse impact on the system safety properties and HA activities.
		4G1.2	If an assumption is found to be invalid or there is a change from the previous assumption: 1. There is a corresponding change impact analysis, maintained as an independently evaluated configuration item. 2. The affected part of the HA is repeated 3. Commensurate changes in constraints or requirements are identified. 4. There is an analysis of the impact of those changes. 5. The change impact analysis is an independently evaluated configuration item.
		4G2	Hazards from the physical environment are analyzed. See Appendix E.4
		4G3	Hazards from the DI&C system on its

			environment are analyzed. See Appendix E.5
5	**Unrecognized inter-dependencies in the system:** Inter-dependencies in the system, its elements, and its environment are not understood, recognized or explicitly identified, leaving some vulnerability, which can lead to the degradation of a safety function. [H0_8↑]	5G1	All inter-dependent systems, elements, processes, and factors affecting a safety function are identified.
		5G2	These are configuration items.
		5G3	The inter-dependencies or relationships among these items are unambiguously described, especially those affecting emergent behavior. [**H-ProcState-**5G1↑]
		5G4	Semantics of the relationships are explicit: Relationships may not merely be sequential (chained) or tree-structures, but also cycles – often feedback control loops[48]. [**H-ProcState-**5G1↑]
		5G5	The inter-relationships of these configuration items are identified (e.g., by means of an overall NPP-level system architecture). [**H-ProcState-**5G1↑]
		5G6	These inter-relationships are also a configuration item or set of configuration items. [H-ProcState-5G5↑]
		5G7	Independent verification assures that these configuration items represent reality. [**H0-**8.1G1↑]
		5G8	Effect of these dependencies is analyzed to prove that the safety function is not degraded.
		5G9	Any change in any of these configuration items is managed through a change control process, with a documented analysis of the impact of change. (Generalized from CP 2.7.3.1.5 in [14]) [**H-ProcState-**5G1↑]
		5G10	The change impact analysis is independently verified. [H-ProcState-5G8↑]
		5G11	The change impact analysis is a configuration item. [H-ProcState-5G8↑]
5.1	Dependencies through the environment of the digital safety	5.1G1	Effect of these dependencies is analyzed to prove that the safety

[48] Contrast with a chain of events initiated by failure of a hardware component

	system are not recognized; for example: • The physical processes • Degraded behavior of related instrumentation and peripheral equipment		function is not degraded.
6	**Unrecognized inter-dependencies in the development process:** Inter-dependencies in the system development process, feeder processes, supporting processes, elements, and environments, are not understood, leaving some vulnerability, which can lead to a defect in the system, which could lead to the degradation of a safety function. [H-0-9↑]	6G1	All inter-dependent processes (including feeder and supporting processes), resources used in these processes and factors affecting these processes and resources are identified (e.g., see Figure 4).
		6G2	These are configuration controlled items (henceforth, configuration items). [**H-ProcState-6G1**↑]
		6G3	The inter-dependencies or relationships among these items are unambiguously described, including cycles created through feedback loops[49]. [**H-ProcState-6G1**↑]
		6G4	The inter-relationships across these configuration items are identified (e.g., by means of an overall process architecture), and are also a configuration item or set of configuration items. [**H-ProcState-6G1**↑]
		6G5	Some combination of independent assessment, audit, and verification assures that these configuration items represent reality. [**H-ProcState-6G1**↑]
		6G6	Any change in any of these configuration items is managed through a change control process. [**H-ProcState-6G1**↑]
		6G7	Effect of these dependencies is analyzed to prove that the safety function is not degraded.
7	Dependencies through supporting services and processes are not recognized	7G1	Effect of these dependencies is analyzed to prove that the safety function is not degraded.
8	Dependencies through resource[50] sharing are not recognized; examples:	8G1	Effect of resource-sharing is analyzed to prove that the safety function is not degraded.

[49] These can also be analyzed as control loops influencing safety properties of the affected system.

[50] Examples: Skilled resources for development; Computing memory or processor-time during execution.

	• Contention for the shared resource • Corruption of resource (e.g., data)		

The following notes explain certain contributory hazards identified in Table 4

H-ProcState-{3-4}: The intent of reviewing for these factors is to check that the system on which HA is to be performed and its context (environment) are correctly identified, the dependencies are correctly understood, the primary hazards (external and internal) are identified, and the commensurate constraints are identified.

H-ProcState-3: When a large complex system, such as an NPP (including its environment and processes for operation and maintenance) is decomposed into manageable subsystems and components, the constraints necessary to prevent the losses at the top level (e.g., NPP-level) may become obscure. For example, subtle couplings across the decomposed elements might arise. In an evolving configuration of the overall (e.g., NPP-level) system, the boundary of the system being analyzed and assumptions about its environment may not be well-defined, leading to appropriate considerations "falling through the cracks."

H-ProcState-{4-7}: Whereas "ineffective hazard recognition" has been recognized as a serious issue [6], unrecognized dependencies are an increasing contributor to this issue, as the complexity of organizations, processes, and systems is increasing. In addition to the lack of awareness, lapses could occur because of inability to track and maintain a consistent understanding of the dependencies[51].

Figure 5 depicts a "gradual[52]" migration from normal operational process state region (shown in green) to an unsafe state region (shown in red). Actions to avoid the unsafe state region (i.e. to effect safe recovery) need some time (shown as the brown region). To allow for the needed time, the temporal aspect of change in the monitored phenomena must be understood well and departure from normal operational state (shown in yellow), monitored.

[51] The state of practice in representing and analyzing such dependencies is relatively weak, as discussed in Appendix C..

[52] Premise: Degradation is not sudden or unpredictable, and progression can be monitored.

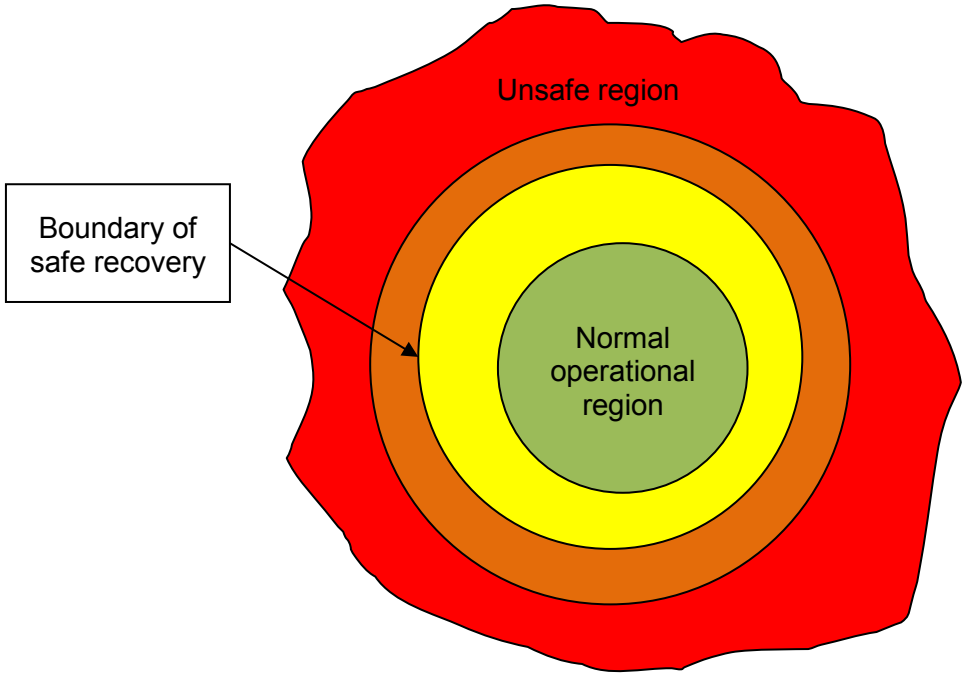

Figure 5: Regions of state space for hazard analysis

3.4.1.2 Contributory hazards from NPP-wide I&C architecture

The scope of NPP-wide system architecture includes the safety system under evaluation and its relationship with its environment, that is, all systems, elements, processes and conditions that support or affect the performance of a safety function. "Relationship" includes interfaces, interconnections, and interactions, whether these are direct, intended, explicit, static, "normal," indirect, implicit, unintended, dynamic, or "abnormal." HA of the NPP-wide I&C architecture should examine it for hazards relevant to the safety related system to be analyzed. Figure 6 provides a simplified view.

Constraints on the NPP-wide I&C architecture are derived from the quality[53] attributes or properties of the safety related system being analyzed. Quality attributes are discussed in Section 3.5.1.1, including Table 7, which also applies to the NPP-wide I&C architecture.

> Note: Criteria for the HA-evaluation the NPP-wide architecture are predicated on the correct and complete performance of HA, as illustrated in Table 1, including considerations of combinations of multiple contributory hazards, exemplified through Table 4, Table 2, Table 3, Table 5, and Table 6.

[53] Other terms for these properties: Quality-of-service (QoS) properties; non-functional requirements

Table 12, derived from considerations in Table 7, also applies to the NPP-wide I&C architecture. in the context of hazards contributed through interference.

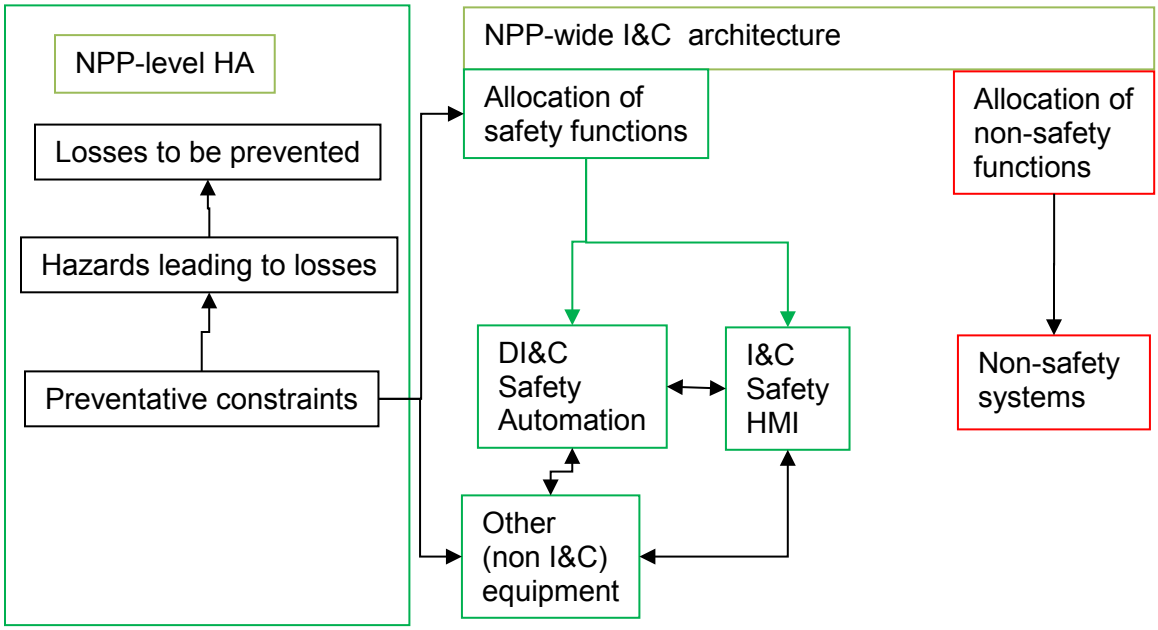

Figure 6: NPP-wide I&C architecture - allocation of functions in concept phase

3.4.1.3 Contributory hazards from human machine interactions

Hazards of the kind grouped in Table 1-Table 4 could also affect human-automation interactions

Table 5 supplement those with some examples of more specific hazards contributed through human-automation interactions and Table 6, those through inadequacies in the associated engineering.

Table 5: Contributory hazards from human machine interactions: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-hmi-	**Description (**e.g., Scenario**)**	**ID H-hmi-**	**Description**
1	Inconsistency between human-perceived process state and real process state	1G1	Process state presented to the human represents the real physical state in value and time.
2	Inconsistency between human-perceived state of an instrument and real state of the instrument	2G1	Instrument (e.g., actuator) state presented to the human represents the real physical state of the instrument in value and time.
3	Mode confusion	3G1	Human is notified of the current mode and a mode change in progress (the loop is closed with feedback).
		3G2	Human has a correct understanding of the mode-change model (human is equipped with correct mental model of the mode-switching behavior of the automation)
		3G3	Potential for mistaken interpretation of the information presented by the human-machine interface is eliminated.
		3G4	Inconsistent behavior of automation is avoided; or, automation detects its inconsistency and notifies human.
		3G5	Unintended side effects are avoided
3.1	Confusion about line of authority (who or what entity is in control at the moment)	3.1G1	Multiple concurrently active paths of control authority (logical control flow) are avoided
		3.1G2	Change of mode by automation without human confirmation is avoided.
		3.1G3	Correct division of tasks is ensured through analysis of human tasks, including human-automation interactions.
4	Inappropriate division and allocation of tasks between human and automation.	4G1	H-OTproc-3G1

5	Normally useful cognitive processes are defeated or fooled by a particular combination of conditions [18]	5G1	

Table 6: Contributory hazards from human machine interaction engineering: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-hmiP-	**Description (**e.g., Scenario**)**	**ID H-hmiP-**	**Description**
1	Loss of information across disciplines (e.g., automation engineering, human factors engineering, control room design). [H-culture-9↑][H-SR-3↑]	1G1	System is engineered holistically, including crosscutting analysis. (Adapted from [13] Appendix A.3 footnote 82)
2	Confusing human-machine interface design	2G2	H-hmi-3G3
3	Cognitive overload	3G3	H-OTproc-3G1

3.4.2 Contributory hazards in conceptual architecture

The term “conceptual architecture” refers to the architecture of the system concept, as it evolves in relation to its environment (also see Sections 3.4.1.2).

Here, the focus shifts from the interactions of the conceived system with the environment to its internal architecture, as driven by the requirements allocated to it, that is, the inter-relationships of the various requirements and constraints to be satisfied by the conceived system. The information in Table 7 and Table 12 is applicable to the conceptual architecture, especially with respect to the following concerns:

1. Freedom from interference across redundant divisions [Table 12 H-S-3G3 - 2↑].
2. Freedom from interference between a monitoring element and its monitored element [Table 12 H-S-3G3 - 4↑].
3. Compromise of redundancy through a dependency (e.g., input data; resource-sharing). Also see Table 1 items 5 and 8-10.
4. Compromise of redundancy in the concept of voting[54] logic.

The conditions (to reduce the respective hazard spaces) provided in Table 7 and Table 12 apply recursively to the finest grain level of the system architecture and recursively to the finest grain level of the software architecture. These conditions also apply to the mappings from one level to

[54] Example: In a quad-redundant system for a space system, four computers were connected by a multiplexor/de-multiplexor module. A diode in the interconnections failed in an unanticipated way, such that the condition was not observed by the 4 computers similarly. (In [13] Appendix A.3 footnote 84)

another in the architecture hierarchy[55] and through all stages of derivation of requirements & constraints and the subsequent development lifecycle stages.

3.4.3 Contributory hazards from conceptualization processes

Examples of hazards contributed through weaknesses in the cultural and general technical processes of the organization (Table 2 and Table 3), which were introduced in Section 3.2, apply to the concept phase of the system development lifecycle strongly.

Requirements engineering (Section 3.5) and architectural engineering (Section 3.6) apply to the concept phase also – see Table 11 and Table 12 - Table 13 for the respective hazard groups.

3.5 Evaluation of contributory hazards in Requirements Engineering

Identifying valid requirements for the digital safety system is one of the weakest links in the overall process. Inadequacy in requirements is one of the most common causes of a system failing to meet expectations. Failures traceable to shortcomings in requirements cannot be caught through such verification activities as simulation and testing alone. Formal methods do not help in understanding intent or eliciting missing requirements, when the intent is not clear [13].

3.5.1 System Requirements

In the general context of systems engineering, the specification of a primary function, valued and required by its user, is called a functional requirement. In the context of digital safety systems, example groups of functional requirements include (but are not limited to) monitoring departure from a safe state, detecting threshold for intervention, and intervention for mitigating the consequence of departure from safe state. Key prerequisite activities for identifying safety requirements were discussed in Sections 3.1 (overall hazard analysis, understanding dependencies leading to loss events), 3.4.1 (understanding hazards in relation to the environment of the safety system), including hazards contributed from inadequate definition of the boundary of the safety system, invalid assumptions, and interactions with other systems and humans). The analysis covered in those sections should be viewed as an early stage of requirements engineering. Given the requirements resulting from those analytical activities, Section 3.5.1.1 introduces the concept of associated quality requirements, also known as quality attributes or quality-of-service (QoS) requirements or QoS properties. - Section 3.5.1.1 also introduces the concept of derived quality characteristics or requirements in an organizing framework, known as a "quality model" [22]. Section 3.5.1.2 identifies some common weaknesses in formulating verifiable requirements, and Section 3.5.1.3 identifies some common weaknesses in the associated requirements engineering processes.

3.5.1.1 Quality requirements

Figure 7 shows quality[56] requirements associated with functional requirements. In the context of this RIL, examples of top-level quality requirements are Safety and Security.[57]

[55] The mapping could contribute a hazard, e.g., Some abstractions can mask problems.

[56] More popularly, it is known as a non-functional requirement

[57] Other examples of quality requirements: Maintainability, Reconfigurability, Portability.

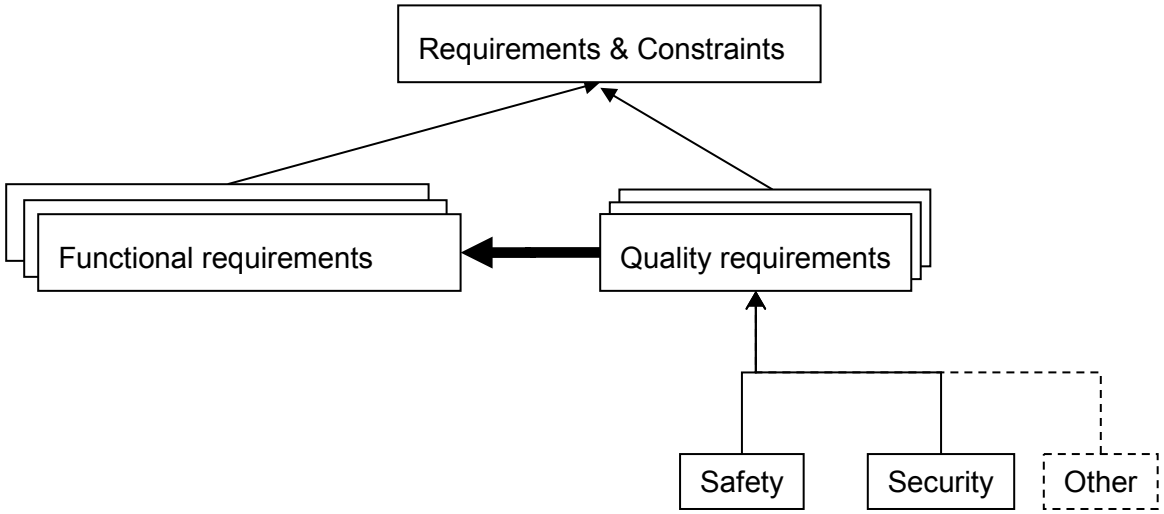

Figure 7: Quality requirements should be explicit

For a safety system, as shown in Figure 8, the "Assurability" property distinguishes it from systems that are not safety-related. Figure 8 also shows other quality attributes that contribute to or support "Assurability." The corresponding quality requirements may also be viewed as constraints to be satisfied by the digital safety system, that is, constraints on the solution space, such that system concepts that do not satisfy these constraints are eliminated from further consideration. Table 7 shows the logical derivation of these constraints (with the derivation relationships shown in Figure 8) to support the "Assurability" property with the following (informally expressed) reasoning:

1. To be able to assure that a system is safe, one must be able to verify[58] [H-S-1] that it meets all its safety requirements.
2. For a system to be verifiable, it should not be possible for one element of the system to interfere with another. [H-S-3]
3. If the conceived system is too complex, adequate verification is infeasible. [H-S-1.1]
4. If one cannot even understand it, how can one assure that it is safe? [H-S-2]
5. Verification also includes analysis at various phases in the development lifecycle, well before[59] an artifact is available for physical testing. Analysis may take various forms:

    5.1. Quantitative
        5.1.1. Numerical (e.g., analysis of a continuous control algorithm)
        5.1.2. Logical

[58] Verification includes testing.

[59] When performed on a computer program (code), it is known as static analysis. However, analysis in the same "static" sense can also be performed on work products of earlier phases, e.g. on models. [H-S-1.1.1]

5.1.3. Mechanized, but requiring manual interventional activities

5.2. Qualitative, but consistently[60] repeatable across comparably qualified performers.
5.2.1. Mechanizable
5.2.2. Human-dependent

6. To satisfy the property “analyzability”:
6.1. The system must have predictable and deterministic behavior. [H-S-1.2]

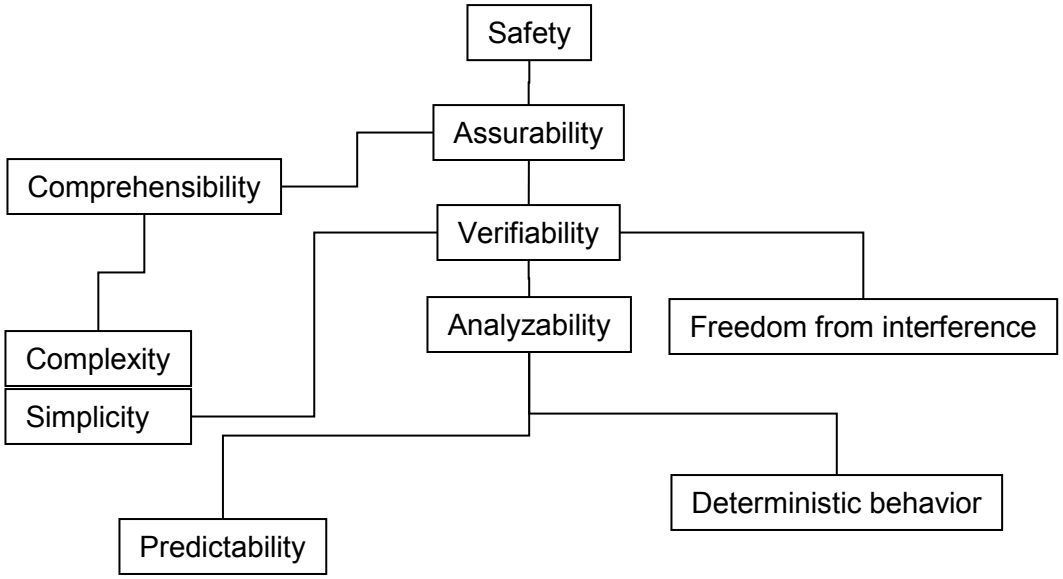

Figure 8: Quality characteristics to support safety

Table 7: Constraints derived from quality attributes: Scenario-based examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-S-	**Description (**e.g., Scenario**)**	**ID H-S-**	**Description**
1	Scenario: Safety cannot be assured, because the system is not sufficiently verifiable and understandable. Appropriate considerations and criteria are not formulated at the beginning of the development lifecycle; therefore, corresponding architectural constraints are not formalized and checked. When work products are available for testing, it is discovered that adequate testing is	1G1	Verifiability is a required system property, flowing down from the system to its constituents progressing to the finest-grained element. (Adapted from CP 2.2.3.11 in [14])

[60] If the analysis is not consistently repeatable or the analysis method/tool itself is not adequate, the purpose of this RIL treats the system as un-analyzable.

Contributory hazard		Conditions that reduce the hazard space	
ID H-S-	**Description (**e.g., Scenario**)**	**ID H-S-**	**Description**
	not feasible (e.g., the duration, effort, and cost are beyond the project limitations).		
1.1	System is not verifiable (e.g., it is not analyzable or very difficult to analyze).	1.1G1	Avoidance of unnecessary[61] complexity
		1.1G1.1	The behavior is unambiguously specified, and this constraint flows down to the finest-grained element in the system (in [13] Appendix A.4 item 4). The flow-down ensures that • Allocated behaviors satisfy the behavior specified at the next higher level of integration; • Allocated behaviors do not specify or allow any unwanted behavior.
		1.1G1.2	The behavior of the system is a composition of the behaviors of its elements, such that when all the elements are verified individually, their compositions may also be considered verified[62]. This property is satisfied at each level of integration, flowing down to the finest-grained element in the system.
1.1.1	There are unanalyzed or un-analyzable conditions. For example, all system states, including unwanted ones such as fault states leading to these, are not known and not explicit. To that extent, verification and validation (V&V) of the system is infeasible. [H-S-1.1↑]	1.1.1G1	Static analyzability: System is statically analyzable. 1. All states, including unwanted ones, are known. 2. All fault states, leading to failure modes, are known (in [14]CP 2.2.3.14 1st item) 3. Safe state space of the system is known (in [14] CP 2.2.3.14 2nd item)
1.1.2	There is inadequate evidence (e.g., a verification plan) of verifiability. [H-S-1.1↑]	1.1.2G1	Verification plan shows the coverage needed for safety assurance
1.2	System behavior is not deterministic. [H-S-1.1.1↑]	1.2G1	System has a defined initial state.

[61] Unnecessary: [Complexity] that is not essential to support a safety function.

[62] No unpredictable behavior emerges.

Contributory hazard		Conditions that reduce the hazard space	
ID H-S-	**Description (**e.g., Scenario**)**	**ID H-S-**	**Description**
		1.2G2	System is always in a known configuration.
		1.2G3	System is in a known state at all times (e.g., through positive[63] monitoring and indication): 1. Initiation of function 2. Completion of function (in [14] CP 2.1.3.4 last item) 3. Intermediate state, where needed to maintain safe state in case of malfunction.
1.3	System behavior is not predictable. [H-S-1.1.1↑]	1.3G1	Each transition from a current state (including initial state) to some next state is known
		1.3G2	A hazardous condition can be detected in time to maintain the system in a safe state. (in [14] CP 2.2.3.14 3rd item)
2	Comprehensibility: System behavior is not understood completely and correctly by its community of users (e.g., reviewers, architects, designers, and implementers), that is, the people and the tools they use. [H-S-1↑]	2G1	Behavior is completely specified
		2G2	Behavior is completely understandable.
		2G3	Behavior is understood completely, correctly, consistently, and unambiguously by different users interacting with the system.
		2G4	The allocation of requirements to some function and that function to some element of the system is bi-directionally[64] traceable. (in [13] Appendix A.4 item 2)
		2G5	The behavior specification avoids mode confusion, esp. when functionality is nested (in [13] Appendix A.4 item 3)
		2G6	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), that is, the people and the tools they use (in [13]

[63] If indirect indication or inference is used, HA confirms satisfaction of H-ProcState-1G1.2.

[64] It is not implied that one-to-one relationships are necessary.

Contributory hazard		Conditions that reduce the hazard space	
ID H-S-	**Description (**e.g., Scenario**)**	**ID H-S-**	**Description**
			Appendix A.4 item 9)
Note: A symbol of the form [H-S-1↑] indicates that this item "supports" or is derived from the linked item (e.g., H-S-1)			

Considering that the state of practice is especially weak in the derivation of verifiable constraints from QoS properties, a careful review is needed. The architecture should satisfy these constraints, starting from the system concept phase and continuing at every successive phase of development, refinement and decomposition, including all phases of the software development lifecycle. Supporting architectural constraints are identified in Section 3.6.

3.5.1.2 Contributory hazards through inadequate system requirements

Activities leading to identification of functional requirements for safety were introduced in Sections 3.1 (overall hazard analysis, including understanding dependencies leading to loss events), 3.4.1 (understanding hazards in relation to the environment of the safety system), including hazards contributed from inadequate definition of the boundary of the safety system, invalid assumptions, and interactions with other systems and humans). Table 8 identifies further contributory hazards due to weaknesses in identifying and formulating requirements. The content of Table 8 is adapted mostly from Appendix A.3 in [13]; other sources are referenced within the respective item in Table 8. For hazards contributed through weaknesses in interfaces and interactions across elements of the system, see Section 3.6.1.

Table 8: Hazards contributed through inadequacy in system requirements: Examples

Contributory hazard			Conditions that reduce the hazard space
ID H-SR-	**Description (**e.g., Scenario**)**	**ID H-SR-**	**Description**
1	Mistakes occur due to misunderstanding the environment	1G1	[H-SRE-{1G1, 1G2, 1G3}↓]
2	Input constraints misunderstood or improperly captured [H-SR-1↑]	2G1	[H-SRE-{1G1, 1G2, 1G3}↓]
		2G2	Criteria for input validation are correctly established.
3	Incompleteness	3G1	See Table 1
		3G2	
		3G3	Scope of HA includes interactions with the environment of the system – see Section 3.4.1.
		3G4	Inter-relationships and interactions with the environment are analyzed in all configurations and modes (including degraded ones), and changes from one to another. [H-SR-3G3↑]

		3G5	In HA at system concept phase (Section 3.4), an architectural model or representation of the system (e.g., functional; behavioral) concept includes a (functional; behavioral) model or representation of the environment, especially the physical processes (Appendix H) [17]. [H-SR-3G3↑] [H-SAE-{1G1, 2G1, 3G1, 4G1}↓]
		3G6	Process behavior models[65] (H-SR-3G5) include identification of safe state regions and trajectory[66] of safely recoverable process state. [17], See Figure 5
		3G7	Process behavior models (H-SR-3G5) include time-dependencies, relationships and constraints. [18] [H5-G0]
		3G8	[H-SRE-{1G1, 1G2, 1G3}↓]
4	Inadequate provision of resiliency and robustness[67] against residual faults[68]. [H-SR-3↑] Note tension with [H-SR-20]	4G1	Monitoring: Feasible trajectories[69] of appropriate state variables[70] or parameters and expected values are known and monitored. (Generalized from [14] CP 2.1.3.2.3, 2.1.3.2.4)
		4G2	Detection: Requirements are formulated to monitor system or element parameters for anomaly (e.g., by applying the discriminating[71] logic on the monitored parameters) in conjunction with predictive behavior models, but considering [H-SR-{19, 20}]
		4G3	Intervention: Corresponding requirements are formulated for intervention to maintain plant in safe state. (Adapted from [14] CP 2.2.3.7)
		4G4	Containment: Ability to locate and isolate or contain the source of the fault (e.g., a hardware or software component).
		4G5	Notification: Notification is timely, but avoids "flooding.
		4G6	[H-SRE-{1G1, 1G2, 1G3}↓]
5	Inadequacy in identifying sources of uncertainty, their effects, and their mitigation. [H-SR-3↑]	5G1	[H-SRE-{1G1, 1G2, 1G3}↓]

[65] The scope is limited to I&C-relevance.

[66] State space within which recovery is provable.

[67] Robustness is identified as a functional characteristic in BTP 7-14

[68] Context: Internal to digital safety system and its elements

[69] For example: Values over time; rate of change.

[70] Include inputs and outputs.

[71] e.g. through infeasible or unexpected value.

6	A fault propagates due to deficiency in requirements for fault containment. [H-SR-3↑]	6G1	[H-SRE-{1G1, 1G2, 1G3}↓]
7	Inadequate or improper generalization to capture classes of issues	7G1	[H-SRE-{1G1, 1G2, 1G3}↓]
8	Inconsistency	8G1	[H-SR-15↓] [H-SRE-{1G1, 1G2, 1G3}↓]
9	Invalid input see H_SR_2G2	9G1	Validity of value of each input is monitored (in [14] CP 2.1.3.2.4).
		9G2	Intervention upon detecting invalid input is specified to maintain system safe state.
10	Uncorrected or inadequately instrumentation errors	10G1	Required calibrations and corrections are known and applied (in [14] CP 2.1.3.2.5)
11	Effects of invalid inputs.	11G1	[H-SRE-{1G1, 1G2, 1G3}↓]
12	Implicit assumptions about the environment. Invalid assumption about the environment.	12G1	Each assumption about the environment is made explicit (e.g., documented; in [13] Appendix A.3 item 3). H-0-{5G1.1, 5G1.2}↑
		12G2	Each assumption about the environment is validated (e.g., as a “constraint or condition to be validated).”
13	Unclear understanding of the consequences of	13G1	With each assumption [H-SR-12G1], include the consequences if the assumption turns out to be false. (In [13] Appendix A.3 item 4)

	an assumption [Table 1][H-SR-12↑]	13G2	Requirements include measures to mitigate the consequences of assumptions that fail to hold. (In [13] Appendix A.3 item 4)
		13G3	Each assumption (e.g., constraint or condition to be validated) is tracked as a configuration item.
		13G4	Assumptions about the downstream design are made explicit (e.g., through explicit derived requirements or constraints on the architecture, design and implementation, and the associated methods and tools). (In [13] Appendix A.3 item 3.1). Examples: 1. Requirements from the application software on system platform services (HW & SW), including HW and SW resources to support the workload. 2. Timing constraints to be satisfied. 3. Compatibility across maintenance updates.
		13G4.1	The safety plan and supporting plans include activities and tasks specifying how and when these assumptions will be validated.
14	Unmitigated consequence of invalid assumption	14G1	With each assumption [H-SR-12G1], include how and when it will be validated. (In [13] Appendix A.3 item 3)
15	Incorrect order of execution or timing behavior [H-ProcState-1.3]	15G1	An explicit, verifiable (as determined through mathematical analysis) specification for the order of execution and timing inter-relationships, especially considering multiple concurrent physical processes, inter-process synchronization and shared resources (in [14] CP 2.1.3.2.2, 2.2.3.5).

16	Inter-relationships and inter-dependence across requirements are not clearly understood or recognized [H0-4 – H0-8], resulting in unanalyzed conditions	16G1	Applicable types of dependencies across requirements are identified (see examples herein), modeled, and tracked. For example, if A and B are two requirements, their relationship types may be: • A requires B • B supports A • B hinders A • B is a selection for A (an exclusive one among many choices) • B is a specialization of A Note: Relationships may be one-to-one, one-to-many, many-to-one, and many-to-many.
17	Interference from unintended (including unwanted) functions or side effects. [H-S-1↑]	17G1	Interactions are limited provably[72] to those required for the safety functions. Absence of other unintended functions or side effects is validated.
18	Effects of sudden hardware[73] failure, esp. semiconductors	18G1	Requirements include failure or fault detection and containment measures, including offline ability to locate and isolate the source of the fault (e.g., a hardware or software component). [H-SRE-7G1↓]
19	Functional requirement set[74] leads to conditions that are unanalyzable or difficult to analyze.	19G1	[H-SRE-{1G1, 1G2, 1G3}↓]
20	Adding backups (or fault protection) can introduce new hidden dependencies and impair analyzability. [H-SR-19↑]	20G1	[H-SRE-{1G1, 1G2, 1G3}↓]

[72] Unwanted functions are not allowed.

[73] Also see Table 16

[74] The bigger your hazard analysis the less useful it is (in [13] Appendix A.3 footnote 78).

21	Although layered protection has benefits, there can be dilemmas from keeping software protected with several layers – analyzability may be impaired. [H-SR-19↑]	21G1	[H-SRE-{1G1, 1G2, 1G3}↓]
22	Inability to integrate correctly elements of a system (e.g., subsystems, hardware components, software components). [H-SR-{1, 2, 3, 8, 12, 13, 15, 16, 19↓] [H-SwR-2↓] [H-SRE-7↓] [H-SwRE-1↓] [H-HwP-1↓]	22G1	[H-SRE-{1G1, 1G2, 1G3}↓] [Table 13↓]
23	Anomaly in the state of the process is not recognized or identified or correctly understood or correctly specified. [H-SR-3↑] [H-SR-4↑]	23G1	See H-SR-3G6 The trajectory of safely recoverable process state variables (i.e., state space within which recovery is provable) is specified correctly. In other words, when departure from this state space or region is recognized, intervention can prevent departure from safe state. See Figure 5.
Notes: A symbol of the form [H-culture-8↑] indicates that the item "supports" or is derived from the linked item H-culture-8 A symbol of the form [H-SRE-7G1↓] indicates that the item requires, spawns, or "is supported by" the linked item, H-SRE-7G1			

3.5.1.3 Contributory hazards from system requirements engineering

The requirements engineering phase of the lifecycle is most sensitive to the quality of processes and resources applied. Requirements elicitation and analysis aspects are most sensitive to the competence [H-SRE-1] applied.

Table 9 identifies hazards contributed through weaknesses in the process of engineering requirements for the system.

Table 9: Hazards through inadequacy in system requirements engineering: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SRE-	**Description (**e.g., Scenario**)**	**ID H-SRE-**	**Description**
1	Inadequate competence [H-SR-1-7↑]	1G1	The team engaged in these activities is an assemblage of high competence in multiple disciplines, capable of creatively eliciting and synthesizing information from diverse sources, including implicit, experiential knowledge about the environment. The combined competence of the team matches the expertise needed in each phase in the engineering lifecycles, noting that the nature of expertise is not the same in all phases.
		1G2	A different and independent diverse team reviews the requirements and their validation.
		1G3	The review team has expertise in discovering the types of mistakes or shortcomings identified in Table 8 and Table 9 H-SRE-{2-6}
2	Ambiguity in the natural language textual description [H-SAE-2↓]	2G1	A subset of the natural language is used such that requirements can be described unambiguously to the community of its users[75]; for example: 1. Closed set of language elements 2. Unambiguous semantics of each language element 3. Unambiguous compositions of language elements and their compositions [H-SAE-1G1↓; H-SAE-1G2↓]

[75] Users include people and tools, employed in creation, modification, interpretation, transformation. maintenance, V&V, and regulation (adapted from CP 2.3.3.1.1 last sentence in [10]).

		2G2	The language subset (H-SRE-2G1) supports distinct identification and description of the following: 1. Assumptions about the environment [17] 2. Input from the environment (e.g., command, that is, some signal requiring state-changing effect + required behavior), query, process state, other data 3. Output (e.g., some signal having state-changing effect, state-notification, exception-notification) 4. Functions assigned to a human 5. Procedure for the execution of each function assigned to a human (required behavior) 6. Other elements of the system 7. Functions assigned to each element; required behavior 8. Interactions required across elements 9. Constraints on the behavior and interactions of each element, e.g. timing constraints [18]; QoS constraints 10. Criteria to monitor and detect violation of a constraint [16]
3	Incorrect formalization from intent or natural language text	3G1	[H-SRE-2G1↓; H-SRE-2G2↓] [H-SAE-1G1↓; H-SAE-1G2↓]
4	Input constraints are ambiguous	4G1	Valid value type and range of each input are explicitly identified (in [14]CP 2.1.3.2.4). Also see Table 1.
5	Loss of information in transfer and traceability of HA-results to requirements	5G1	Activities of HA and Requirements Engineering are formally integrated (also see Table 1)
6	An atomic requirement is not traceable individually	6G1	Each atomic requirement is traceable (in [10] CP 2.1.3.1; in [13] Appendix A.4 item 2) [H-S-2G4↓]
		6G2	Each requirement is a configuration controlled item[76].

[76] Other relevant references: IEEE 828 and 1042

7	Loss of information[77] across disciplines, processes, and organizational units (e.g., system engineering, software engineering, hardware engineering, safety, quality). [H-culture-9↑][H-SR-3↑] [H-SwRE-1↓]	7G1	Engineer systems holistically, including crosscutting analysis. (Adapted from [13] Appendix A.3 footnote 82)
			H_culture_12G1
Notes: A symbol of the form [H-culture-8↑] indicates that the item "supports" or is derived from the linked item (e.g., H-culture-8) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

3.5.2 Software Requirements

Contributory hazards and corresponding constraints and conditions identified for system requirements also apply to software requirements. Even though correct, complete, consistent unambiguous requirements for software are supposed to flow down from the system engineering lifecycle, typically in practice, V&V for these properties occurs from the software engineering perspective[78] as a part of the software engineering lifecycle.

Some of the requirements from the system engineering lifecycle may be allocated directly (as is) to software. For other requirements from the system engineering lifecycle (e.g., QoS requirements) additional requirements and constraints for software may be derived as part of the software engineering lifecycle. Also see Section 3.6 for constraints on software architecture. Contributory hazards and constraints identified in Section 3.6.1 for the system architecture also apply to software. Derived constraints on software design and implementation (D&I) are included in Sections 3.8 and 3.9.

3.5.2.1 Contributory hazards in software requirements

The contributory hazards identified in Table 8 also apply to software requirements. Table 10 provides examples of additional hazards contributed through weaknesses in software requirements.

[77] Current practice divides systems engineers, software engineers, and hardware engineers; often failures occur due to gaps in between. (From [13] Appendix A.3 footnote 82)

[78] Focus: Check correctness of understanding; make explicit and unambiguous.

Table 10: Hazards contributed through inadequacy in software requirements: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SwR-	**Description (**e.g., Scenario**)**	**ID H-SwR-**	**Description**
1	Inadequate flow-down of QoS properties (Table 7) and other constraints from the system engineering lifecycle (Table 8) [H-SwR-2↓]	1G1	Corresponding constraints are derived and applied to software (Table 7; Table 8)
2	Inadequate flow-down of requirements & constraints to support integration of elements into a correctly working system.	2G1	Corresponding constraints are derived and applied to software
3	Software produces an output of infeasible value	3G1	Identify infeasible conditions in the real world and use these to establish requirements that monitor anomalous[79] behavior of software. (Adapted from [14]CP 2.3.3.1.5) , but considering [H-SR-{19, 20}]
Notes: A symbol of the form [H-culture-8↑] indicates that the item "supports" or is derived from the linked item (e.g., H-culture-8) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

[79] Intent: Defend against weakness in requirements.

3.5.2.2 Contributory hazards from software requirements engineering

The contributory hazards identified in Table 9 also apply to software requirements engineering. Table 11 provides examples of additional hazards contributed through weaknesses in engineering software requirements.

Table 11: Hazards through inadequacy in software requirements engineering: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SwRE-	**Description** (e.g., Scenario)	**ID H-SwRE-**	**Description**
1	Loss of information across disciplines, processes, and organizational units (e.g., system engineering, software engineering, hardware engineering, safety, quality) due to discipline-wise division of organizations, people, and work [H-culture-9↑]	1G	H-SRE-7G1↑
2	Loss of information across disciplines due to incompatible paradigms, methods, and tools across disciplines. [Example contributor: H-HwP-5↓]	2G	Methods and languages to describe or specify requirements allocated to software support unambiguous mapping and integration across dissimilar elements (e.g., interactions across hardware, software and human elements). [H-SAE-{2G1, 3G1}↑] [H-HwP-5G1↓]
Notes: A symbol of the form [H-culture-9↑] indicates that the item "supports" or is derived from the linked item H-culture-9. A symbol of the form [H-HwP-5G1↓] indicates that the item spawns the linked item H-HwP-5G1			

3.6 Evaluation of contributory hazards in Architectural engineering

System failures traceable to architecture rank high in the experiences of various safety-critical, mission-critical, high-quality DI&C systems. For example, unwanted and unnecessary interactions, hidden couplings and side effects have led to unexpected failures; traditional testing or simulation based verification did not detect such flaws [13].

3.6.1 Contributory hazards in System Architecture

While the overall scope of system architecture includes the safety system under evaluation and its relationship with its environment, this section focuses on system-internal elements (e.g., hardware and software) and their inter-relationships (i.e., interfaces, interconnections, and interactions) whether these are direct or indirect, intended or unintended, explicit or implicit, static or dynamic, "normal" or "abnormal."

The scope of system architecture activities includes the allocation of requirements and constraints to elements identified in the system architecture.

> Note: Criteria for the HA-evaluation for the architecture are predicated on the correct and complete performance of HA, as illustrated in Table 1, including considerations of combinations of multiple contributory hazards, exemplified through Table 4, Table 2, Table 3, Table 5, and Table 6.

Evaluation of contributory hazards in system architecture Table 7 and Table 12 summarize significant contributory hazards related to architecture, and provide examples[80] of corresponding conditions that reduce the respective hazard spaces. These considerations are applicable to architecture-related contributory hazards in every phase in the development lifecycle (from conception to implementation), to every level in the system architecture integration hierarchy, and to transformations from one level to another.

Table 12: Interference: Example scenarios and conditions that reduce the hazard space

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SA-	**Description (**e.g., Scenario**)**	**ID H-SA-**	**Description**
3	Scenario: Safety cannot be assured, because a system, device, or other element (external or internal to a safety system) may affect a safety function adversely through unintended (including unwanted)	3G1	[H-SR-17G1↑]
		3G2	Interactions and interconnections that preclude complete[81] V&V are avoided, eliminated, or prevented. (CP 2.2.3.11 in [14])
		3G3	Freedom from interference is assured[82] provably[83] across:

[80] Neither the contributory factors nor the preventative constraints are enumerated exhaustively.

[81] This condition leads to "independence" and "simplicity" constraints – see in glossary note concerning influence of complexity on verifiability.

	interactions - possibly with contribution from one or more hazards (e.g. defects, deficiencies, disorders, malfunctions, oversights). [H-SR-17↑]		1. Lines of defense 2. Redundant divisions of system (CP 2.2.3.6 in [14] 3. Degrees of safety qualification[84] (CP 2.2.3.3 in [14]) 4. Monitoring & monitored elements of system.
		3G4	Analysis of the system demonstrates that unintended (including unwanted) behavior is not possible[85]. 1. Interaction across different sources of uncertainty is avoided. 2. The architecture precludes unwanted interactions and unwanted, unknown hidden coupling or dependencies (in [13] Appendix A.4 item 6). 3. Specified information exchanges or communications occur in safe ways (in [13] Appendix A.4 item 6).
		3G5	Only well-behaved interactions are allowed [H-S-1.2G{1,2,3}, H-S-1.3G{1,2}↑]
		3G7	The impact of change is analyzed to demonstrate no adverse effect. [Table 1]
4	Scenario [H-S-3G4↑]: A function, whose execution is required at a particular time, cannot perform as required, due to interference through sharing of some resource it needs.	4G1	Analysis of the execution-behavior of the system should prove that such interference will not occur. For example, worst-case execution time is guaranteed.
5	Timing constraints are not correctly specified and not correctly allocated.	5G1	Timing requirements for monitoring a continuously-varying phenomenon are derived, specified, and allocated correctly to the services and elements upon which their satisfaction depends. Example: Sampling interval that characterizes the monitored variable with fidelity.
		5G1.1	Commensurate required sampling interval is determined through mathematical analysis.
		5G1.2	Discretization and digitization do not affect the fidelity required, as determined through

[82] This condition leads to the independence criterion, stated as a “principle” by the ACRS I&C Subcommittee chairman.

[83] Example: There is no pathway.

[84] In other application domains, the corresponding concept is known as “mixed criticality.”

[85] Example: There is no pathway.

			mathematical analysis.
		5G1.3	Aliasing is avoided.
		5G1.	Sampling periods to monitor discrete events are established correctly, as determined through mathematical analysis.
6	Sampling and update intervals are not commensurate to the timing constraints of the associated control actions. [H-SR-15]	6G1	Update intervals support the timing constraints of the required control actions, as determined through mathematical analysis.
Notes: A symbol of the form [H-culture-8↑] indicates that the item "supports" or is derived from the linked item (e.g., H-culture-8) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

3.6.2 Contributory hazards from system architectural engineering

Using the reference model depicted in Figure 4 to the activities of architectural engineering, Table 13 identifies hazards contributed through some of the resources and elements employed in these activities and commensurate constraints on these process activities.

Table 13: Hazards contributed in system architectural engineering: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SAE-	**Description (**e.g., Scenario**)**	**ID H-SAE-**	**Description**
1	The architecture[86] description (including requirements allocated to its elements) is ambiguous, rendering it vulnerable to interpretations other than intended. For example, textual descriptions use words and expressions and graphic representations use symbols, for which unambiguous meanings have not been agreed upon by the community of its users. [H-S-2G-6↑] [H-SAE-2↓; H-SAE-3↓]	1G1	Description method supports distinct description of the following: 1. Assumptions about the environment 2. Input from the environment (e.g., command (some signal requiring state-changing effect + required behavior), query, data 3. Output (e.g., some signal having state-changing effect), state-notification, including exception-notification. 4. Functions assigned to a human 4.1. Procedure for the execution of each function assigned to a human (required behavior) 5. Other elements of the system 5.1. Functions assigned to each element; required behavior 6. Inter-relationships of elements 7. Interactions required across elements 8. Constraints on the behavior and interactions of each element, e.g. timing constraints; QoS constraints 9. Criteria to monitor and detect violation of a constraint_
		1G2	The language (graphic or text-based) used in the description or specification is unambiguous; for example: 1. Closed set of language elements 2. Unambiguous semantics of each language element 3. Unambiguous semantics of the compositions (e.g., rules of composition) of language elements and their compositions

[86] The term is used in its comprehensive sense, e.g., it includes conceptual architecture (or requirements architecture), system design architecture, software design architecture, hardware design architecture, software implementation architecture, function/procedure-architecture.

		1G3	The method and language are applied correctly.
2	Transformation, refinement or elaboration of architecture from one lifecycle phase to another does not preserve semantics and leads to unintended behavior	2G1	Methods and languages to describe, represent, or specify architectures (including requirements allocated to various elements) support unambiguous transformations or mappings across architectural artifacts (e.g., transformation from system conceptual or requirements level to system design level to software design level to software implementation level to procedure or subroutine or function level).
3	When dissimilar elements are integrated (have to work together), their interaction leads to unintended behavior, due to semantic mismatch (e.g., a signal from a sender does not have the same meaning for the receiver).	3G1	Methods and languages to describe, represent, or specify architectures (including requirements allocated to various elements) support unambiguous mapping and integration (including composability and compositionality for essential properties) across dissimilar elements (e.g., interactions across hardware and software elements).
4	When elements from different sources or suppliers are integrated (have to work together), their interaction leads to unintended behavior, due to semantic mismatch (e.g., a signal from a sender does not have the same meaning for the receiver).	4G1	Methods and languages to describe, represent, or specify architectures support unambiguous transformations or mappings and integration (including composability and compositionality for essential properties) across elements from different sources or suppliers.
5	A tool used in architectural engineering is not qualified to produce, manipulate or handle a safety grade architectural artifact (e.g., system, element, and data).	5G1	Each tool is qualified for safety grade use.
		5G2	Restrictions necessary for safe use of a tool are identified and the set of restrictions, tracked as a configuration controlled item.
6	Tools used in engineering a system, engineering software, or engineering hardware do not integrate correctly, that is, semantics may not be preserved in information exchanged across the tools	6G1	Tool-sets intended to be used collectively or in an integrated process are qualified for safety grade use.
		6G2	Restrictions on individual tools, their information exchange functions, and their interactions, which are needed for safe use of the tools as a set, are identified and the set of restrictions, tracked as a configuration controlled item.

		6G3	Semantics of the information accepted and provided by the tools are explicitly represented.
7	A reused element (e.g., from some previous project or system; previously verified to satisfy its specifications), when integrated in this system, does not provide the intended system behavior (e.g., semantics may not be preserved in the flowdown of specifications or their realization).	7G1	Pre-existing element is qualified for the environment[87] in which it is to be reused.
		7G1.1	Allocation of requirement specifications from system to the element is validated to be correct.
		7G1.2	Pre-existing specification of the element satisfies the requirement specification allocated from this system
		7G1.3	The element satisfies the allocated requirements specification
		7G2	Restrictions on the use of a pre-existing element in the target environment are identified and the set of restrictions, tracked as a configuration controlled item.
7.1	Some assumption about the reused element or its usage environment is violated. Also see H-SR-13.	7.1G1	H-ProcState-4G1.2; H-culture-12G1; H-SR-13G3
8	Individuals performing architectural engineering functions may not be cognizant of the usage-limitations of the tools, elements, and artifacts accessible to them.	8G1	Human resources employed in architectural engineering are qualified to perform their roles, especially usage-limitations of the tools, elements, and artifacts available to them, commensurate to the overall complexity of the cognitive activities to be performed.

3.6.3 Contributory hazards in Software Architecture

While the scope of architecture of the software in the safety system includes its relationship with its environment (e.g., hardware elements and human elements) this section focuses on software elements that are internal to the safety system and their inter-relationships, that is, interfaces, interconnections, and interactions, whether these are direct or indirect, intended or unintended, explicit or implicit, static or dynamic, "normal" or "abnormal[88]."

The scope of software architecture activities includes the allocation of requirements and constraints to elements identified in the software architecture.

> Note: The contents of this section are predicated on correct performance of HA, as discussed in preceding sections and complete satisfaction of the criteria to prevent, avoid, eliminate, contain, or mitigate the categories of hazards identified in those sections.

Table 7 and Table 12 are also applicable[89] to the software architecture, with software-related refinements added in Table 14. These considerations are applicable to architecture-related

[87] including assumptions about the environment – also see H-culture-12

[88] Examples: Invalid input; hardware malfunction; human mistake.

[89] Replace "system" with "software" or consider the scope of the system to be narrowed down to software.

contributory hazards in every phase in the software development lifecycle (from conception to implementation), to every level in the software architecture integration hierarchy[90], and to transformations from one phase or level to another.

Table 14: Hazards through software architecture and hazard-reducing conditions: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SwA-	**Description (**e.g., Scenario**)**	**ID H-SwA-**	**Description**
1	Scenario: Software contributes to or exacerbates complexity of the system, making it difficult to verify [H-S-1.1↑] and understand [H-S-2↑].	1G1	The behavior of a non-atomic element is a composition of the behaviors of its constituent elements, with well-defined unambiguous rules of composition[91]. (In [13] Appendix A.4 item 5) 1. Interfaces of elements are unambiguously specified, including behavior (adapted from [10] CP 2.3.3.2.2 last sentence). 2. Interactions across elements occur only through their specified interfaces, that is, adhering to principles of encapsulation (adapted from [14] CP 2.3.3.2.2).
		1G2	The system is modularized using principles of information hiding and separation of concerns, avoiding unnecessary interdependence (in [13] Appendix A.4 item 7).
		1G3	Each element (e.g., a software unit) is internally well-architected (that is, satisfying conditions stated earlier), such that QoS properties [Table 4] can be assured. For example: 1. A software unit implementing some NPP safety function(s) is composed from semantically unambiguous atomic functions using well-defined unambiguous rules of composition. [H-SwA-1G1↑] 2. Paths from inputs to outputs avoid unnecessary coupling. [H-SAE-1G2↑] 3. Unnecessary remembering of state information across execution cycles is avoided. (Adapted from CP 2.3.3.2.8 in [10])
2	Order of execution or timing behavior are not analyzable correctly, because of system complexity	2G1	Complexity-increasing behaviors are avoided [H-S-1.1.1G1↑]; simplicity-increasing features are preferred; for example: 1. Static configuration of tasks[92] to be executed (adapted from [14] CP 2.4.3.8.1 2nd and 3rd bullets).

[90] Examples: Subsystem; module; subroutine.

[91] Including conditions for composability and compositionality for required properties.

[92] Dynamic creation and destruction of tasks is avoided

			2. Tasks in execution are run to completion (adapted from [14] CP 2.4.3.8.1 1st bullet). 3. Static allocation of resources[93] [H-S-4G1↑] (Generalized from [14] CP 2.4.3.8.1 4th bullet).
3	Behavior is not analyzable mathematically or analysis is not mechanize-able for lack of a semantically adequate paradigm or model underlying the behavior specification or description. [H-SAE-{1,2,3}]	3G1	Behavior specification or description method is based on a semantically adequate, unambiguous paradigm [H-SAE-1G1↑; H-SAE-1G2↑], supporting association of timing constraints [H-SR-13G4↑], other QoS properties (Table 7↑), hierarchical nesting and abstraction [H-S-1.1G1↑]. Example paradigm: Extended finite state machine (adapted from [17] and 2.3.4.1.1 in [14]).
Notes: A symbol of the form [H-S-1.1↑] indicates that the item "supports" or is derived from the linked item H-S-1.1 A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

3.6.4 Contributory hazards in Software architectural engineering

Table 13 is also applicable to the architectural engineering of software, with software-related refinements added in Table 15.

Table 15: Hazards through inadequacy in software architectural engineering: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SwAE-	**Description (**e.g., Scenario**)**	**ID H-SwAE-**	**Description**
1	Loss of information across disciplines (e.g., system engineering, software engineering, and hardware engineering) due to discipline-wise division of organizations, people, and work [H-culture-9↑]	1G	H-SRE-tG1↑
2	Loss of information across disciplines due to incompatible paradigms, methods, and tools across disciplines	2G	H-SAE-{2G1, 3G1}↑
Notes: A symbol of the form [H-SAE-2G1↑] indicates that the item "supports" or is derived from the linked item (e.g., H-SAE-2G1) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

[93] Examples: Memory (information storage); Processor (execution time)

3.7 Evaluation of Hardware-Related Hazard Analysis

3.7.1 Contributory hazards in hardware architecture

Table 16: Hazards through hardware and hazard-space reducing conditions: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-Hw-	**Description (**e.g., Scenario**)**	**ID H-Hw-**	**Description**
1	Failure of hardware leads to unanalyzed conditions [H-S-1.1.1↑] (e.g., unknown state).	1G1	Only hardware with predictable, well-understood, well-known degradation behavior is used.
		1G2	Degradation is detectable before failure that could lead to unanalyzed conditions (e.g., unknown state) [H-S-1.2G3↑]. (Adapted from CP 2.2.3.7 1st clause in [14])
		1G3	Safety requirements are specified to maintain system in a safe, known state at all times, in all modes of usage, including degraded states and including maintenance. Safety functions may be online or offline; for example: 1. Monitor hardware condition [H-SR-4G1↑]; for example: 1.1. Online monitoring (e.g., cyclic; periodic) 1.2. Offline surveillance 2. Detect hardware fault [H-SR-4G2↑] – see H-Hw-1G4 3. Notify (other automation or human) [H-SR-4G5↑] 4. Intervene (to maintain system in safe state) [H-SR-4G3↑] 5. Preventative maintenance (e.g., scheduled replacement) 6. Provision of redundancy 6.1. Provision of diverse redundancy (Items 1-4 adapted from CP 2.2.3.7 in [14]); (Item 4 is generalized from CP 2.2.3.7 in [14])

		1G4	Requirements are identified for independent, timely detection of a contributory hazard in an instrument or other element upon which a safety function is dependent; for example: 1. In the case of a bi-stable device, the device can be feasibly only in one stable state or the other; then, an indication of both states at the same time is an anomaly. 2. In the case of a continuously controlled electric motor for a motor-operated valve, if the trajectory {electric current; displacement; time} for the transition from actuation command to completion is outside the envelope of feasibility, it is an indicator of an instrument anomaly. 3. The trajectory of feasible process state variables (set of values over time) is identified, such that indication of an instrument anomaly can be derived from sensed values in the infeasible region.
2	Anomaly in the state of the process is not recognized or identified or correctly understood due to inadequacy in instrumentation [H-SR-23↑]	2G1	Progressive degradation, drift, and such other changes in the behavior of instrumentation are properly accounted for; for example: 1. Monitoring and tracking such phenomena 2. Compensation 3. Calibration; recalibration; 4. Allowances (margins) for unaccounted, uncompensated, or unknown changes 5. Detection of unacceptable deviation and appropriate intervention – see H-Hw-1G3 items 2,4.
3	Anomaly in the state of the instrumentation for the safety functions or other element in the environment, upon which a safety function is dependent, is not correctly understood or recognized.	3G1	Instrument or element has behavior (including behavior in fault states), which satisfies requisite quality-of-service (Q-o-S) properties such as those identified in Table 7.
4	Loss or interruption of power	4G1	Safety functions are specified to maintain system in a safe, known state (adapted from CP 2.2.3.7 last sentence in [14])
5	Disturbance in power supply		
6	Inadvertent alteration of invariant information (e.g., program code; fixed data).	6G1	Store invariant information in read only memory (ROM). (Adapted from CP 2.7.3.3.2 in [14]).

7	Change in hardware that is nominally "equivalent" to replaced hardware (e.g., functionally, electrically, mechanically "interchangeable") leads to some subtle change that impairs a safety function.	7G1	Establish criteria for equivalence correctly and completely; for example: 1. Analyze differences in timing behavior. 2. Analyze differences in signal-noise discrimination. Also see Table 1
Notes: A symbol of the form [H-S-1.1.1↑] indicates that the item "supports" or is derived from the linked item H-S-1.1.1 A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

3.7.2 Contributory hazards from hardware engineering

Table 17: Hazards through hardware engineering processes: Examples

Contributory hazard		**Examples of conditions that reduce the hazard space**	
ID HwP-	**Description** (e.g., Scenario)	**ID H-HwP-**	**Description**
1	Loss of information across disciplines (e.g., system engineering, software engineering, and hardware engineering) due to discipline-wise division of organizations, people, and work [H-culture-9↑]	1G1	H-SRE-tG1↑
2	Loss of information across disciplines due to incompatible paradigms, methods, and tools across disciplines	2G1	H-SAE-{2G1, 3G1}↑
3	Preventative maintenance activities on which a safety function is dependent are not performed [19] when needed or scheduled [H-Hw-1G3].	3 G1	Maintenance schedules specify the preventative actions explicitly [H-Hw-1G3↑].
		3G2	These maintenance schedules are treated as safety related activities (e.g., including, performance; verification; audit) [Table 1].
4	Preventative protection against age-related degradation is not provided in maintenance plans (generalization from [20]).	4G	[see H-Hw-{1G1; 1G2}]

5	Computation is incorrect due to incorrect mapping of algorithm onto arithmetic hardware; for example, due to incompatibility in one or more of the following: 1. Hardware 2. Hardware interfacing software 3. Algorithm 4. Mapping algorithm software onto hardware 5. Associated library software [H-SwRE-2↑]	5G1	The hardware (e.g., floating point processor), algorithm (e.g., formula and data types in the application software), and the transformation (e.g., compiler and its configuration) are specified correctly. (Generalized from CP 2.4.3.5.8 in [14]).
		5G2	The hardware, software, and transformation are qualified and configured correctly for conformance to the specs (H-HwP-5G1).
6	Selection of output destination (e.g., actuator) or input source (e.g., sensor) is incorrect, for example, due to incorrect mapping from software to hardware.	6G1	I/O-identifying mappings from requirements to architecture to detailed design to implementation are verified to be correct. (Generalized from CP 2.3.3.1.7 1st sentence in [14]).
Notes: A symbol of the form [H1↑] indicates that the item "supports" or is derived from the linked item (e.g., H1) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

3.8 Evaluation of Hazard Analysis related to Software Detailed Design

The purpose of evaluating HA related to software design is to identify constraints to be satisfied by design activities performed after the licensing phase. These constraints become part of ITAAC commitments.

Many defects found during software detailed design are traceable to (rooted in) deficiencies from earlier phases in the development lifecycle. Earlier sections of this RIL have identified examples of those deficiencies as contributory hazards. The conditions to reduce the respective hazard spaces affect software detailed design also.

3.8.1 Contributory hazards in software detailed design

Table 18: Hazards through inadequate software design: Examples

Contributory hazard		**Examples of conditions that reduce the hazard space**	
ID H-SwD-	**Description** (e.g.,: Scenario)	**ID H-SwD-**	**Description**
1	Loss of information across disciplines (e.g., software architecture engineering and detailed software design). [H-SwAE-1↑]	G1	H-SAE-{2G1, 3G1}↑
2	Scenario: Software contributes to or exacerbates complexity of the system, making it difficult to verify [H-S-1.1↑] and understand [H-S-2↑]. [H-SwA-1↑]	G2	

3	Names of functions, data items, inputs, outputs, and variables in software are such that it becomes difficult to trace back to system requirements and further back to the application domain. (Adapted from [14] 2.3.4.1.2).	G3.1	Naming conventions and data dictionaries are established for ease of comprehension and bidirectional traceability.
		G3.2	Naming conventions and data dictionaries are used consistently.
Notes: A symbol of the form [H-culture-8↑] indicates that the item "supports" or is derived from the linked item (e.g., H-culture-8) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

3.9 Evaluation of Hazard Analysis related to Software Implementation

The purpose of evaluating HA related to software implementation is to assure that constraints to be satisfied by implementation performed after the licensing phase have been identified. These constraints become part of ITAAC commitments.

Many defects found during software implementation (coding) are traceable to (rooted in) deficiencies from earlier phases in the development lifecycle. Earlier sections of this RIL have identified examples of those deficiencies as contributory hazards. The conditions to reduce the respective hazard spaces affect software implementation also.

3.9.1 Contributory hazards in software implementation

Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumeration (CWE) are forms of contributory hazards in computer programs. Safe programming languages or safe subsets of appropriately selected programming languages reduce these hazard spaces effectively.

Table 19: Hazards contributed in software implementation: Examples

Contributory hazard		**Conditions that reduce the hazard space**	
ID H-SwI-	**Description (**e.g., Scenario**)**	**ID H-SwI-**	**Description**
1	Behavior is not analyzable mathematically or analysis is not mechanize-able, due to the complexity introduced through the improper use of interrupts or other mechanisms affecting order of execution.	1G1	Unnecessary use of interrupts is avoided, for example, not using interrupts to cover for inadequately understanding timing behavior of the physical phenomena (Table 1; H-SR-3G7) or the design and implementation (H-SR-13G4, H-SR-15G1)
		1G2	Schedulability analysis or proof is provided to verify that timing behavior of the implementation satisfies the specifications (H-SR-15G1).

2	Timing problems detract from deterministic behavior. Timing problems are difficult to diagnose and resolve.	2G1	The results produced by the programmed logic is not dependent on either: – the time taken to execute the program, or – the time (referenced to an independent "clock") at which execution of the program is initiated. (Adapted from [25])
		2G2	Execution speed does not affect correct order of execution.
Notes: A symbol of the form [H1↑] indicates that the item "supports" or is derived from the linked item, (e.g., H1) A symbol of the form [H-S-2G4↓] indicates that the item requires, spawns, or "is supported by" the linked item (e.g., H-S-2G4)			

4 Regulatory Significance & Relationship with NRC regulations

HA of a digital safety system is an approach to address clause 4h in [26] quoted below, where a "condition having the potential for functional degradation of safety system performance" is a hazard and a "provision … incorporated to retain the capability for performing the safety functions" is a requirement or constraint to eliminate, prevent or otherwise control the hazard.

> Clause 4 and sub-clause h in [3] A specific basis shall be established for the design of each safety system of the nuclear power generating station. The design basis shall also be available as needed to facilitate the determination of the adequacy of the safety system, including design changes. The design basis shall document as a minimum …:
>
> h) The conditions having the potential for functional degradation of safety system performance and for which provisions shall be incorporated to retain the capability for performing the safety functions …

Pursuant to 10 Code of Federal Regulations (CFR) 50.34(a)(1)(i), and corresponding clauses of 10 CFR 52.47(a)(2), HA of I&C for safety functions is a part of the "analysis…of the major structures, systems, and components…"

Pursuant to 10 CFR 50.34(a)(2), and corresponding clauses of 10 CFR 52.47(a), a part of the HA for DI&C systems identifies design characteristics and unusual or novel design features, and associated principal safety considerations. Recognizing from recent licensing review experiences, trends in design characteristics and unusual or novel design features, generally accepted engineering standards (as mentioned in 10 CFR 50.34(a)(ii)(B), and as referenced in NRC's regulatory guides) are not sufficiently specific to ensure consistent application and require significant judgment relying on high level of subject matter competence. In consideration of these trends and similar trends in other application domains and issues encountered in respective safety reviews, this research information letter identifies some contributory hazards and corresponding system characteristics and conditions that reduce the respective hazard spaces.

Pursuant to 10 CFR 50.34(a)(3)(i) and 10 CFR 52.47(a)(3)(i), the HA leads to the principal design criteria, additional[94] to or overlapping the general design criteria (GDCs) in 10 CFR 50 Appendix A, which provide only minimum requirements.

Pursuant to 10 CFR 50.34(a)(3)(ii) and 10 CFR 52.47(a)(3)(ii), the HA leads from the principal design criteria to design bases, that is, functions to be performed (functional requirements) and restraints (e.g., quality requirements, constraints on the architecture, and constraints on design and implementation), such that their satisfaction is verifiable through ITAAC activities. These derived requirements and constraints lead to the level of design information to which the following requirement in10 CFR 52.47 refers:

> "*The application must contain a level of design information sufficient to enable the Commission to judge the applicant's proposed means of assuring that construction conforms to the design and to reach a final conclusion on all safety questions associated with the design before the certification is granted. The information submitted for a design certification must include performance requirements and design information sufficiently detailed to permit the preparation of acceptance and inspection requirements by the NRC…"*

Pursuant to 10 CFR 50.34(a)(4), the HA is that part of the preliminary analysis, which yields principal design criteria, design bases, and derived requirements and constraints with the degree of specificity needed for consistent V&V. HA naturally organizes this information along flow-down paths from the degradation of a safety function, since it follows a cause-effect course of enquiry and reasoning, originating from the loss events of concern, examining the success paths, as well as the failure paths.

HA's cause-effect course of enquiry and reasoning also leads to specific information required per 10 CFR 50.34(a)(5-8) and 10 CFR 52.47(a)(7, 19), where critical to safety analysis.

5 Conclusions

This RIL provides the US Nuclear Regulatory Commission (NRC)'s licensing staff the technical basis to support their review of hazard analysis (HA) performed on a digital safety system by an applicant seeking a license, amendment to a license, or design certification. Content selection is based on the premise that NRC's existing guidance for reviewers and applicants is being utilized fully.

The RIL has been focused on issues encountered in NRO's recent licensing reviews – particularly hazards, which are rooted in systemic causes and are contributed through the development of a digital safety system; these hazards are called contributory hazards. Examples of associated contributory hazards include the following: Inadequate definition of the boundary of the digital safety system being analyzed; incorrect decomposition and allocation of NPP-level safety functions into NPP-wide I&C architecture and then to the digital safety system being analyzed; inadequate identification of safety requirements and associated quality properties and their flow-down into constraints on the architecture of the system and then the architecture of the software or other forms of logic; inadequate flow-down to identify requirements and constraints on technical processes, supporting processes, and organizational processes.

[94] These additional requirements or constraints may be specific to a facility, system, component or structure.

RIL-1101 reflects the state of the art. RES found very little published information organized specifically to support HA reviews for the environment characterized above. Therefore, information assimilated in the RIL includes knowledge acquired through consultation with external experts.

RES engaged diverse external subject matter experts to acquire knowledge from their respective experiences, and refined the RIL accordingly.

6 Follow-on R&D activities

Issues identified in the commenting period for which existing knowledge is inadequate will be considered as candidates for further R&D activities. While it is expected that some issues might be resolved through further consultation with experts, other issues might have to be studied further through pilot applications (learning cycles).

7 Abbreviations and Acronyms

ACRS	Advisory Committee on Reactor Safeguards
CFR	Code of Federal Regulations
CP	common position[95]
DI	design and implementation
DI&C	digital instrumentation and control
FMEA	fault modes effects and analysis
FTA	fault tree analysis
GDC	general design criteria
HA	hazard analysis
HAZOP(S)	hazard and operability studies
I&C	instrumentation and control
IT	information technology
ITAAC	inspections, tests, analyses, and acceptance criteria
NPP	nuclear power plant
NRC	U.S. Nuclear Regulatory Commission
NRR	Office of Nuclear Reactor Regulation
NRO	Office of New Reactors
PHA	preliminary hazard analysis
QoS	quality-of-service
R&D	research and development
RAI	request for additional information
RIL	research information letter
SAR	safety analysis report
SRP	standard review plan
TFSCS	Task Force[96] for Safety Critical Software
TMI	Three Mile Island
V&V	verification and validation

[95] A term used in [14] for a requirement on which the TFSCS has total consensus

[96] It consists of regulatory experts from the UK, Germany, Sweden, Belgium, Finland, and Spain

8 References

[1] US NRC Design-Specific Review Standard for the mPower Design, "Appendix A – Instrumentation and Controls: Hazard Analysis," ML12318A200, 2013.

[2] Corcoran, W.R., "Hazard recognition for quality, safety, and performance improvement," Special issue of the Firebird Forum, volume 15, number 3, March 2012.

[3] IEEE Standard 603-2009, "IEEE standard criteria for safety systems for nuclear power generating stations" 2009.

[4] MIL-STD-882E, "Standard Practice for System Safety," U.S. Department of Defense, 2012.

[5] Ericson II., C.A., "Hazard Analysis Techniques For System Safety," John Wiley and Sons, August 24, 2005.

[6] U.S. Air Force, "The Air Force *System Safety Handbook"*, Kirtland AFB, NM, July 2000.

[7] National Aeronautics and Space Administration, "NASA Software Safety Guidebook*"*, NASA-GB-8719.13, Washington, DC, March 31, 2004.

[8] U.S. Nuclear Regulatory Commission, "Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition," Branch Technical Position 7-14, "Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems," NUREG-0800, Revision 5, Washington, DC, 2007.

[9] U.S. Nuclear Regulatory Commission, "Recommendations for Enhancing Reactor Safety in the 21st Century," The Near-Term Task Force Review of Insights from the Fukushima Dai-Chi Accident, Washington, DC, July 12, 2011.

[10] U.S. Nuclear Regulatory Commission, "Inadequate Flooding Protection Due to Ineffective Oversight," Licensee Event Report 285-2011-003, May 1, 2011.

[11] Garrett, C. and Apostolakis, G., "Context in the risk assessment of digital systems" Risk Analysis Vol. 19 No. 1 1999.

[12] U.S. Nuclear Regulatory Commission, Fault tree handbook (NUREG 492). URL: http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/sr0492.pdf

[13] NASA, "Fault tree handbook with aerospace applications." URL: http://www.hq.nasa.gov/office/codeq/doctree/fthb.pdf

[14] SAE J1739, "Potential Failure Mode and Effects Analysis in Design (Design FMEA), Potential Failure Mode and Effects Analysis in Manufacturing and Assembly Processes (Process FMEA), 2009" URL: http://standards.sae.org/j1739_200901/

[15] NASA, "Standard for Performing a Failure Mode and Effects Analysis (FMEA) and Establishing a Critical Items List (CIL) (DRAFT): Flight Assurance Procedure (FAP)-322-209," Nov. 2011, Available: rsdo.gsfc.nasa.gov/documents/Rapid-III-Documents/MAR-Reference/GSFC-FAP-322-208-FMEA-Draft.pdf

[16] Perrow, Charles, "Normal Accidents: Living with High Risk Technologies , New York: Basic Books, 1984"

[17] Proceedings of the NRC Advisory Committee for Reactor Safeguards 591st meeting, Rockville, Maryland, February 10, 2012. URL: http://pbadupws.nrc.gov/docs/ML1205/ML12054A637.pdf

[18] U.S. Nuclear Regulatory Commission, "Fort Calhoun Station – NRC Follow-up Inspection – Inspection Report 05000285/201007; Preliminary Substantial Finding," NRC Inspection Report 05000285/20010007, July 15, 2010.

[19] U.S. Nuclear Regulatory Commission, "Research Information Letter 1001: Software-related uncertainties in the assurance of digital safety systems Expert Clinic Findings, Part 1", ADAMS Accession Number ML1035402040, January, 2011.

[20] U.S. Federal Aviation Administration, "*System Safety Handbook"*, Washington, DC, December 2000.

[21] Task Force for Safety Critical Software, "Draft Licensing of Safety Critical Software for Nuclear Reactors," Common Position of Seven European Nuclear Regulators and Authorised Technical Support Organizations, Revision 2010. URL: http://www.hse.gov.uk/nuclear/software.pdf

[22] PNO-77-146_8-19-77

[23] ISO/DIS 26262-2, "Road Vehicles – Functional Safety – Part 2: Management of functional safety", 2009.

[24] Garrett, C. and Apostolakis, G., "Automated hazard analysis of digital control systems" Reliability Engineering and Safety Society 77 (2002) 1-17

[25] NRC "Recent Operating Experience on Ineffective Use of Vendor Technical Recommendations" July 26, 2012, URL: http://nrr10.nrc.gov/forum/forumtopic.cfm?selectedForum=03&forumId=AllComm&topicId=3165&bookMark=316520110210144245673&searchId=1

[26] NRC IN-2012-11, "Age-related capacitor degradation" July 23, 2012 URL: http://pbadupws.nrc.gov/docs/ML1203/ML120330272.pdf

[27] P. Clements, R. Kazman, and M. Klein. Evaluating software architectures. Addison-Wesley, 2005

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

[29] CVE, see: http://cve.mitre.org/ (last accessed August 1st 2013)

[30] CWE, see: http://cwe.mitre.org/ (last accessed August 1st 2013)

[31] IEC 60880:2006 Nuclear power plants – Instrumentation and control systems important to safety – software aspects for computer-based systems performing category A functions.

Appendix A: Glossary

The scope of this glossary is limited to this document.

Where a word is not defined explicitly in the glossary, it is understood in terms of common usage as defined in published dictionaries of the English language (e.g., [1]).

The glossary focuses on terms that are not commonly understood in the same way, removing or reducing ambiguity by selecting and using more specific definitions. Where needed, notes elaborate the definition.

Where possible, the definition of a technical term is traceable to an authoritative reference source. In cases where the authorities have different, inconsistent definitions, the glossary adapts the definition and includes explanatory notes to reduce ambiguity.

The meanings of compound words, terms, and expressions are derived from the meanings of their constituent words, as defined in this glossary.

Aliasing

In signal processing and related disciplines, **aliasing** refers to an effect that causes different signals to become indistinguishable (or *aliases* of one another) when sampled. It also refers to the distortion or artifact that results when the signal reconstructed from samples is different from the original continuous signal [2].

Analysis

A course of reasoning showing that a certain result is a consequence of assumed premises (definition (2) in [1]).

Notes:

1. The term “course” is interpreted to mean “process” (adapted from definition 4 in [4]).
2. Derived forms:
    2.1. Analyzability
    2.2. Analyzable
    2.3. Un-analyzable
    2.4. Unanalyzed

Architecture

The structure or structures of the system, which comprise elements (e.g., software), the externally visible properties of those elements, and the relationships among them and with the environment (adapted from [5])

Where:

1. Externally visible properties of an element include behavior – normal, as well as abnormal – as seen from outside the boundary (interface) of an element.
2. Relationships include interactions and interconnections (communication paths).

3. Environment of the system includes the combination of systems and elements (e.g., hardware, software, and human) external to this system, human elements interacting directly with the system and the commensurate manual procedures.
4. System means 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. For example, if a program comprises of subroutines, then the subroutines are elements and the program is a system.
5. Element is a discrete part of a system that can be implemented to fulfill specified requirements. A system element can be hardware, software, data (structure), human, process (e.g., process for providing service to users), procedure (e.g., operator instructions), facility, materials, and naturally occurring entity (e.g., water, organism, mineral), or any combination.

Assure

Confirm the certainty of correctness of the claim, based on evidence and reasoning.

Notes:

1. For example, by proof
2. Examples of claims: (1) The system is safe. (2) Property X of the system holds.
3. Derived forms:
   3.1. Assurance
   3.2. Assurable
   3.3. Assurability

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 [33])

Notes:

1. The means of distinction may be manual or automated.
2. Also see "Quality measure" and "Scale."

Byzantine Behavior

In a distributed system, arbitrary behavior in response to a failure is called Byzantine behavior [6].

Note:
Arbitrary behavior of an element that results in disruption of the intended system behavior.

Claim

A true-false statement about the value of a defined property of a system. (Adapted from [13])

Notes:

1. A property is a quality attribute of the system. (Adapted from 4.3.9 and 4.4.1 in [14])
   1.1. Example of property: Safety.
2. A property may have supporting sub-characteristics [14]**.**.
   2.1. Example: Verifiability ← Analyzability ← "Freedom from interference"

3. Unlike physical quantities, a property sub-characteristic may not be measurable on an absolute scale [14] .
   3.1. Indicators may be associated with a sub-characteristic for its estimation or indirect measurement.
4. A sub-characteristic may be specified in terms of conditions or constraints on its behavior [14] .
   4.1. Example sub-characteristic of safety property: Restriction on allowed system states.
   4.2. Example sub-characteristic of "Freedom from interference": Constraints on flows or interactions.
5. "Value" may be a single value, a set of single values, a range of values, a set of ranges of values, and limits on values. Value can be multi-dimensional [14].
6. "Value" may be invariant, dependent on time, or dependent on some other conditions [14].
7. Associated with a property may be the duration of its applicability (i.e., not limited to the present). For example, the property may concern the future behavior of the system [13].
8. Uncertainty (lack of certainty) may be associated with the property [13].
   8.1. The value of uncertainty may not necessarily depend upon probability..
   8.2. Uncertainty may be associated with a sub-characteristic.
   8.3. Uncertainty may be associated with the duration of applicability
   8.4. Uncertainty may be associated with other conditions of applicability
   8.5. For example, evaluation of a claim may be based upon certain conditions, formulated in terms of assumptions that the identified uncertainties do not exist.

Complexity

The degree to which a system or component has a design or implementation that is difficult to understand and verify. (Definition (1)(A) in [3])

Notes:

1. The selection[97] of this definition was favored by Dr. Gerard Holzmann [7].
2. The term, Simplicity, the converse of Complexity, is often used to discuss the same issues.
3. A "complexity measure or indicator" is often confused with the concept of "complexity", but should be distinguished as follows:
   3.1. A complexity measure pertains to any of a set of structure-based metrics that measure the attribute in Definition (1)(A) in [3]. (Definition (1)(B) in [3])
   3.2. Example of an indicator: The number of linearly independent paths (one plus the number of conditions) through the source code of a computer program is an indicator of control flow complexity, known as McCabe's cyclomatic complexity [3].
   3.3. Sometimes, the term "size-complexity" is used to refer to the effect of the number of states and number of inputs and their values and combinations.
4. Complexity theory is concerned with the study of the intrinsic complexity of computational tasks, that is, a typical Complexity theoretic study considers the computational resources required to solve a computational task (or a class of such tasks); it studies what can be achieved within limited time (and/or other limited natural computational resources) [8]. For example, the time required to solve a problem – calculated as function *f(…)* of the size of the instance, usually the size of the input, *n* – is studied for its scalability (e.g., bounded by "order of" *O(…)* with respect to the input size *n*). Similarly, instead of time, one could study the scalability with respect to some other resource constraint (e.g., space or memory). An example of a useful result from this theory is a premise that only those problems that can be solved in polynomial time, denoted as $O(n^k)$ for some constant *k*, can be feasibly computed on some

[97] Various standards provide different definitions; there is no broadly accepted definition.

computational device [9]. Applying this thesis to evaluation of system architecture, one could conclude that, if the input space of a system is not bounded, the system is not verifiable. One could further conclude that, if the interactions across elements of the system are not bounded, the system is not verifiable.

Complex Logic

An item of logic for which it is not practicable to ensure the correctness of all behaviors[98] through verification alone.

Notes:

1. This definition is derived from a combination of the definition of complexity given above and the following definition in DO-254/ED-80 in Appendix C [11], for "simple hardware item": "A hardware item is considered simple if a comprehensive combination of deterministic tests and analyses can ensure correct functional performance under all foreseeable operating conditions with no anomalous behavior." The conditional clause "if a comprehensive combination of deterministic tests and analyses…" is summarized as "verification."
2. Therefore, in addition to verification, the demonstration of correctness of Complex Logic requires a combination of evidence from various phases of the development life cycle, integrated with reasoning to justify the completeness of coverage provided (summarized as development assurance). Examples include the following:
    1.1. Evaluation of the system concept (and conceptual architecture)
    1.2. Evaluation of the verification and validation plan
    1.3. Criticality analysis
    1.4. Evaluation of the architecture including requirements allocation
    1.5. Evaluation of the system-internal hazard analysis
    1.6. Validation of requirements and constraints on the design and implementation
    1.7. Assessment and audit of all processes, including supporting and management processes.
    1.8. Certifying[99] organizations developing software
    1.9. Evaluation of the independence[100] of the assurance activities
    1.10. See [11] for more detail.
3. Complex Logic is typically produced by techniques such as software or hardware description languages and their related tools. Thus, the assurance of correctness also requires commensurate assurance of the languages and tools.

Contribute

To play a significant part in bringing about an end or result (Definition 1b in [12])

Notes:

1. Derived forms:
    1.1. Contribution: The thing contributed
    1.2. **Contributory:** Of, relating to, or forming a contribution

Defect

---

[98] This refers to behaviour under all foreseeable operating conditions with no anomalous behaviour.

[99] Certification of the development organization should be a continual process of certification and recertification much in the same manner as reactor operators are certified periodically. For example, the capability maturity model integrated certification process developed by the Software Engineering Institute focuses on assessing the capabilities of development.

[100] For example, independence can be evaluated through certification of the assurance process for the Complex Logic (e.g., software).

An imperfection or deficiency in a project component where that component does not meet its requirements or specifications and needs to be either repaired or replaced. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) — Fourth Edition.* [31]

Notes:

1. The condition "that component does not meet its requirements or specifications" would exclude cases where the requirement or specification itself is deficient.
2. Another definition in [31] "a problem which, if not corrected, could cause an application to either fail or to produce incorrect results. *ISO/IEC 20926:2003, Software engineering — IFPUG 4.1 Unadjusted functional size measurement method — Counting practices manual*" depends upon the definition of "failure" and "correctness" both of which, in turn, are evaluated with respect to requirements. Thus, this definition would also exclude cases where the requirement or specification itself is deficient.
3. From notes 1 and 2, it can be seen that a system may not be defective; yet it may lead to a hazard.
4. In RIL-1101, the term is used primarily in the context of the engineering phases of the product lifecycle.

Diverse team

A team composed of individuals with complementary attributes needed to perform the assigned task (e.g., thought processes, communication styles, and competence, including education training, and experience in different domains and disciplines).

(System) Element

A discrete constituent of a system (adapted from [16]).

Notes:

1. The term "discrete constituent" is substituted for the word "component" used in the definition from [16]. Reason: Avoid confusion with other meanings of "component" in the context of software. The word "discrete" implies that the constituent has a distinct boundary, that is, interface with its environment (per definition in [17]), and an intrinsic, immutable, unique identity (adapted from [16]).
2. Examples:
   2.1. Hardware element
   2.2. Software element
   2.3. Human element
   2.4. Data element
   2.5. Process
   2.6. Procedure (e.g., operating instructions)
3. An element may have other elements in it (e.g., a subsystem).
4. A system may itself be an element of a larger system.

Environment

A general term relating to everything (including every condition) that supports or affects the performance of a system or a function of the system. (Combination of 9A and 9B in [3] which refer to (C) 610.12-1990)

Note:

1. The environment of a software component consists of all the elements (in their respective states or conditions), with which it interacts, by which it is affected, and on which it depends. Examples of elements:
   1.1. Other software components
   1.2. Operating system (common services and resources shared by software components)

1.3. Execution hardware

2. The environment of an electronic hardware component consists of physical environmental conditions and other hardware components (in their respective states or conditions) with which it interacts, by which it is affected, and on which it depends. Examples of physical environmental conditions:
   2.1. Temperature
   2.2. Humidity
   2.3. Electromagnetic radiation

Error

The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition (Definition (8)(A) in [3])

Evidence

Data supporting the existence or verity of something. (Adapted from 3.1936 in [31])

Note:

1. Examples of means of obtaining "raw" evidence: Test; measurement; observation.
2. Examples of evidence incorporating reasoning:
   2.1. Confirmation by static analysis that an implementation satisfies its design specification.
   2.2. A claim at one level of integration used as evidence in claim for next higher level of integration of a system.

Failure

The termination of the ability of an item to perform a required function. [18]

Notes:

1. After failure, the item has a fault. [18]
2. "Failure" is an event, as distinguished from "fault" which is a state. [18]
3. This concept as defined does not apply to items consisting of software only.[18]
4. The following definitions represent the perspectives of different disciplines to reinforce the definition given above:
   4.1. The termination of the ability of an item to perform a required function (Definition (1)(A) in [3]).
   4.2. The termination of the ability of a functional unit to perform its required function (Definition (1)(N) in [3]).
   4.3. An event in which a system or system component does not perform a required function within specified limits; a failure may be produced when a fault is encountered (Definition (1)(O) in [3]).
   4.4. The termination of the ability of an item to perform its required function (Definition 9 in [3] from "nuclear power generating station").
   4.5. The loss of ability of a component, equipment, or system to perform a required function (Definition 13 in [3] Safety systems equipment in "nuclear power generating stations").
   4.6. An event that may limit the capability of equipment or a system to perform its function(s) (Definition 14 in [3] "Supervisory control, data acquisition, and automatic control").
   4.7. The termination of the ability of an item to perform a required function (Definition 15 in [3] "nuclear power generating systems")

Failure Analysis

The logical, systematic examination of a failed item to identify and analyze the failure mechanism, the failure cause, and the consequences of failure. (191-16-12 in [18])

Fault

The state of an item characterized by inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources. (191-05-01 in [18])

Notes

1. A fault is often the result of a failure of the item itself but may exist without prior failure.
2. Also see "defect."
3. Distinguish from failure, mistake, and error.

Fault Analysis

The logical, systematic examination of an item to identify and analyze the probability, causes, and consequences of potential faults. (191-16-11 in [18])

Fault Mode

One of the possible states of a faulty item, for a given required function.

Note:
RIL-1101 does not use the term "failure mode" in this sense.

Fault Modes and Effects Analysis (FMEA)

A qualitative method of reliability analysis, which involves the study of the fault modes, which can exist in every sub-item of the item, and the determination of the effects of each fault mode on other sub-items of the item and on the required functions of the item. (191-16-03 in [18])

Note:
RIL-1101 does not use the term "failure mode and effects analysis" in this sense.

Fault Tree Analysis (FTA)

An analysis to determine which fault modes of the sub items or external events, or combinations thereof, may result in a stated fault mode of the item, presented in the form of a fault tree. (191-16-05 in [19])

Faulty

Pertaining to an item that has a fault.

Feasible

Capable of being done with the means at hand and circumstances as they are. [20]

Notes:

1. Other definitions also impose such constraints as
   1.1. Practicably
   1.2. Reasonable amount of effort, cost, or other hardship [21]
   1.3. Cost-effectiveness. [22]
2. Such constraints distinguish "feasibility" from "possibility."

Hardwired

Pertaining to a circuit or device whose characteristics and functionality are permanently determined by the interconnections[101] between components[102] (Adapted from Definition 3 in [3]).

Note:
The referred-to connections are at the printed circuit board level (or cabinet level), not internal to integrated circuits.

Hazard

Potential for harm[103]

Examples:

1. A condition;
2. A circumstance;
3. A scenario.

Notes:

1. RIL-1101 bounds the scope to the entity (system; element) in the context of a defined environment.
2. At the initial stage of hazard logging (before any analysis of the initial finding), the log may include an item, which, after some analysis, is re-characterized (differently from the originally characterized hazard; pssoibly, an event).
3. Definition A in [15] (same as definition 3.1283-1 in [31]) elaborate on the "potential for harm" as follows, "An intrinsic property or condition that has the potential to cause harm or damage."

Contributory hazard

Factor contributing to potential for harm.

Notes:

1. (Excerpt from [23]) …. An unsafe act and / or unsafe condition which contributes to the accident[104], ...
2. Figures 7-1 - 7-4 in [24] illustrate contribution paths.

Examples:

1. The potential for adverse energy flow [23]
2. Inappropriate functions (from Figure 7-5 in [24])
3. Normal functions that are out of sequence (from Figure 7-5 in [24])
4. Functional damage and system degradation (from Section 7.1.1 in [24])
5. Machine-environment interactions resulting from change or deviation stresses as they occur in time and space (from Section 7.1.1 in [24])

Hazard Analysis

Hazard analysis (HA) is the process of examining a system throughout its lifecycle to identify inherent hazards (see) and contributory hazards, and requirements and constraints to eliminate, prevent, or control them.

Notes:

[101] Examples: Wiring in cabinets; Printed paths in circuit boards

[102] Examples: Relays; AND-gates; OR-gates

[103] In general, "loss" of any kind that is of concern. Focus of RIL-1101: Harm.

[104] in our case, degradation of a safety function

1. "Hazard identification" part of HA includes the identification of losses (harm) of concern.
2. This definition is narrower than many definitions of HA, some of which correspond to the NRC's usage of the term "safety analysis" (as in a safety analysis report).
    a. The scope of the definition excludes the verification that the requirements and constraints have been satisfied.
    b. Various HA definitions and descriptions identify artifacts (results, including intermediate results) of HA by different names. The expression "requirements and constraints" used in this definition (to align and integrate them in well-established systems engineering terms) subsumes them.
    c. The scope of the definition does not include quantification explicitly. Where appropriate (e.g., for a hardware component, quantification of its reliability would be implicit in the activity of formulating requirements and constraints).

Hazard Identification

The process of recognizing that a hazard exists and defining its characteristics [31].

Indicate

To be a sign, symptom, or index of [1].

Note:

1. Derived form: **Indicator –** A device or variable that can be set to a prescribed state based on the results of a process or the occurrence of a specified condition. [3]
2. Often an indicator is an estimate or a result of evaluation, possibly incorporating judgment, and not measured on a standardized scale (or norm).
3. An indicator is created for its potential utility by facilitating comparison of current state with goal state, rather than for absolute accuracy.
4. Contrast with quality measure.

Information hiding

The principle of segregation of design decisions in a computer program that is most likely to change, thus protecting other parts of the program from extensive modification if the design decision is changed. The protection involves providing a stable interface which protects the remainder of the program from the implementation (the details that are most likely to change).

Item (Entity)

Any part, component, device, subsystem, functional unit, equipment, or system that can be individually considered. (191-01-01 in [19])

Notes:

1. An item may consist of hardware, software, or both, and may, in particular cases, include people.
2. A number of items (e.g.,, a population of items) or a sample may itself be considered an item.

Mistake

A human action that produces an unintended result (Definition 1 in [3] “electronic computation”)

Editorial note (contrary to the note attached to Definition 1 in [3]): In the context of software engineering, this definition should be applied to mistakes concerning requirements development (including elicitation, transformation of intent into requirement or constraint specification, and explicit statement of assumptions (e.g., about the environment) and respective validation.

A human action that produces an incorrect result (Definition 3 in [3] “software”)

Note: The fault tolerance discipline distinguishes between the human action (a mistake), its manifestation (a hardware or software fault), the result of the fault (a failure), and the amount by which the result is incorrect (the error). [3]

Editorial note (complementing the note in the previous definition of “mistake”): In the context of software engineering, this definition should be applied to mistakes concerning transformation of requirements specifications and constraints into successive work products and their respective verification.

Mode confusion

A situation in which an engineered system can behave differently from its user’s expectation, because of a misunderstanding or inadequate understanding of the system state.

Noninterference

Absence of cascading failures between two or more elements that could lead to the violation of a safety requirement [22].[105]

Example 1: Element 1 is interference-free of Element 2 if no failure of Element 2 can cause Element 1 to fail.

Example 2: Element 3 interferes with Element 4 if there exists a failure of Element 3 that causes Element 4 to fail.

Process

A set of interrelated activities, which transforms inputs into outputs. (Definition 12(A) in [3]. Definition 3.2217-1 in [31])

Notes

1. Definition 4 in [3] makes “including the transition criteria for progressing from one (activity) to the next” explicit.
2. In definition 4 in [3], the expression “that bring about a result” corresponds to “which transforms inputs into outputs.” The latter is used in the definition above, because it identifies a set of starting conditions (inputs), a set of end conditions (outputs) and the transformational purpose of the process.
3. Examples of transformational processes in an engineering lifecycle of a product: Requirements; Architecture; Detailed design; Implementation. If the overall engineering is considered a lifecycle process, then these may be identified as phases in that lifecycle process.

[105] This reference uses the term “freedom from interference.”

Product

Result of a process. (3.2257-4 in [31])

Notes:

1. Referring to Note 3 for process, the term "product" may be used for the final product or for a result of a particular phase of a lifecycle process; for example: System requirements specification; System architecture specification; Detailed design specification; (Software) source code; (Software) executable code.

Quality

Capability of product to satisfy stated and implied needs when used under specified conditions. (Adapted from 4.51 in [32])

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 [32])
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.
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.

Reason

Argument: A logical sequence or series of statements from a premise to a conclusion. (Adapted from http://www.merriam-webster.com/dictionary/argument

Notes:

1. **Argument:** Also see http://www-rohan.sdsu.edu/~digger/305/toulmin_model.htm
2. Derived forms:
    2.1. **Reasoning**: The use of reason

2.2. **Reasonable**: Being in accordance with reason. (http://www.merriam-webster.com/dictionary/reasonable)

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

The probability that an item can perform a required function under given conditions for a given time interval (t_1, t_2). (191-12-01 in [19])

Notes:

1. It is generally assumed that the item is in a state to perform this required function at the beginning of the time interval.[106]
2. The term "reliability" is also used to denote the reliability performance quantified by this probability (see 191-02-06 in [19]).
3. This definition does not apply to items for which development mistakes can cause failures, because there is no recognized way to assign a probability to development mistakes.

Requirement

Expression of a perceived need that something be accomplished or realized. (Adapted from 4.47 in [32])

Notes:

1. Functional requirement: Requirement that specifies a function that a system or its element must be able to perform, (Adapted from 4.22 in [32])
2. Quality requirement: Requirement that specifies a quality of a system or its element, where quality may be one of the following:
   2.1. Quality in use (e.g., safety). Quality in use requirements specify the required level of quality from the end user's point of view. Also see note 5 in definition of quality.
   2.2. External quality. Also see note 6 in definition of quality.
   2.3. Internal quality. Also see note 7 in definition of quality.

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 [33])

Notes

1. The type of scale depends on the nature of the relationship between values on the scale [33].
2. Four types[107] of scale are commonly defined [33]:
   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.

[106] For a software component that is faulty to begin with, use of the term reliability is neither meaningful nor helpful; instead, it leads to the misapplication of analysis techniques that served well for traditional hardware.

[107] See [34] for other types of scale.

4. The mapping of the magnitude of the measured attribute to a value on the scale is predetermined.

Separation of concerns

The process of separating a computer program into distinct features that overlap in functionality as little as possible. A concern is any piece of interest or focus in a program. Typically, concerns are synonymous with features or behaviors. [25]

State

The present condition of a (dynamic) system or entity.

Note:

A state is a complete set of observable properties (also known as state variables) that characterize the behavior of a system, that is, response to stimuli (set of inputs).

State space

The set of all possible states of a dynamic system [26].

Note: Each state of the system corresponds to a unique point in the state space.

System

Combination of interacting elements organized to achieve one or more stated purposes [27].

Notes

1. A system may be considered as a product or as the services it provides (adapted from [27]). For example, at its conceptualization stage, a system may be described in terms of the services it provides and its interactions with its environment, without identifying its constituent elements.
2. The expression “combination…organized…” (instead of collection) emphasizes that a system is an “integrated composite” as characterized from the definition in [28] of system.
3. The expression “to achieve its stated purposes” corresponds to the expression “a capability to satisfy a stated need or objective” used in the definition in [28] of system.
4. In practice, the interpretation of its meaning is frequently clarified by the use of an associated noun (e.g., reactor protection system). (Adapted from [27])
5. System elements may include people, products and processes (adapted from [28]). In the boundary of NRC’s licensing review plan (DSRS) Chapter 7, the review of a digital safety system is focused on the safety automation. Operators, thermo-hydraulic processes, and related supporting, peripheral processes are part of the environment of the digital safety system. The scope of Chapter 7 review includes Interactions of the digital safety system with its environment.

Systemic

Embedded within and spread throughout and affecting a group, system, or body. Also see “systemic cause” in [29].

Systematic Failure

Failure, related in a deterministic way to a certain cause, that can be eliminated only by a modification of the design or of the manufacturing process, operational procedures, documentation, or other relevant factors. [19]

Notes

1. Corrective maintenance without modification will usually not eliminate the failure cause.

2. A systematic failure can be induced by simulating the failure cause.
3. In International Electrotechnical Commission 61508-4 CDV 3.6.6 [30]: Examples of causes of systematic failures include human mistakes in the following areas:
    a. The safety requirements specification
    b. The design, manufacture, installation, and operation of the hardware
    c. The design, implementation, etc. of the software
4. Other examples include mistakes in modification and configuration.
5. Also, see “systemic cause” in [29].

Traceability

Discernible association among two or more logical entities, such as requirements, system elements, verifications, or tasks.

Validation

Confirmation that a product satisfies the needs of the customer and other identified stakeholders. (Adapted from 3.3264-5 in [31]).

Notes

1. “Confirmation” is used instead of “Assurance,” the word used in [31]. Rationale:
    1.1. Avoid confusion with the use of the word “Assurance” in RIL_1101.
    1.2. Consistency with the use of “Confirmation” in the definition of “Verification.”
    1.3. “Confirmation” subsumes the term, “the process of evaluating” used within definition A in [15].
    1.4. “Confirmation” subsumes the term, “the process of providing evidence” used within definition B in [15].
2. “Product” subsumes the elaboration, “system, software, or hardware and its associated products” used within definition B in [15].
3. “Satisfies” is used instead of “meets,” the word used in [31]. Rationale: Consistency with usage in the definition of “Verification.”
4. A clarification of the expression, “the needs of the customer and other identified stakeholders” is provided within definition B in [15] as follows: Solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions), and satisfy intended use and user needs.
5. The concept of “validation,” as defined, subsumes the concept of “verification.” However, there is a lack of clear agreement across various authorities on the subsumption of “verification” in “validation.”.
6. The elaboration “….satisfy requirements allocated to it at the end of each life cycle activity” within definition B in [15] is subsumed in the expression, “satisfies the needs of the customer and other identified stakeholders”.
7. The activity of validation includes the confirmation that the specification for each lifecycle phase satisfies the needs of the customer and other identified stakeholders.
8. The stakeholder requirements definition activity includes the transformation of various needs into requirements, including the requirements for validation [10].
9. In [15], validation of stakeholder requirements definition includes HA.
10. “Validation” includes confirmation that the requirements are correct, complete, consistent, and unambiguous

Verification

Confirmation that specified requirements have been satisfied. (Adapted from **3.3282-3 in** [31]).

Notes

1. Various standards and authorities have different definitions, which are inconsistent with each other. The definition given above abstracts commonality to the extent possible. The following notes provide explanations, including attempts to reconcile some differences across certain definitions where possible.

2. The term is also used to mean the process of confirmation that specified requirements have been satisfied. The usage context will distinguish the two meanings.

   2.1. Definition A in [15] characterizes the verification process "… evaluating … to determine whether … product …. satisfy … " If the result of the determination is TRUE, then it is "confirmation."

   2.2. The object of verification is implied in the definition (e.g., confirmation that a product satisfies its specified requirements).

3. Definition 3 in [31] uses the term "fulfilled"; however, to reduce potential ambiguity, the term "satisfied" is used (which is also used in definition 1 within [31]) in the general sense of propositional satisfaction (⊨) and constraint satisfaction. Definition 2 in [31] uses the term "formal proof" favoring this substitution. Definition 6 in [31] uses the term "comply with" which may be mapped conservatively into "satisfies." Definition B in [15] uses the term "conforms to" which may be mapped conservatively into "satisfies."

4. Definitions 3 and 6 in [31] also include the phrase "through the provision of objective evidence." This phrase is omitted, because the concept "satisfied," as explained in Note 3 subsumes it,

5. Definition A in [15] uses the expression "satisfy the conditions imposed at the start of that phase"; this expression is mapped into "specified requirements" in the definition above.

6. Definition B in [15] elaborates "… for all life cycle activities during each life cycle process"; the definitions of product and process subsume this elaboration.

7. Definition B in [15] elaborates "satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities"; the term "specified requirements" in conjunction with definitions of product and process subsumes this elaboration.

8. Definition B in [15] includes the statement "Verification of interim work products is essential for proper understanding and assessment of the life cycle phase product(s)." This statement does not add to the definition of verification.

9. Definition 3 in [3] elaborates "The act of reviewing, inspecting, testing, checking, auditing, or otherwise determining and documenting whether …"; the term "process" in the definition given in Note 2 abstracts this elaboration.

10. Verification at each lifecycle phase does not imply verification of the end product, because its scope does not include the confirmation that the specification for each lifecycle phase satisfies the requirements at the initial phase (e.g.,, stakeholder requirements [15] for the end product). This confirmation is considered a part of validation activities; however, there is a lack of clear agreement across various standards and authorities on this separation of verification and validation.

References for Appendix A

[1] Merriam-Webstert.com, "Merriam-Webster Dictionary,"<http://www.merriam-webster.com>, October 1, 2012.

[2] Wikipedia.org, "Aliasing," <http://en.wikipedia.org/wiki/Aliasing>, October 16, 2012.

[3] Institute of Electrical and Electronics Engineers, "The Authoritative Dictionary of IEEE Standards Terms," IEEE Standard 100-2000, 7th edition, 2000.

[4] Merriam-Webstert.com, "Course,"<http://www.merriam-webster.com/dictionary/course>, October 15, 2012.

[5] Bass, Clements, Katzman, "Software Architecture in Practice (2nd edition)" Addision-Wesley 2003; quoted at the URL: http://www.sei.cmu.edu/architecture/start/glossary/moderndefs.cfm

[6] Schneider, F., "Understanding protocols for Byzantine clock synchronization" Dept of Computer Science, Cornell University, Ithaca, New York, Technical Report # 87-859, August 1987.

[7] U.S. Nuclear Regulatory Commission, “Research Information Letter 1001: Software-related uncertainties in the assurance of digital safety systems Expert Clinic Findings, Part 1”, ADAMS Accession Number ML1035402040, January, 2011.

[8] Goldreich, Obed, “Computational Complexity: A Conceptual Perspective,” ISBN 978-0-521-88473-0, Cambridge University Press, May 2008.

[9] Cobham, Alan, "The intrinsic computational difficulty of functions", Proc. Logic, Methodology, and Philosophy of Science II, North Holland, 1965.

[10] ISO/IEC/IEEE 15288 Systems and Software Engineering—System Life Cycle Processes

[11] RTCA DO-254/Eurocae ED-80 Standard, “Design Assurance for Airborne Electronic Hardware,” Radio Technical Commission for Aeronautics/EUROCAE, April 19, 2000.

[12] Merriam-Webstert.com, “Contribute,”<http://www.merriam-webster.com/dictionary/contribute>, October 15, 2012.

[13] ISO/IEC TR 15026-1:2010 Systems and software engineering – Systems and software assurance – Part 1: Concepts and vocabulary, revised as ISO/IEC DIS 15026-1:2013

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

[15] Institute of Electrical and Electronics Engineers, "IEEE Standard for System and Software Verification and Validation,” IEEE Standard 1012-2012, IEEE Computer Society, 2012.

[16] Institute of Electrical and Electronics Engineers, “IEEE Guide for Developing System Requirements Specifications,” IEEE Standard 1233-1998, 1998.

[17] International Electrotechnical Commission, “Information Technology – Vocabulary – Part 1: Fundamental Terms,” ISO/IEC 2382-1:1993, 1993.

[18] International Electrotechnical Commission, “International Electrotechnical Vocabulary, Chapter 191: Dependability and Quality of Service,” IEC 60050-191:1990-12, 1st edition, 1990.

[19] BusinessDictionary.com, “Design Defect,” <http://www.businessdictionary.com/definition/design-defect.html>, December 17, 2010.

[20] WordNet, “Feasible,” Princeton University, <http://wordnetweb.princeton.edu/perl/webwn?s=feasible>, December 17, 2010.

[21] U.S. Department of Transportation, Federal Highway Administration, “Feasible,” <http://www.fhwa.dot.gov/environment/sidewalks/appb.htm>, December 17, 2010.

[22] Georgetown University, “Feasible,” <http://uis.georgetown.edu/departments/eets/dw/GLOSSARY0816.html>, December 17, 2010.

[23] AviationGlossary.com, “Contributory Hazard,” <http://aviationglossary.com/aviation-safety-terms/contributory-hazard/>, October 15, 2012.

[24] FAA System Safety Handbook, Chapter 7: Integrated System Hazard Analysis, December 30, 2000.

[25] Wikipedia.org, “Separation of Concerns,” <en.wikipedia.org/wiki/Separation_of_concerns>, October 15, 2012.

[26] Scholarpedia.org, “State Space,” <http://www.scholarpedia.org/article/State_space>, October 15, 2012.

[27] Institute of Electrical and Electronics Engineers, “Systems and Software Engineering – Software Life Cycle Processes,” IEEE Standard 12207-2008, 2008.

[28] U.S. Department of Defense, “Standard Practice for System Safety,” MIL-STD-882E, May 11, 2012.

[29] International Organization for Standardization, “Road Vehicles—Functional Safety—Part 1: Vocabulary,” ISO/DIS 26262-1, 1st edition, 2009.

[30] Chris Eckert, Apollo Associated Services, LLC, “Identification and Elimination of Systemic Problems,” Proceedings of the Society of Maintenance and Reliability Professionals Annual Symposium, St. Louis, MO, October 20–22, 2009.

[31] ISO/IEC/IEEE 24765 Systems and software engineering – vocabulary, 2010

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

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

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

Appendix B: Technical Peer Review Process

The technical peer review process was designed and executed in iterative phase-wise reviews with the purpose of acquiring knowledge outside the nuclear power plant (NPP) domain relevant to understanding the evaluation of an applicant's hazard analysis (HA) of a digital instrumentation and control (DI&C) system for safety functions in a NPP.

The Office of Nuclear Regulatory Research (RES) employed the services of Safeware Engineering Corporation (SEC) as a neutral elicitation agent. SEC identified nine experts spread across safety-critical software and systems research experience outside of the commercial NPP industry (e.g., space exploration, military defense, aviation industry).

The objective of the technical peer reviews was to obtain high quality professional engineering and scientific services from the contractor to integrate into RIL-1101 relevant knowledge from respective experts. The technical review meant that the expert provided the content needed to bring the report to the expert's expectation/standard, along with an explanation/justification of the modification or addition.

Technical Peer Review of RIL-1101

The technical reviews were designed to do the following:

a) Provide a marked-up copy of RIL-1101 (incorporated by reference), with tracked proposed changes from each expert engaged in that stage;
b) Supporting references (incorporated by reference);
c) Other supporting rationale or explanation provided by the expert; results of discussions with individual experts;
d) Supporting examples or case studies in the expert's experience or research to support an assertion or guidance item (e.g., through abduction or induction or other manner of generalization applicable to the scope of RIL-1101).
e) Consolidation and organization of the information received from various experts, in a manner conducive to understanding, resolving and reconciling the differences and assessing and improving the degree of validity of the affected content;
f) Results of discussions with group of experts with differing positions on the same issue;
g) Recommendations for changes where high degree of consensus is reached;
h) Identification of items on which adequate consensus has not been reached and the reasons for the differences.

The technical reviews were performed at iteratively evolving stages of RIL-1101 (see Review Stages section below):

- Written comments as described above,
- Followed by teleconferenced walk-through and discussion, and
- Followed by a face-to-face discussion with the NRC staff, where remote communications and interaction were not adequate to understand and resolve the issues.

Each execution of the individual technical reviews was treated as a knowledge-acquisition cycle from which results were integrated into the development of RIL-1101 at each review cycle.

References for Appendix B

[1] U.S. Nuclear Regulatory Commission, "Digital Instrumentation and Control – Technical Engineering Services," Statement of Work for Commercial - V6065, March 2012.

Appendix C: Evaluating Hazard Analysis - State of the Art

The scope[108] of this appendix is limited to the scope of RIL-1101, especially analysis of contributory[109] hazards in digital safety systems for NPPs, which are rooted in systemic causes. For example, it does not discuss techniques or aspects for analysis of systems with a mix of safety and non-safety functions (mixed-criticality systems) or analysis of hazards from random hardware failure. Whereas almost all the surveyed publications cover mixed-criticality systems, this appendix maps the extracted information into its narrower scope. For example: (1) only a relevant subset of HA activities is extracted; (2) the starting point of hazard analysis is "loss/degradation of an allocated safety function, rather than the unwanted release of radioactivity.

C.1 Reference model for hazard analysis: Vocabulary

The vocabulary in this appendix is defined in Appendix A. Following is an explanation of the usage context. A hazard is potential for harm, as defined in Appendix A, and elaborated as follows, "an intrinsic property or condition that has the potential to cause harm or damage." In the scope of RIL-1101, the context of "the intrinsic condition" is a safety related system (or its element) being analyzed and dependency on its environment with the potential to cause harm. In other words, a hazard is a state[110] of the object (of analysis) together with its environment, which has the potential to cause harm.

C.1.1 Dependency

Any factor on which an identified hazard depends (or by which it is influenced) is a contributory hazard. Dependency may be through many kinds of coupling; examples:

1. Functional relationship;
2. Control flow;
3. Data or information;
4. Sharing or constraint of resources;
5. Goals;
6. States or conditions (e.g., in the controlled processes; in the supporting physical processes);
7. Conceptual[111] dependency;
8. Explicit preference-ordering;
9. Some unintended, unrecognized form of coupling.

For a system of the kind in RIL-1101 focus, dependencies are not simple chains or trees, but a network (also known as graph); for example:

[108] Thus, the definitions and descriptions are much more narrowly focused than in more broadly applicable publications on hazard analysis.

[109] IEEE1012-2012 [1] introduces the notion of contributory hazards, e.g. software and hardware contributions to system hazards.

[110] Annex J.1 in [1] "…determine whether the contributing conditions to a hazardous state are possible."

[111] Conceptual independence corresponds to functional diversity in [2].

- There are feedback paths.
- The same factor may occur in many places in the network (i.e., common causes).

C.1.2 Object of analysis

Referring to the reference model for system integration levels depicted in Figure 4 of [1], the object of analysis may be any of the following:

1. A work product such as the following:
   1.1. A complete safety system such as a reactor protection system (RPS).
   1.2. One of its four identical divisions; (information source: system architecture).
   1.3. An element responsible for the voting logic; (information source: system architecture).
   1.4. A system at a lower level of integration; (information source: system architecture).
   1.5. The finest-grained component in the integration hierarchy; (information source: software architecture; hardware architecture).
   1.6. An object in the environment of the object being analyzed, on which the latter depends; (information source: NPP-wide I&C architecture).
   1.7. Result of an intermediate phase to produce any of the above; (information source: development lifecycle model).
2. A process activity producing a work product mentioned above; (information source: process activity model).
3. A resource used in a process activity mentioned above; (information source: process activity model). See in RIL-1101 Figure 4.
4. Any other object in a path of contributory hazards.

The dependency network of the top-level system provides an organizing framework for these objects. For each object, the starting point of its HA would correspond to the derived requirements assigned to it, its boundary with respect to its environment, its relationship to its environment, and associated assumptions. If HA of different objects is occurring concurrently (e.g., impact of changes), based on assumptions about their place and relationships in the dependency network, then, for implications of these assumptions, see in RIL-1101 Table 2, H-culture-12; Table 4, H-ProcState-4; Table 8, H-SR-12-14; Table 9, H-SRE-2G2; Table 13, H-SAE-1G1, H-SAE-7.1.

Hazard analysis (HA) of an object is the process of examining the object throughout its lifecycle to identify hazards (including contributory hazards), and requirements and constraints to eliminate, prevent, or otherwise control these hazards. In terms of clause 4h in [3] quoted below, a “condition having the potential for functional degradation of safety system performance” is a hazard and a “provision … incorporated to retain the capability for performing the safety functions” is a requirement or constraint to eliminate, prevent or otherwise control the hazard.

> Clause 4 and sub-clause h in [3] A specific basis shall be established for the design of each safety system of the nuclear power generating station. The design basis shall also be available as needed to facilitate the determination of the adequacy of the safety system, including design changes. The design basis shall document as a minimum …:
>
> h) The conditions having the potential for functional degradation of safety system performance and for which provisions shall be incorporated to retain the capability for performing the safety functions …

C.2 Reference model for hazard analysis in development lifecycle

Hazard analysis of a digital safety system is part of its safety analysis activities, which are independent from the mainstream system engineering activities, within which also some form of HA and V&V occurs. Nevertheless, the independent HA is interrelated with associated systems engineering activities, as depicted in Figure 1 and charted in Table 20.

In the context of hazards contributed through engineering deficiencies, a contributor may be detected and controlled in (a) the mainstream system development, which includes some form of HA [4] and V&V; (b) independent V&V processes; or (c) independent HA. In general, the higher the quality of the upstream processes, the smaller will be the hazard space downstream, and the lower will be the amount of hazards detected downstream. On the other hand, ill-controlled upstream processes could render downstream V&V and HA infeasible. Recognizing the wide variation in the practice of upstream system engineering, for the purpose of consistent comprehensible concise treatment of the inter-relationship of HA with the other processes, the state-of-the-art in system and safety engineering is used as a baseline and reflected in the lifecycle reference model, depicted in Figure 1. The reference model is derived from [1] for integrity level 4. Thus, the independent HA activities are characterized under the following premises:

1. Mainstream system development activities are performed in accordance with the specifications of their respective processes.
2. Resources used in these development activities are qualified to meet their respective specified requirements or criteria.
3. V&V processes fulfill the objectives stated in Section 1.4 of [1].
4. Verification activities (on the object of verification) confirm that the requirements specified for that object are satisfied.

   4.1. Anomalies are detected as early in the lifecycle as possible, in accordance with [1].

   4.2. Detected anomalies are resolved in accordance with [1]
5. Supporting audits of the process activities in execution examine whether these activities are being performed in accordance with their specifications, using resources that conform to their respective requirements. Deficiencies are corrected promptly.
6. Mainstream validation activities confirm that the various specifications collectively satisfy the requirements intended from the NPP level safety analysis.
7. The “object” of analysis has passed its V&V criteria.

Under these premises, independent HA activities provide an independent search for the remaining “conditions having the potential for functional degradation of safety system performance” (known as hazard identification) and seek their control (e.g., avoidance or elimination) through corresponding requirements and constraints. This search starts from the safety function of concern, first identifying the direct hazards and, then, for each hazard, progressing “upstream” through the dependency paths to identify the contributory hazards. The independent HA perspective is broader than the mainstream activities; for example, it may re-examine:

- Interpretations of a requirement specification;
- Flow-down of derived requirements and constraints;

- Flow-down of quality requirements[112];
- Premised validity of the process specifications and resource qualification criteria;
- Other assumptions.

To the extent that the premises under which the independent HA activities are characterized herein are not satisfied, the difference results in additional burden on these HA activities, requiring commensurate additional skills and effort.

A regulatory review of HA may be viewed as yet another round of independent HA. Thus, the review activities follow the same pattern.

[112] These are also known as “non-functional” requirements.

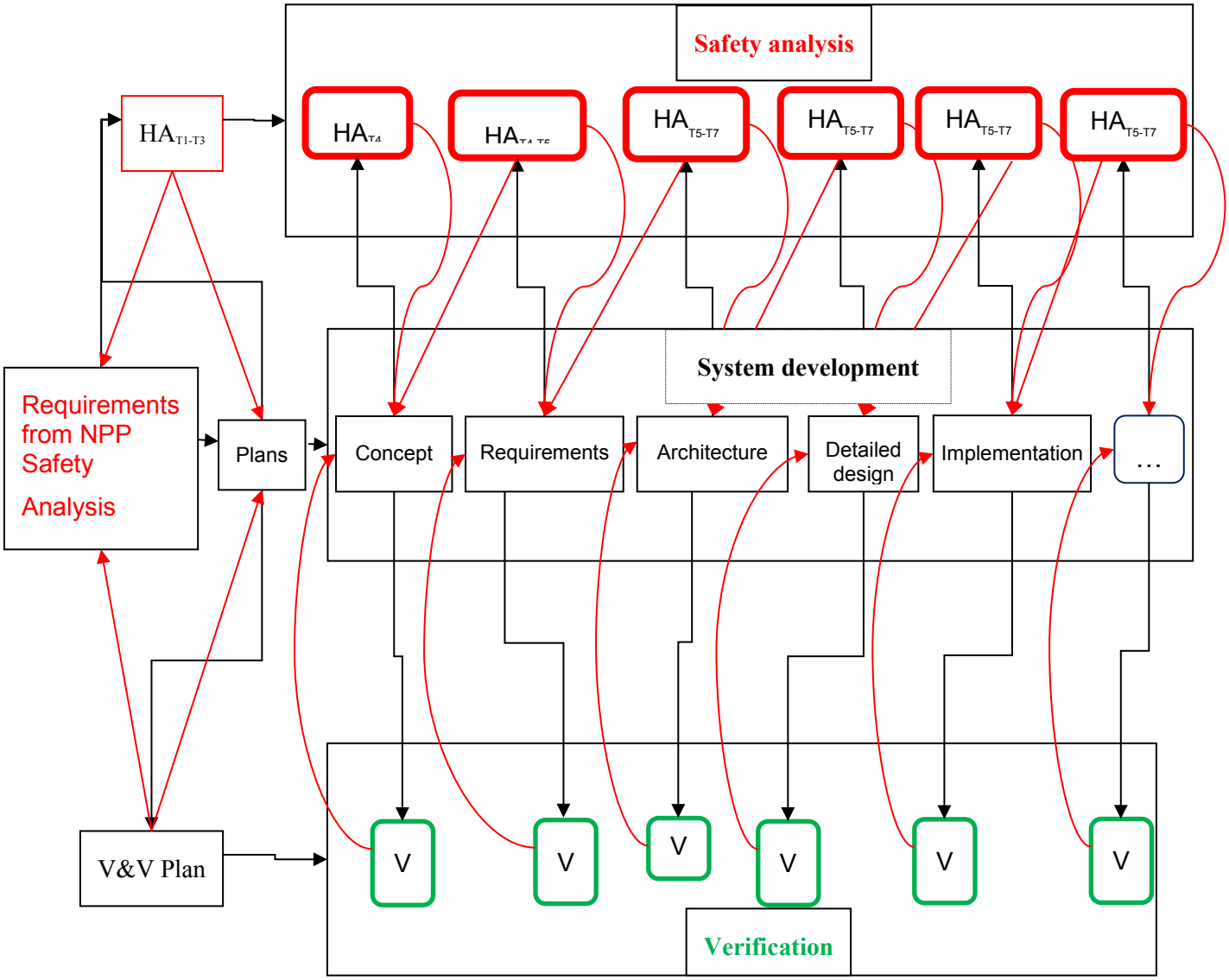

Figure 9: Hazard analysis in relation to development lifecycle and verification activities

C.3 HA tasks – an example set

Referring to Table 20, tasks T1-T3 start in the planning phase of the system engineering lifecycle; however, at every change, the plans are reviewed to identify corresponding changes needed.

Task T4 is started in the concept phase of the system engineering lifecycle. In a "green-field" concept, the information available may only be a functional concept. Yet, it is sufficient to develop the questions to be addressed from the HA perspective, accomplished through the "hazard logging" process. In this case, task T4 may be iterated many times, as the concept evolves. Systematized management of change and configuration (e.g., through minor or internal version identifiers) enables recorded, track-able rationale underlying the evolution path. In a modification of an existing NPP, the concept may be much more developed (e.g., a proposed NPP-level I&C re-architecture), enabling more detailed investigation for the identification of (contributory) hazards.

When the system concept and requirements specification become stable, task T4 transitions into T5, at the start of which, the term "object" refers to the system requirements specification (corresponds to task 203 in [5]). Tasks T5 and T7 are iterated as the system architecture evolves. The iterations include task T6, when a lower level of integration is identified in the system architecture.

Table 20: HA activities and tasks - a reference model

HA activity / task	Input	Output	Remarks. References.
T1. Generate baseline HA plan for all lifecycle phases.	1. Concept [1], incl. interactions with and dependencies on its environment. 2. Requirements from NPP level safety analysis. 3. Premises & assumptions upon which the expected outcome depends, incl. conditions & modes of operation and maintenance. 4. Plan to validate assumptions. 5. Consequences of behavior shortfalls, incl. invalid assumptions/premises. 6. Overall V&V plan, incl. HA. 7. Mainstream development plan. 8. Corresponding information about or from entities in the dependency paths (e.g., up the supply chain).	Baseline[113] HA plan.	Adapted from [1] Table 1a Tasks 7.1:1-4 and Task 101.2.2 in [5].
T2. Identify dependencies of HA plan (e .g. other information; resources; dependencies on supply chain)		Dependencies of plan.	Adapted from: [1] Table 1a Tasks 7.1:1-4; [5].
T3. Evaluate other plans, following the dependencies identified above. T3.1. Coordinate information exchanges with HA activities (e.g., timing; semantic compatibility; format).		1. Evaluation report. 1.1. Deficiencies. 1.2. Changes needed. 1.3. Request for additional information (RAI).	Adapted from [1] Table 1a Tasks 7.1:1-4. 7.4, 7.5.
		2. Rejection or Acceptance (incl. phase-advance clearance)	Adapted from [1] Table 1a Tasks 1-4.
		3. Revision to HA plan as needed.	Adapted from [1] Table 1a Tasks 7.1:1-4.
T4. Understand HA-relevant characteristics of the object to be analyzed; examples: 1. Differences from previously licensed systems. 2. Exposure to unwanted interactions.	Items above + 9. Other requirements allocated to the object. 10. Non-safety related constraints on the object. 11. Relationship with NPP-wide I&C architecture.	1. Revision to HA plan. 2. Addition to hazard log. [15] 3. Change needed; examples: 3.1. Making assumptions explicit;	Adapted from [1] Table 1a Tasks 7.2:(1)a, f, g), (2)b,d), (3)a,b) and Tasks 201-202 in [5],

[113] While mainstream HA producrs the baseline, independent HA identifies changes needed.

3. Presence of functions not needed for the primary safety function. 4. Division of work and communication challenges across organizational units/interfaces. 5. Compatibility of lifecycle models, processes, information-exchange interfaces, etc. 6. Qualification and compatibility of tools across these interfaces. 7. Compatibility of conditions of use for reused objects. 8. Correct, complete flow-down or decomposition or derivation of requirements. 9. Identification of dependencies (e.g., feedback paths; hidden or obscure couplings). 10. Premises and assumptions – explicit and implicit. 11. Other challenges to analyzability.	12. Distribution of responsibilities across organizational units/interfaces. 13. Provisions for information exchange across organizational units/interfaces. 14. Lifecycle models; processes; resources (e.g., tools; competencies); information exchange interfaces. 15. Identification of reused objects and conditions of use. 16. Explicit record of dependencies. 17. Prior HA results, if any.	3.2. Improvement in knowledge of dependencies. 3.3. Making lifecycles, processes compatible; 3.4. Making information-exchange interfaces compatible; 3.5. Consistency across automation and human roles/ procedures. [7] 3.6. Qualification of reused objects (e.g., tools); 3.7. Change in allocation of a requirement; 3.8. Other constraints; 3.9. Other derived requirements. [12]; 4. RAI	
T5. Analyze object[114] for (contributory) hazards. See corresponding section and table in RIL-1101. For a safety system or its element, it includes, for example, search for: 1. Single point failure; 2. Common mode dependency; 3. Common cause dependency.	Items above + Information specific to object of analysis (see Section C.1.2).	1. Addition to hazard log. 2. Change needed. Examples: 2.1. See in T4; 2.2. Derived requirement (on process) to prove that a contributing hazard cannot occur. 2.3. Derived requirement or constraint on object. 3. Rejection | Acceptance (incl. phase-advance clearance) 4. Revision to HA plan as needed 5. RAI	Adapted from [1] Table 1a Tasks 7.1:5-6; Tables 1b and [1]. [14]
T6. Integrate analyses from lower levels in the integration hierarchy and contribution paths up to the top-level analysis.	Items above + information needed about inter-object dependencies for overall system HA	As in T5.	Adapted from [1] Table 1a Task 7.1:7; [1].
T7. Analyze change proposal (e.g., hazard control proposal).	Change proposal, including information on which it depends (e.g, items listed above).	As in T5.	Abstracted from [1]

[114] Examples of objects: Work product from any phase in the development lifecycle; Work product for the top-level digital safety system; some element in a lower level of integration; associated processes; associated resources; any other entity in the dependency paths (e.g., in the supply chain).

C.3.1 Evaluating the quality of HA output

The quality of the HA output depends upon three major factors:

1. Competence – see Section C.4.
2. Quality of the input(s) – see Section C.5.
3. Technique – see Section C.6.

Evaluation of the HA plan is based on the degree to which the planned HA fulfills the following objectives:

1. Identify all hazards.
   1.1. Identity the constraints on the system and its environment, which would enable item 1.
2. Identify all contributory hazards.
   2.1. Identity the constraints on the system and its environment, which would enable item 3.
3. Identity the constraints needed to control the identified (contributory) hazards.

Consequently, evaluation of a selected HA technique is based on its ability to fulfill the objecitves stated above and identifying the associated critical conditions, namely:

1. A specification of the competence required to apply the technique, such that the competence can be evaluated with consistency.
2. A specification of the information required to apply the technique, such that the object of analysis can be evaluated with consistency.

C.3.2 Hazard logging and identification

Hazard identification, especially in the concept phase, requires extra-ordinary individual capabilities, teamwork, and a conducive organizational culture. If any analyst or contributor to HA perceives a safety concern, a hazard, or a contributory hazard, the individual is encouraged to express it. The expressed item is recorded in a "hazard log" without immediate evaluation. Sometimes, a team engages in brainstorming to stimulate thought and encourage expression. The "hazard log" [15] is a means of tracking an item from initial expression to final disposition and closure. An "entry" is never deleted. All the related information may be in a single document or it may be distributed across a set of linked databases; in any case, an analyst is able to make an entry readily. Examples of related information include the following:

1. Information to identify the logged item:
   1.1. Item identifier;
   1.2. Descriptive title;
   1.3. Originator;
   1.4. Origination date;
   1.5. Description;
   1.6. Perceived consequence/effect of inaction;
2. Information to track progress:
   2.1. Action plan (from origination to closure);
   2.2. Action assignee(s);
   2.3. Status of progress in the action plan (e.g.,
      2.3.1. Identified change needed to eliminate hazard);
   2.4. Basis to allow closure (e.g.

2.4.1. Evaluation revealed that hazard control is already in place.
2.4.2. Evaluation resulted in restatement of the hazard (another entry in the hazard log);
2.4.3. Addition of a constraint or derived requirement in the system engineering activities;
2.5. Date of closure;
2.6. Name and Signature authorizing closure.

Every addition or modification of a constraint or (derived) requirement is a configuration controlled item with associated change controls.

When the object is the overall system, the corresponding HA task is the exercise of the selected HA technique (see Section C.6) on the information available about the object (see Section C.5). Execution of this process may assist in the evaluation of some other item in the hazard log; or may raise a new concern, which is then entered in the hazard log.

C.3.3 Evaluation of a logged hazard

Whereas published standards and handbooks (whose scope includes mixed-criticality systems) suggest evaluation in terms of levels of severity and likelihood of occurrence, in the RIL-1101 context, the severity of the loss of a safety function is of the highest level and, for systemic causes, the analysis first seeks their correct identification and then, pursues their elimination or avoidance, as explained next.

In practice, a "quick" filtering or screening evaluation (e.g., see 2.4.1-2.4.2 above) is performed on each logged item, before delving deeper. If an accurate dependency model is available, the evaluation seeks to fit the logged item in the dependency model. The search may reveal that the dependency model is inaccurate (requiring change) or that the logged item is not a (contributory) hazard (leading to its closure). When the logged item is matched to an object in the dependency network (i.e., its sequence in the contributory path is found), a corresponding HA task is formulated and sequenced in accordance with its place in the contributory path.

As the evaluation of a logged item progresses, it may expose inadequacies or uncertainties in the information about the object being analyzed. Figure 10 depicts a structure for reasoning (adapted from [16]) about these uncertainties. Suppose that the HA team is considering an assertion that the result of their work will control the logged (contributory) hazard. Then, the team clarifies its reasoning[115] through discussion, evoking challenges to the assertion and rebuttals to the challenges. The discussion may also reveal inconsistencies in the reasoning. In this manner, the team identifies factors affecting the validity of their assertion. Qualifiers are associated with the assertion; for example:

1. Condition(s) under which the assertion is supported.
   1.1. Uncertainties may be stated as assumptions, for which the truth has to be validated.
   1.2. Changes needed may be stated as constraints to be satisfied.
2. Degree or strength of the assertion: {Strong …. Weak}

[115] It is labeled "warrant" in [14].

The results are recorded, showing how the assertion is supported by the evidence[116], identifying the inference rule to assert the evidence-assertion link, and the technical basis for the rule such as a causal model[117].

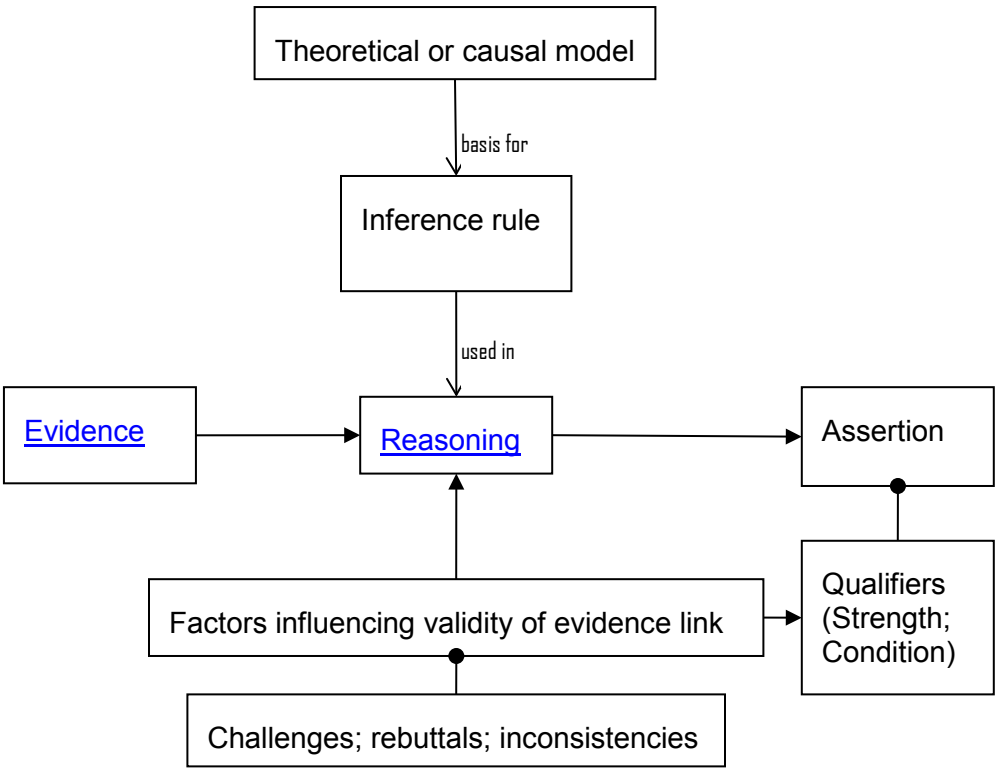

Figure 10: Structure to reason about the evaluation of a (contributory) hazard item

C.4 Effect of competence on quality of HA work products

When HA is performed on an early-stage concept, with little explicit information in the concept, the "competence" factor (see Section C.3.1) is most dominant. For example, the analyst has to elicit information about assumptions and dependencies through systematic enquiry, devised for the circumstances. Based on this information, the analyst would have to construct an analyzable model of the dependencies (e.g., control structures, showing feedback paths, interactions, and nested levels). These activities require extremely high competence. For an approach to competence management, see [17], in which reference 7 is a technical competence framework developed through wide consultation in the UK.

Competence is a critical factor - see in RIL-1101 Table 1 items H-0-2G{0, 1, 2}, H-0-3G1; Table 2 H-culture-6G3, Table 9 H-SRE-1G{1,2,3}. Competence to perform HA of an NPP digital safety system includes a complement of the following:

[116] It is labeled "grounds" in [14].

[117] It is labeled "backing" in [14].

1. Proven self-learning[118] 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.
   1.4. Reasoning capability (see Figure 10);
      1.4.1. Ability to abstract and generalize from one context and apply to another.
      1.4.2. Ability to recognize fallacies in some chain of reasoning.
2. Continuing update of professional knowledge through training; examples:
   2.1. Application domain: How an NPP works (energy conversion from fuel to power on the grid); heat exchange; critical functional elements, processes and process state variables in an NPP and their inter-dependencies; associated (contributory) hazards; study of operating experience (event reports; root cause analysis reports).
   2.2. Industrial automation domain: Elements for sensing, actuation, computation; control logic; communication; software/firmware; power; associated (contributory) hazards; study of operating experience (event reports; root cause analysis reports).
   2.3. Science and engineering of distributed systems, including computation, communication.
   2.4. Hazard and safety analysis and assurance methods and techniques for such systems.
3. Experience:
   3.1. Working under the guidance of an expert in hazard analysis.
   3.2. Working independently in the analysis of systems of similar complexity and criticality.
4. Interpersonal skills:
   4.1. Ability to communicate effectively, objectively with a wide range of stakeholders.
   4.2. Ability to elicit information needed.
   4.3. Ability to listen for understanding and learning from others.
   4.4. Ability to explain one's reasoning (see Figure 10) to others.
   4.5. Teamwork.

C.5 Quality of information input to HA at each development phase

Table 21 provides a broad-brush characterization of the quality of the work products (in terms of information richness) available for HA. For each major lifecycle phase work product, Table 21

[118] 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.

compares characteristics in common practice with state-of-the-practice (best in class), and state-of-the-art (leading-edge implementations, not yet scaled up).

Table 21: Characterization information richness in phase work products

Row ID	Work product of lifecycle phase	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. [18]	Use case scenarios [19].
			SpecTRM-RL [20]	Framework for specification & analysis [21].
			Requirements engineering support in Naval Research Labs [22]. Requirements tables as used for Darlington NPP [23][24]. Models to support mechanized reasoning. Examples: SysML [25].	
2	Plans {Safety plan; V&V plan; HA plan}	Low level of detail; relatively late in the lifecycle.	V&V plan [1] Safety plan [26]-[28]	Integrated safety and security plan.
3	Concept	Combination of (a) block diagram without semantics on the symbols and (b) textual narrative	Models to support mechanized reasoning [29]. (See note 1) SysML [25]; AADL [30] Extended EAST-ADL [31]	META [32]
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 [32]
6	Requirements for software in digital safety system	See row 1	[29][33][34]	See row 1
7	Architecture for software in digital safety system	See row 3	See row 3. MASCOT [34] AADL [30]	META [32]
8	Detailed design of software	For application logic: Function block diagram [35]. For platform software: Combination of (a) block diagram without semantics on the symbols and (b) textual narrative.	SPARK [36][37]	META [32] 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 [38][39] Language for programming FPGAs [40]	Auto-generation from detailed design.

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

C.6 Hazard Analysis Techniques – useful extractions from survey

The selection and role of HA techniques (the third factor influencing the quality of an HA product mentioned in Section C.3.1) will depend upon the nature of the system to be analyzed and the quality of the information contained in the various intermediate work products, characterized in Section C.5.

Table 22 summarizes some applicable techniques surveyed. As difficulties and limitations were encountered in the earlier techniques (such as those in the first three rows of Table 22), these techniques were extended, adapted and transformed into newer techniques (such as the ones in the last three rows of Table 22); the references for the latter describe some of the difficulties and limitations encountered in using the earlier techniques. The "salient feature(s)" column identifies concepts found useful. However, the adaptations devised to evolve newer techniques require extraordinary ingenuity; utility of the adaptations is very dependent upon the skills of the analysts.

When HA is applied to an early concept phase, it is called preliminary hazard analysis (PHA) [41][42].

For a broad survey of HA techniques, see [7][43][44], and for additional guidance, see [45]-[49]. For a tutorial overview of HA in relation to safety critical system development, see [51]. These references are not included in Table 22, if technique-specific references are listed.

Table 22: Salient features of techniques relevant to NPP digital safety systems

HA technique		Reference(s)	Salient feature(s)
Acronym	**Expanded name**		
HAZOP(S)	Hazard and operability studies	[8]	Concept of using teamwork, facilitated by HAZOP process expert. Systematizing enquiry through key words. Systematizing understanding effects through understanding the associated deviations.
FTA	Fault Tree Analysis	[52][53][54]	Representation and understanding of fault propagation paths, when the paths are branches of a tree.
DFMEA	Design Failure Mode and Effects Analysis	[55][56][57] [58]	Representation of faulted behavior of a hardware component for understanding its effect, without requiring knowledge of its internals.
FFMEA	Functional Failure Mode and Effects Analysis	[57][69]	Understanding effect of unwanted behavior of a function of the system, without requiring knowledge of its internals. Useful in concept phase.
FuHA	Functional Hazard Analysis	[7]	
FHA	Fault Hazard Analysis	[43][46] [49]	
CCA	Cause Consequence Analysis	[43][49]	Concept of using causality model to understand fault propagation paths.
W/IA	What if analysis	[47][49]	
CCFA	Common Cause Failure Analysis	[43][46] [49]	
HACCP	Hazard Analysis & Critical Control Points	[50]	Concept of focusing on critical process variables that affect the outcome.
SHARD	Software hazard analysis and resolution	[10]	Adaptation of HAZOP to software, through customization of the key words.
FPTN/FPTC	Fault propagation and transformation network/calculus	[59]	Representation and analysis of fault propagation, when the faults are transformed during propagation, and when there are feedback paths, supporting mechanized traversal and reasoning.
DFM	Dynamic Flowgraph	[64]-[66]	Behavior modeling in the finite state machine paradigm:

	Method		• Facilitates analysis of interactions. • Facilitates analysis of dynamic (time-dependent) behavior across different elements in a system • Facilitates analysis of interactions between the digital safety system and it environment. • Allows mechanized traversal and reasoning. • Allows mathematical underpinning.
STPA	System-Theoretic Process Approach	[74]-[76]	• Applicable at concept phase (does not require a finished design). • Applicable to understanding of organizational culture level systems.

HAZOP has been adapted to analyze software [8], and this adaptation has been extended to data flow oriented software architecture [10], and, later, extended to systems with feedback and systems in which the initial fault is transformed into other faults as it propagates [59][59]. These concepts and principles have influenced the AADL [30] error annex, supporting analysis of fault propagation. For an indication of promising research to extend AADL for hazard analysis, see [71].

Recently, a technique similar to the adaptations of HAZOP mentioned above, namely STPA, has been demonstrated in NPP applications [74][75][76].

For a comparative experimental study of six techniques, see [74].

If HA is performed on a state-of-the-practice or state-of-the-art work product, such as the ones shown in Table 21, and if all behavior-influencing assumptions and dependencies were already explicit in a system architecture model, the search for (contributory) hazards could be automated [61]-[65], reducing the dependence on extremely high competence. However, model-based approaches introduce their own contributory hazards [63], to analyze which highly specialized competence is needed.

For adaptation of the concepts in [11], [59] and [59] for HA of device interfaces and, then, HA of operating systems, see [66]-[69].

For an adaptation of the concepts in [59]-[60] to address the fault propagation problem for FPGAs, see early experimental work reported in [70].

For an example of showing freedom from exceptions in software implementations (which are contributing hazards), in addition to showing conformance to specifications, see [37].

For an example of analysis for hazards contributed through timing aspects of multi-core computing processor resources, see [72].

Static analysis tools, such as [37] identify data, information and control flow dependencies in software.

For emerging guidance on HA of complex hardware, such as FPGAs, see [71]. For ongoing developments in this field, track [72].

C.7. References for Appendix C

[1] IEEE Standard 1012-2012, “IEEE standard for system and software verification and validation,” March 29, 2012.

[2] NUREG/CR-7007, “Diversity strategies for nuclear power plant instrumentation and control systems” 2010.

[3] IEEE Standard 603-2009, “IEEE standard criteria for safety systems for nuclear power generating stations” 2009.

[4] Joint Software System Safety Committee, “Software System Safety Handbook – A Technical & Management Team Approach,” December 1999. URL: http://www.system-safety.org/Documents/Software_System_Safety_Handbook.pdf

[5] MIL-STD-882E, “Standard Practice for System Safety,” U.S. Department of Defense, May 11, 2012.

[6] European Committee for Electrotechnical Standardisation (CENELEC), EN 50126, Part 1: 1999, Railway applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS). (A new version is soon to be released; I assume that it will not have changed much in concept.)

[7] Society of Automotive Engineers (SAE), ARP-4761 – Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, 1996. (This standard contains a description of Functional Hazard Analysis (FHA) and of Preliminary System Safety Assessment (PSSA).)

[8] CISHEC (The Chemical Industry Safety and Health Council of the Chemical Industries Association Ltd.), A Guide to Hazard and Operability Studies, 1977.

[9] McDermid, J.A., Nicholson, M., Pumfrey, D.J. & Fenelon, P., (1995), Experience with the application of HAZOP to computer-based systems, COMPASS '95: Proceedings of the Tenth Annual Conference on Computer Assurance, Gaithersburg, MD, pp. 37-48, IEEE, ISBN 0-7803-2680-2.

[10] Redmill F., Chudleigh, M., Catmur J., System Safety: HAZOP and Software HAZOP. John Wiley and Sons Ltd., Chichester, U.K., 1999.

[11] McDermid, J.A. & Pumfrey, D.J., (1998), Safety Analysis of Hardware / Software Interactions in Complex Systems, Proceedings of the 16th International System Safety Conference, Seattle, WA, pp. 232-241, System Safety Society.

[12] IEC Standard 61882, “Hazard and Operability Studies (HAZOP Studies) – Application Guide,” International Electrotechnical Commission, First Edition, 2001.

[13] McDermid J.A., Safety critical software, in: Encyclopedia of Aerospace Engineering, Online, Wiley 2012 (accessible via DOI: 10.1002/9780470686652.eae506).

[14] Hawkins R.D., Habli I., Kelly T.P., The Principles of Software Safety Assurance, in Proceedings of the 31st International System Safety Conference, Boston, MA, International System Safety Society.

[15] SMP 11, MoD Hazard Log Requirements, see: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/27584/SMP11v22final.pdf (last accessed 31st July 2013)

[16] Toulmin, Stephen. The Uses of Argument. Cambridge: University Press, 1958

[17] HSE statement on competency management, see: http://www.hse.gov.uk/consult/condocs/competence.pdf (last accessed 4th August 2013)

[18] 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

[19] 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.

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

[21] 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

[22] 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

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

[24] 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.

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

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

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

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

[29] 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.

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

[31] 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.

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

[33] 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.

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

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

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

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

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

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

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

[41] Gowen, L.D., Collofello, J.S., Calliss, F.W., "Preliminary Hazard Analysis for Safety-Critical Software Sysems," Proceedings from IPCCC'92, 1992.

[42] Safeware Engineering Corporation, "Preliminary Hazard Analysis," < http://www.safeware-eng.com/Safety%20White%20Papers/Preliminary%20Hazard%20Analysis.htm>, December 6, 2011.

[43] Ericson II., C.A., "Hazard Analysis Techniques for System Safety," John Wiley and Sons, August 24, 2005.

[44] U.S. Nuclear Regulatory Commission, "Software Safety Hazard Analysis," NUREG/CR-6430, Washington, DC, February 1996 (Agencywide Documents Access and Management System (ADAMS) Public Legacy Library Accession No. 9602290270).

[45] National Aeronautics and Space Administration, "NASA Software Safety Guidebook", NASA-GB-8719.13, Washington, DC, March 31, 2004.

[46] U.S. Air Force, "The Air Force System Safety Handbook", Kirtland AFB, NM, July 2000.

[47] European Strategic Safety Initiative, "Guidance on Hazard Identification," European Strategic Safety Initiative – Safety Management System and Safety Culture Working Group, March 2009.

[48] Ippolito, L., Wallace, D., "A Study on Hazard Analysis in High Integrity Software Standards and Guidelines," National Institute of Standards and Technology, NISTIR 5589, Gaithersburg, MD, January 1995.

[49] System Safety Society, "System Safety Society Handbook: A source Book for Safety Practitioners," The System Safety Society, 1993.

[50] Hazard analysis and critical control points HACCP principles and application guidelines, URL: http://www.fda.gov/Food/GuidanceRegulation/HACCP/ucm2006801.htm

[51] Johnson, Chris, "Safety Critical System Development", University of Glasgow – Department of Computing Science, Part II of Notes, October 2006.

[52] U.S. Nuclear Regulatory Commission, Fault tree handbook (NUREG 492). URL: http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/sr0492.pdf

[53] NASA, "Fault tree handbook with aerospace applications." URL: http://www.hq.nasa.gov/office/codeq/doctree/fthb.pdf

[54] Park, G.Y., Koh, K.Y., Jee, E., Seong, P.H., Kwon, K.C., and Lee, D.H., "Fault Tree Analysis of KNICS RPS Software," Nuclear Engineering Technology, Vol. 41, No. 4, May 2009.

[55] SAE J1739, "Potential Failure Mode and Effects Analysis in Design (Design FMEA), Potential Failure Mode and Effects Analysis in Manufacturing and Assembly Processes (Process FMEA), 2009" URL: http://standards.sae.org/j1739_200901/

[56] NASA, "Standard for Performing a Failure Mode and Effects Analysis (FMEA) and Establishing a Critical Items List (CIL) (DRAFT): Flight Assurance Procedure (FAP)- 322-209," Nov. 2011, Available: rsdo.gsfc.nasa.gov/documents/Rapid-III-Documents/MAR-Reference/GSFC-FAP-322-208-FMEA-Draft.pdf

[57] P.L. Goddard, "Software FMEA Techniques," Proceedings of the Annual Reliability and Maintainability Symposium, IEEE, 2000, pp. 118–123 Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=816294.

[58] G.-Y. Park, "Software FMEA Analysis for Safety Software," International Conference on Nuclear Engineering, Brussels, Belgium: ASME

[59] Fenelon P., McDermid J.A., Nicholson M., Pumfrey D.J., Towards Integrated Safety Analysis and Design, ACM Applied Computing Review, Aug. 1994.

[60] Wallace M., Modular Architectural Representation and Analysis of Fault Propagation and Transformation, Electronic Notes in Theoretical Computer Science 141 (2005) 53–71.

[61] Hecht, H. and Menes, R., "Software FMEA automated and as a design tool" SAE 08WATC-0023, 2008.

[62] Hecht, H., An, X., Hecht, M., "Computer aided software FMEA for unified modeling language based software."

[63] UML Safety Analysis, see: https://www.ibm.com/developerworks/community/blogs/BruceDouglass/entry/safety_analysis_with_the_uml8?lang=en (last accessed August 2nd 2013)

[64] Garrett, C. and Apostolakis, G., "Context in the risk assessment of digital systems" Risk Analysis Vol. 19 No. 1 1999.

[65] Garrett, C. and Apostolakis, G., "Automated hazard analysis of digital control systems" Reliability Engineering and Safety Society 77 (2002) 1-17

[66] Aldemir, Guarro, et al, "A Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation and Control Systems," NUREG/CR-6985, U.S. Nuclear Regulatory Commission, Washington, D.C. (2009).

[67] McDermid, J.A. & Pumfrey, D.J., (2000), Assessing the Safety of Integrity Level Partitioning in Software, Lessons in System Safety: Proceedings of the Eighth Safety-critical Systems Symposium, Southampton, UK, Ed. Redmill, F. & Anderson, T., pp. 134-152, Springer, ISBN 1-85233-249-2.

[68] Conmy P.M., Crook-Dawkins S.K., A Systematic Framework for the Assessment of Operating Systems, Safety-Critical Systems Symposium, Warwick, UK, February 2004.

[69] Conmy P.M., Safety Analysis of Computer Resource Management Software, PhD Thesis, University of York, 2005.

[70] Conmy P.M., Bate I.J., Component-Based Safety Analysis of FPGAs, IEEE Transactions on Industrial Informatics, Vol 6, No 2, May 2010. pp 195-205

[71] Bozzano M., Cimatti A., Katoen J-P., Nguyen V.Y., Noll T., Roveri M., Safety, Dependability and Performance Analysis of Extended AADL Models, The Computer Journal, Vol. 54 No. 5, 2011

[72] ParMERASA, see: http://www.parmerasa.eu/ (last accessed August 2nd 2013)

[73] iScade, see: http://iscade.co.uk/ (last accessed 31st July 2013)

[74] Torok, R. and Geddes, B. "Systems Theoretic Process Analysis (STPA) Applied to a Nuclear Power Plant," MIT STAMP Workshop, March 26-28, 2013. <http://psas.scripts.mit.edu/home/wp-content/uploads/2013/04/02_EPRI_MIT_STAMP_Mar2013.pdf>

[75] Song, Yao, "Applying system-theoretic accident model and processes (STAMP) to hazard analysis" McMaster University dissertation. URL: http://digitalcommons.mcmaster.ca/opendissertations/6801/

[76] Thomas, J. et al, "Evaluating the safety of digital instrumentation and control systems in nuclear power plants" November, 2012. URL: http://sunnyday.mit.edu/papers/MIT-Research-Report-NRC-7-28.pdf

C.8. Bibliography[119] for Appendix C

[77] Atchison B., The Integration of Safety Analysis and Functional Verification Techniques for Software Safety Arguments, 2004, PhD Thesis, University of Queensland.

[78] Chambers, L., "A Hazard Analysis of Human Factors in Safety-Critical Systems Engineering," 10th Annual Workshop on Safety-Related Programmable Systems (SCS-05), Conference in Research and Practice in Information, Vol. 55, Sydney, Australia, 2005.

[79] Alexander, R., Kelly, T., "Can We Remove the Human from Hazard Analysis," University of York.

[80] ISO/IEC 15026-2:2011, "System and Software Engineering - System and Software Assurance - Part 2: Assurance Case.

[119] References Reviewed but not Yet Cited in Appendix C

Appendix D – Example of dependency not well understood

For Want of a Nail
For want of a nail the shoe was lost. *For want of a shoe the horse was lost.* *For want of a horse the rider was lost.* *For want of a rider the message was lost.* *For want of a message the battle was lost.* *For want of a battle the kingdom was lost.* *And all for the want of a horseshoe nail.*

Appendix E: Hazard Checklists

This Appendix is a collection of checklists from various safety-critical applications of digital systems. This should not be considered an exhaustive coverage of hazards relevant to an NPP DI&C safety system .

E.1 General Hazards Checklist – NASA

Table E.1 is a checklist taken from the NASA Reference Publication 1358 [1], organized by categories of hazard origination and corresponding potential for loss or effect leading to potential loss .

Table E.1: NASA General Hazards Checklist

Hazard Category of Origination	Potential loss or effect leading to potential loss
Acceleration/Deceleration/Gravity	Inadvertent motion Loose object translation Impacts Failing objects Fragments/missiles Sloshing liquids Slip/trip Falls
Chemical/Water Contamination	System-cross connection Leaks/spills Vessel/pipe/conduit rupture Backflow/siphon effect
Common Causes	Utility outages Moisture/humidity Temperature extreme Seismic disturbance/impact Vibration Flooding Dust/dirt Faulty calibration Fire Single-operator coupling Location Radiation Wear-out Maintenance error Vermin/varmints/mud daubers
Contingencies (Emergency Response by System/Operators to “Unusual” Events)	“Hard” shutdown/failures Freezing Fire Windstorm Hailstorm Utility outrages Flooding

	Earthquake Snow/ice load
Control Systems	Power outage Interfaces (EMI/RFI) Moisture Sneak circuit Sneak software Lighting strike Grounding failure Inadvertent activation
Electrical	Shock Burns Overheating Ignition of combustibles Inadvertent activation Power outage Distribution back feed Unsafe failure to operate Explosion/electrical (electrostatic) Explosion/electrical (arc)
Mechanical	Sharp edges/points Rotating equipment Reciprocating equipment Pinch points Lifting weights Stability/topping potential Ejected parts/fragments Crushing surfaces
Pneumatic/Hydraulic Pressure	Over-pressurization Pipe/vessel/duct rupture Implosion Mislocated relief valve Dynamic pressure loading Relief pressure improperly set Backflow Crossflow Hydraulic ram Inadvertent release Miscalibrated relief device Blown objects Pipe/hose whip Blast
Temperature Extremes	Heat source/sink Hot/cold surface burns Pressure evaluation Confined gas/liquid Elevated flammability Elevated volatility Elevated reactivity Freezing

	Humidity/moisture Reduced reliability Altered structural properties (e.g., embrittlement)
Radiation (Ionizing)	Alpha Beta Neutron Gamma X-Ray
Radiation (Non-Ionizing)	Laser Infrared Microwave Ultraviolet
Fire/Flammability—Presence of	Fuel Ignition Source Oxidizer Propellant
Explosive (Initiators)	Heat Friction Impact/shock Vibration Electrostatic discharge Chemical contamination Lightning Welding (stray current/sparks)
Explosives (Effects)	Mass fire Blast overpressure Thrown fragments Seismic ground wave Meteorological reinforcement
Explosive (Sensitizes)	Heat/cold Vibration Impact/shock Low humidity Chemical contamination
Explosives (Conditions)	Explosive propellant present Explosive gas present Explosive liquid present Explosive vapor present Explosive dust present
Leaks/Spills (Material Conditions)	Liquid/cryogens Gases/vapors Dusts—irritating Radiation sources Flammable Toxic Reactive Corrosive Slippery Odorous Pathogenic

	Asphyxiating Flooding Runoff Vapor propagation
Physiological (See Ergonomic)	Temperature extremes Nuisance dusts/odors Baropressure extremes Fatigue Lifted weights Noise Vibration (Raynaud's syndrome) Mutagens Asphyxiants Allergens Pathogens Radiation (See Radiation) Cryogens Carcinogens Teratogens Toxins Irritants
Human Factors (See Ergonomics)	Operator error Inadvertent operation Failure to operate Operation early/late Operation out of sequence Right operation/wrong control Operated too long Operate too briefly
Ergonomic (See Human Factors)	Fatigue Inaccessibility Nonexistent/inadequate "kill" switches Glare Inadequate control/readout differentiation Inappropriate control/readout labeling Faulty work station design Inadequate/improper illumination
Unannunciated Utility Outages	Electricity Steam Heating/cooling Ventilation Air conditioning Compressed air/gas Lubrication drains/slumps Fuel Exhaust
Mission Phasing	Transport Delivery Installation Calibration

	Checkout Shake down Activation Standard start Emergency start Normal operation Load change Coupling/uncoupling Stressed operation Standard shutdown Shutdown emergency Diagnosis/troubleshooting Maintenance

Table E-1: General Hazard Checklist

E.2 Hazard Checklist for General and Energy Sources

Following is checklist from [2] of items to be considered in the HA of an NPP DI&C safety system.

1. Acceleration
2. Contamination
3. Corrosion
4. Chemical dissociation
5. Electrical
   a. Shock
   b. Thermal
   c. Inadvertent activation
   d. Power source failure
   e. Electromagnetic radiation
6. Explosion
7. Fire
8. Heat and temperature
   a. High temperature
   b. Low temperature
   c. Temperature variations
9. Leakage
10. Moisture
    a. High humidity
    b. Low humidity
11. Oxidation
12. Pressure
    a. High
    b. Low
    c. Rapid change
13. Radiation
    a. Thermal
    b. Electromagnetic
    c. Ionizing
    d. Ultraviolet

14. Chemical replacement
15. Shock (mechanical)
16. Stress concentrations
17. Stress reveals
18. Structural damage or failure
19. Toxicity
20. Vibration and noise
21. Weather and environment

The following is a checklist of energy sources, collected from a variety of industrial applications, for consideration in the HA of NPP DI&C safety systems:

1. Fuels
2. Propellants
3. Initiators
4. Explosive charges
5. Charged electrical capacitors
6. Storage batteries
7. Static electrical charges
8. Pressure containers
9. Spring-loaded devices
10. Suspension systems
11. Gas generators
12. Electrical generators
13. Radio frequency sources
14. Radioactive energy sources
15. Failing objects
16. Catapulted objects
17. Heating devices
18. Pumps, blowers, fans
19. Rotating machinery
20. Actuating devices
21. Nuclear

E.3 Hazard Checklist for Semiconductor Manufacturing Equipment

Table E.2 is a set of examples from the semiconductor manufacturing industry [3], organized by categories of sources of hazards and the corresponding potential loss or effect leading to potential loss. It may be useful in HA of NPP DI&C systems, as a checklist from another perspective.

Table E.2: SEMATEC General Hazards Checklist of semiconductor manufacturing equipment

Hazard Source/Description	Potential loss or effect leading to potential loss Effect
Chemical Energy Chemical disassociation or replacement of fuels, oxidizers, explosives, organic materials or compounds	Fire Explosion Non-explosive exothermic reaction Material degradation Toxic gas production Corrosion fraction production
Contamination Producing or introducing contaminants to surfaces, orifices, filters, etc.	Clogging or blocking components Deterioration of fluids Degradation of performance sensors or operating components

Hazard Source/Description	Potential loss or effect leading to potential loss Effect
Electrical Energy System or component potential energy release or failure. Includes shock, thermal, and static.	Electrocution/involuntary personnel reaction Personnel burns Ignition of combustibles Equipment burnout Inadvertent activation of equipment Release of holding devices Interruption of communications (facility interface) Electrical short circuiting
Human Hazards Human hazards including perception (inadequate control/display identification), dexterity (inaccessible control location), life support, and error probability (inadequate data for decision making). Conditions due to position (hazardous location/height), equipment (inadequate visual/audible warnings or heavy lifting), or other elements that could cause injury to personnel.	Personnel injury due to: Skin abrasion, cuts, bruises, burns, falls etc. Muscle/bone damage Sensory degradation or loss Death Equipment damage by improper operation/handling may also occur
Kinetic/Mechanical Energy (Acceleration) System/component linear or rotary motion. Change in velocity, impact energy of vehicles, components or fluids.	Impact Disintegration of rotating components Displacement of parts or piping Seating or unseating valves or electrical contact Detonation of shock sensitive explosives Disruption of metering equipment Friction between moving surfaces
Material Deformation Degradation of material due to an external catalyst (i.e., corrosion, aging, embrittlement, fatigue, etc.).	Change in physical or chemical properties; corrosion, aging, embrittlement, oxidation, etc. Structural failure De-lamination of layered material Electrical insulation breakdown
Natural Environment Conditions including lighting, wind, flood, temperature extremes, pressure, gravity, humidity, etc.	Structural damage from wind Equipment damage Personnel injury
Pressure System/component (e.g., fluid systems, air systems) potential energy including high, low, or changing pressure.	Blast/fragmentation from container over-pressure rupture Line/hose whipping Container implosion System leaks Aero-embolism, bends, choking, or shock Uncontrolled pressure changes in air/fluid systems
Radiation Conditions including electromagnetic, ionizing, thermal, or ultraviolet radiation (including lasers/and optical fibers).	Uncontrolled initiation of safety control systems & interlocks Electronic equipment interference Human tissue damage Charring of organic material Decomposition of chlorinated hydrocarbons into toxic gases Fuel ignition
Thermal High, low, or changing temperature	Ignition of combustibles Initiation of other reactions Expansion/contraction of solids or fluids Liquid compound stratification
Toxicants Inhalation or ingestion of substances by personnel	Respiratory system damage Blood system damage Body organ damage Skin irritation or damage Nervous system effects
Vibration/Sound System/component produced energy	Material failure Pressure/shock wave effects Loosing of parts Chattering of valves or contacts Verbal communications interference Degradation or failure of displays

E.4 Hazards from physical environment of an NPP DI&C safety system

Disruption in or emissions from the environment or physical conditions in the environment may impair a safety function of the analyzed DI&C system in an NPP, e..g:

1. Water in unwanted space
2. Transfer of unwanted energy in various forms; for example:
   2.1. Fire
   2.2. Lightening
   2.3. Heat
   2.4. Light
   2.5. Sound
   2.6. Vibration
   2.7. Radiation
   2.8. Shock
   2.9. Seismic event or effect
   2.10. Tsunami
   2.11. Flooding
   2.12. Electrostatic discharge
   2.13. Electromagnetic interference, causing spurious signal or signal change.
   2.14. Electromagnetic radiation, e.g.:
      2.14.1. Pulse
      2.14.2. Sunspot; solar flare
3. Disturbance in incoming signals
4. Interruption of services (primary; secondary; other forms of back-up) ; for example:
   4.1. Electric power supply.
5. Disturbance in services, propagating to a disturbance in a main signal; for example:
   5.1. Electric power supply.
   5.2. Service water [15]
   5.3. Service air
6. Intrusions through breaches of isolation barriers; for example
   6.1. Cable penetration
   6.2. Other duct penetration
7. Adverse conditions in temperature, pressure, or humidity/moisture; for example
   7.1. Too high
   7.2. Too low
   7.3. Rapid changes

E.5 Hazards contributed by the DI&C system, affecting its environment

Emissions or outputs from or behavior of the DI&C system having an effect on its environment may affect safety adversely; for example:

1. Emission of energy in various forms; for example:
   1.1. Heat
   1.2. Light
   1.3. Sound
   1.4. Vibration

1.5. Electromagnetic radiation
1.6. Electrostatic discharge.
2. Output of signals (data; commands) ; for example:
2.1. Byzantine behavior.
2.2. Behaving like a “babbling idiot” in a connected network.
3. Other unwanted, unplanned effluents, ; for example, those leading to
3.1. Toxicity
3.2. Inflammability
4. Excessive[120], load or demand on resources; for example:
4.1. Electric power overload, due to a short circuit
4.2. Communication bus overload
4.3. Locking up resources, to the exclusion of other “users” of those resources.

Note: Item 2 and 4.2 are hazards contributed through “logical” causes rather than physical causes.

References for Appendix E

[1] NASA Reference Publication 1358, “System Engineering “Toolbox” for Design Oriented Engineers,” 1994.

[2] Ericson II., C.A., “Hazard Analysis Techniques For System Safety,” John Wiley and Sons, August 24, 2005.

[3] International SEMATECH, “Hazard Analysis Guide: A Reference Manual for Analyzing Safety Hazards on Semiconductor Manufacturing Equipment,” Technology Transfer # 99113846A-ENG, *November 30, 1999.*

[120] i.e.: Disruptive; not accounted for in a validated design

Appendix G: Example case studies

These cases studies illustrate how much can be learned from a single event to prevent or avoid a broader range of mishaps. When a specific mishap is examined for its causes (contributory hazards), pre-existing knowledge of cause-effect relationships can be used as the basis for generalizing from the specific contributory occurrences to more general contributory hazards.

The concept of generalization has been used in a systems engineering process, where a set of scenarios are used (in addition to general requirements) to imply and represent many similar situations, conditions, and cases; these scenarios drive the engineering of the system. The resulting system not only satisfies the requirements explicit in the scenarios, but also many other implied scenarios.

Experts [1] in such generalization have identified two types of reasoning processes, abduction and induction.

G.1 Ft Calhoun Event

Following is an excerpt from the “Ft Calhoun Oversight Increase Dec 13 announcement” [2] and the Fort Calhoun Station Inspection Report [3].

The plant was shut down on April 9 for a refueling outage. The outage was extended due to flooding along the Missouri River. Then an electrical fire on June 7 led to the declaration of an “Alert” and caused further restart complications.

The fire had resulted in the loss of spent fuel pool cooling capability for a brief time and caused significant unexpected system interactions.

The Alert caused by the (electrical circuit) breaker fire resulted from inadequate design or installation of electrical components. Deficiencies were noted with environmental qualification analyses for plant structures, systems and components. These analyses are relied on to demonstrate that key systems will be able to perform their safety functions under a variety of challenging accident conditions like earthquakes, loss of coolant accidents, high radiation fields, seismic events, etc.

Figure illustrates the causality relationships extracted from the textual information above.

Figure illustrates a generalization from the specific occurrence in Ft Calhoun. In this example, the deficiency in the component design was not caught in the V&V activities. However, if we survey known causes of “deficient designs”, the leading cause is “deficient requirements.” Experience in software-reliant systems for many application domains has consistently shown this to be the leading cause. In the context of RIL-1101, “deficient requirements” implies inadequate HA (e.g., inadequate understanding of contributory hazards; inadequate formulation of requirements to avoid or prevent such contributory hazards, and inadequate validation of the HA and the resulting requirements).

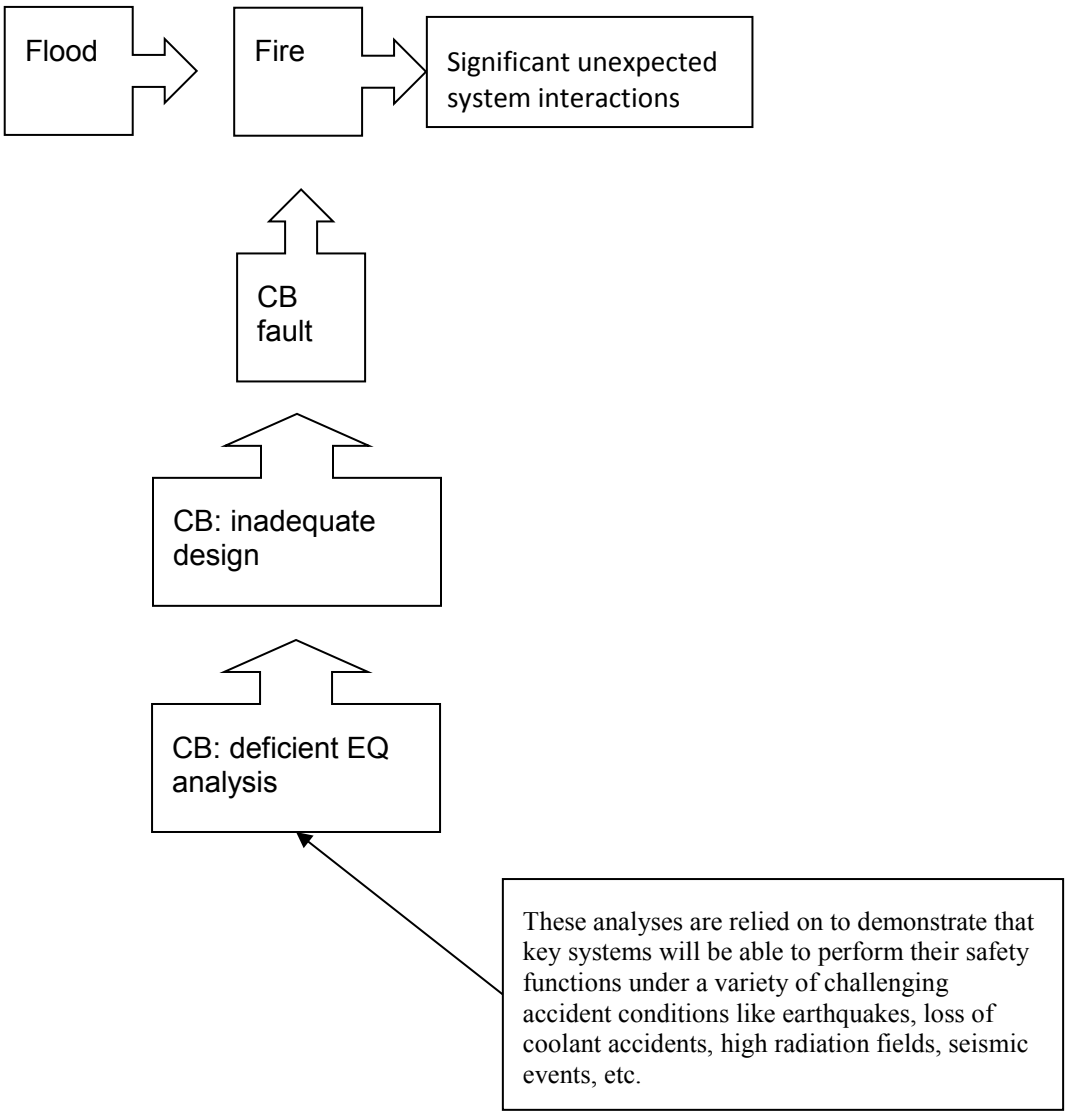

Figure G-1: Example from event on June 7, 2011 at Ft Calhoun NPP

"The power of generalizing ideas, of drawing comprehensive conclusions from individual observations, is the only acquirement, for an immortal being, that really deserves the name of knowledge. "Mary Wollstonecraft (1759–1797), British feminist. A Vindication of the Rights of Woman, ch. 4 (1792)." [4]

References for Appendix G

[1] Abduction and Induction – Essays on their relation and integration, Kluwer Academic Publishers, ISBN 0-7923-6250-0, editors Peter A. Flach and Antonis C. Kakas, 2000

[2] U.S. Nuclear Regulatory Commission, "Inadequate Flooding Protection Due to Ineffective Oversight," Licensee Event Report 285-2011-003, May 1, 2011.

[3] U.S. Nuclear Regulatory Commission, "Fort Calhoun Station – NRC Follow-up Inspection – Inspection Report 05000285/201007; Preliminary Substantial Finding," NRC Inspection Report 05000285/20010007, July 15, 2010.

[4] Dictionary.com, "The_power_of_generalizing_ideas_of_drawing_comprehensive," in *Columbia World of Quotations*. Source location: Columbia University Press, 1996. <http://quotes.dictionary.com/The_power_of_generalizing_ideas_of_drawing_comprehensive>. Available: http://dictionary.reference.com. Accessed: April 27, 2012.

Appendix H: Example checklist of NPP modes

1. On Power
   1.1. Full allowable power
   1.2. Reduced power (including zero power)
   1.3. Raising power or starting up
   1.4. Reducing power
2. Hot Shutdown (reactor sub-critical)
   2.1. Hot standby (coolant at normal operating temperature)
   2.2. Hot shutdown (coolant below normal operating temperature)
3. Cold Shutdown (reactor subcritical and coolant temperature < 93 °C)
   3.1. Cold shutdown with closed reactor vessel
   3.2. Refueling or open vessel (for maintenance)
      3.2.1. Refueling or open vessel – all or some fuel inside the core
      3.2.2. Refueling or open vessel – all fuel outside the core
   3.3. Mid-loop operation (PWR)
4. Construction
5. Preoperational
6. Startup test
7. Commissioning
8. Testing or maintenance being performed
   8.1. Setpoint adjustment
   8.2. Instrument calibration
   8.3. Change (switching) of calibration parameters (in [14] CP 2.1.3.2.5)
9. Decommissioning