1

Draft NISTIR 8179

2

Criticality Analysis Process Model

Prioritizing Systems and Components

5

Celia Paulsen
Jon Boyens
Nadya Bartol
Kris Winkler

10

11

12

13

14

15

16

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Criticality Analysis Process Model

Prioritizing Systems and Components

Celia Paulsen
Jon Boyens
Computer Security Division
Information Technology Laboratory

Nadya Bartol
Boston Consulting Group
Bethesda, MD

Kris Winkler
Boston Consulting Group
Denver, CO

July 2017

71 **Reports on Computer Systems Technology**

72 The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST)
73 promotes the U.S. economy and public welfare by providing technical leadership for the Nation's
74 measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept
75 implementations, and technical analyses to advance the development and productive use of information
76 technology. ITL's responsibilities include the development of management, administrative, technical, and
77 physical standards and guidelines for the cost-effective security and privacy of other than national security-
78 related information in federal information systems.

79

80 **Abstract**

81 In the modern world, where complex systems and systems-of-systems are integral to the functioning of
82 society and businesses, it is increasingly important to be able to understand and manage risks that these
83 systems and components may present to the missions that they support. However, in the world of finite
84 resources, it is not possible to apply equal protection to all assets. This publication describes a
85 comprehensive Criticality Analysis Process Model – a structured method of prioritizing programs, systems,
86 and components based on their importance to the goals of an organization and the impact that their
87 inadequate operation or loss may present to those goals. A criticality analysis can help organizations
88 identify and better understand the systems, subsystems, components and subcomponents that are most
89 essential to their operations and the environment in which they operate. That understanding facilitates better
90 decision making related to the management of an organization's information assets, including information
91 security risk management, project management, acquisition, maintenance, and upgrade decisions. The
92 Model is structured to logically follow how organizations design and implement projects and systems, can
93 be used as a component of a holistic and comprehensive risk management approach that considers all risks,
94 and can be used with a variety of risk management standards and guidelines.

95

96 **Keywords**

97 Baseline criticality; criticality; criticality analysis; critical components; critical programs; critical systems;
98 information security; prioritizing components; prioritizing programs; prioritizing systems; prioritization.

99

100 **Supplemental Content**

101 Criticality Analysis Process Model image file is available at:

102 • http://csrc.nist.gov/publications/drafts/nistir-8179/criticality-analysis-process-model-image.pdf and

103 • http://csrc.nist.gov/publications/drafts/nistir-8179/criticality-analysis-process-model-image.svg

104

105

106

Acknowledgments

108 The authors, Jon Boyens, National Institute of Standards and Technology (NIST), Celia Paulsen (NIST),
109 Nadya Bartol (Boston Consulting Group), and Kristina Winkler (Boston Consulting Group) would like to
110 acknowledge and thank a number of individuals who provided valuable insights and helped improve this
111 publication. We would especially like to thank Kelly Dempsey (NIST), Victoria Pilliteri (NIST), Dr. Ron
112 Ross (NIST), Maureen Moore (NIST), Paul Black (NIST), Dr. Carol Woody (SEI CERT), and John
113 Peterson (Redhorse Corporation) for their contribution to the content during the document development and
114 review.

Note to Reviewers

116 This document is meant to help its users prioritize critical programs, systems, and components. It is not
117 meant as a standalone process, rather it is meant to integrate into already existing processes, such as risk
118 management, information security, security engineering, system and software engineering, safety, quality,
119 and other related disciplines. The authors of the document included references to NIST publications that
120 address risk management, information security, and security engineering.

121 NIST is looking for further suggestions for additional references from the fields of system and software
122 engineering, project management, safety, quality, or other related areas and where to include them in the
123 process. The authors also are looking for further suggestions regarding the methods described in this
124 publication, including what additional methods should be included and references which provide guidance
125 on how to best use the methods.

126 The authors would like to especially get the reviewers' comments on the Illustrative Example in Appendix
127 D. The purpose of the Illustrative Example is to show how the proposed Criticality Analysis Process Model
128 could be used by an organization. Please provide feedback whether the Example makes the uses of the
129 Model clearer. Please also provide suggestions for how we can make the example more helpful and useful.

130

131

132
Executive Summary

134 Draft NIST IR 8179 describes a Criticality Analysis Process Model – a structured method of prioritizing
135 programs, systems and components based on their importance to the mission and the risk that their
136 ineffective or unsatisfactory operation or loss may present to the mission. Criticality Analysis Process
137 Model presented in this document adopts and adapts concepts presented in risk management, system
138 engineering, software engineering, security engineering, safety applications, business analysis, systems
139 analysis, acquisition guidance, and cyber supply chain risk management publications. Criticality analysis is
140 especially pertinent in the current technology environment where organizations rely on information and
141 operation technology product and service providers. The product and service providers take advantage of
142 extended supply chains that make managing information security risks more challenging.

143 A criticality analysis can help organizations identify and better understand the systems, subsystems,
144 components and subcomponents that are most essential to their operations and the environment in which
145 they operate. That understanding facilitates better decision making related to the management of an
146 organization's information assets, including information security risk management, project management,
147 acquisition, maintenance, and upgrade decisions.

148 The Criticality Analysis Process Model can be used as a component of a holistic and comprehensive risk
149 management approach that considers all risks, including information security risks. The Model can be used
150 with a variety of risk management standards and guidelines including the International Organization for
151 Standardization/International Electrotechnical Commission (ISO/IEC) 27000 family of standards and suite
152 of National Institute of Standards and Technology (NIST) Special Publications (SP). The Model can also be
153 used with systems and software engineering frameworks.

154 The notion of criticality analysis originates from Failure Mode Effects and Criticality Analysis (FMECA),
155 used in safety applications. The need for criticality analysis within information security emerged as systems
156 have become more complex and supply chains used to create software, hardware, and services have become
157 extended, geographically distributed, and vast. The first mention of criticality analysis in NIST publications
158 is in NIST SP 800-53 Revision 4 (Rev 4), *Security and Privacy Controls for Federal Information Systems*
159 *and Organizations*. Today, NIST publications mention it in several special publications including those
160 addressing risk management, system security engineering, and supply chain risk management.
161
162 The Model uses existing artifacts, processes, and methods to a maximum extent. It references and uses the
163 outputs of risk management, information security, project management, system design, safety, and other
164 processes that an organization is already performing. To reduce potential redundancy and duplication, the
165 Model identifies integration points with these existing processes.

166 The Criticality Analysis Process Model is structured to logically follow how organizations design and
167 implement projects and systems.

168 The Model consists of five main processes:

169 • A. Criticality Analysis Procedure Definition where the organization develops or adopts a set of
170 procedures for performing Criticality Analysis.

171   •   B. Conduct Program-Level Criticality Analysis where the user defines, reviews, and analyzes the
172   program to identify key activities that are vital to reaching the objectives of the program and for
173   reaching the overall goals of the organization.
174   •   C. Conduct System/Subsystem-Level Criticality Analysis that reviews and analyzes the system or
175   subsystem from the point of view of its criticality to the overall organizational goals.
176   •   D. Conduct Component/Subcomponent-Level Criticality Analysis that reviews and analyzes
177   component or subcomponent from the point of view of its criticality to a specific system or
178   subsystem of which these components and subcomponents are a part.
179   •   E. Conduct Detailed Review of Criticality for Processes B, C, and D that is used to create final
180   criticality levels for Systems/Subsystems and Components/Subcomponents.
181

182 The Model can help increase robustness and granularity of the decisions made about levels of protection
183 afforded to systems and components during system development and acquisition lifecycles. It also provides
184 a means for communicating and coordinating priorities with product and service providers.

185 Using this Criticality Analysis Process Model can help organizations better understand the systems,
186 subsystems, components and subcomponents that are most essential to their operations. Having this
187 information will facilitate holistic information security risk management and integration of security
188 considerations into project management and acquisition.

189

190
191   **Table of Contents**

256

1 Introduction

258 The Criticality Analysis Process Model (hereafter referred to as "the Model") presented in this document
259 adopts and adapts concepts presented in risk management, system engineering, software engineering,
260 security engineering, safety applications, business analysis, systems analysis, acquisition guidance, and
261 cyber supply chain risk management publications. Criticality Analysis is especially pertinent in the current
262 technology environment where organizations rely on third-party product and service providers for the
263 development, integration, and management of the information and operational technology they use.[1] A
264 criticality analysis can help organizations identify and better understand the systems, subsystems,
265 components and subcomponents that are most essential to their operations and the environment in which
266 they operate. That understanding facilitates better decision making related to the management of an
267 organization's information assets, including information security risk management, project management,
268 acquisition, maintenance, and upgrade decisions.

1.1 Purpose and Scope

270 In the modern world, where complex systems and systems-of-systems are integral to the functioning of
271 society and businesses, it is increasingly important to be able to understand and manage risks that these
272 systems and components may present to the missions that they support. And in the world of finite resources,
273 it is not possible to apply equal protection to all assets. Managing risk can be improved with processes,
274 methods, and techniques to prioritize assets for a detailed risk analysis and for applying information security
275 controls. However, existing standards and guidelines provide only high level and fragmented guidance for
276 how to prioritize systems and components in relation to the goals of the organization, the mission, and the
277 environment. Additionally, these existing standards and guidelines are most often focused on prioritizing
278 projects according to organizational goals, or prioritizing components according to system functionality; this
279 can result in an incorrect assumption about the critical nature of a component to organizational goals.

280 The document describes a comprehensive criticality analysis process model – a structured method of
281 prioritizing programs, systems, and components based on their importance to the goals of an organization
282 and the impact that their inadequate operation or loss may present.to those goals. The Criticality Analysis
283 Process Model is intended to be used as a component of a holistic and comprehensive risk management
284 approach that considers all risks, including information security risks. The Model can be used with a variety
285 of risk management standards and guidelines including the International Organization for
286 Standardization/International Electrotechnical Commission (ISO/IEC) 27000 family of standards and suite
287 of National Institute of Standards and Technology (NIST) Special Publications (SP). It can also be used in
288 conjunction with systems and software engineering and project management frameworks. The Model can
289 help increase robustness and granularity of the decisions made about levels of protection afforded to
290 systems and components during system development and acquisition lifecycles. It also provides a means for
291 communicating and coordinating priorities with product and service providers.

---

[1] Operational technology: programmable systems or devices that interact with the physical environment (or manage devices that interact with the physical environment). These systems/devices detect or cause a direct change through the monitoring and/or control of devices, processes, and events. Examples include industrial control systems, building management systems, fire control systems, and physical access control mechanisms.

292   The Model uses existing artifacts, processes, and methods to a maximum extent. It references and uses the
293   outputs of risk management, information security, project management, system design, safety, and other
294   processes that an organization is already performing. The Model is not intended to replace any of these
295   processes, but to reduce potential redundancy and leverage existing efforts; the Model identifies integration
296   points with these existing processes. An organization may have additional processes not listed in this
297   publication, which may also be performed alongside and integrated with the Model.

1.2   Background

299   The notion of criticality analysis originates from Failure Mode Effects and Criticality Analysis (FMECA),
300   used in safety applications. The need for a criticality analysis within information security emerged as systems
301   have become more complex and supply chains used to create software, hardware, and services have become
302   extended, geographically distributed, and vast. The first mention of criticality analysis in NIST publications
303   is in NIST SP 800-53 Revision 4 (Rev 4), *Security and Privacy Controls for Federal Information Systems*
304   *and Organizations*. Today, it is mentioned in several NIST special publications including those addressing
305   risk management, system security engineering, and supply chain risk management.
306
307   Together, these documents provide high-level guidance on criticality analysis, including how to integrate it
308   into the broader risk management, system engineering, or security engineering activities. However, these
309   publications do not provide detailed guidance for how to perform the criticality analysis itself. A number of
310   U.S. Government agencies have implemented criticality analysis processes, but these processes are
311   nonstandard and are often not formally defined. Meanwhile, the need for detailed information for how to
312   identify what is critical and how to prioritize its protection within a system has become more acute due to
313   how modern systems and components are designed, developed, manufactured, acquired, and deployed.
314   Identifying the asset of greatest importance is not a new concept. A number of disciplines have well-
315   established methods for doing so, including business risk management, project management, safety, supply
316   chain management, critical infrastructure protection, and others. These concepts are used in a variety of
317   industries including banking and electric utilities. The authors researched and compared these existing
318   methods and approaches to develop the Model described in this publication; it is anchored in these existing
319   methods and approaches and is tailored specifically to the needs of information security risk management.

1.3   Audience

321   The audience for this publication is a broad set of federal agency leaders and practitioners including those
322   engaged in cybersecurity/information security; information technology; contracting;
323   procurement/acquisitions; system and software development/engineering; security engineering; program
324   management; and system owners. Other personnel or entities are free to make use of the guidance as
325   appropriate to their situation.

1.4   Relationship to other standards and NIST publications

327   The Criticality Analysis Process Model can be integrated into a variety of processes including information
328   security, risk management, system and software engineering, acquisition, and project management. It can
329   also be used in conjunction with safety and business analysis processes. This publication builds on a set of
330   multi-disciplinary publications, standards, and guidelines, developed by NIST, International Organization
331   for Standardization (ISO), and other bodies. Criticality Analysis is mentioned and can be used with the
332   following NIST SPs:

333 • NIST SP 800-53 Revision 4 (Rev 4), *Security and Privacy Controls for Federal Information Systems*
334 *and Organizations,* which describes security control SA-14: *Criticality Analysis*.
335 • NIST SP 800-160 *Systems Security Engineering: Considerations for a Multidisciplinary Approach*
336 *in the Engineering of Trustworthy Secure Systems,* mentions Criticality Analysis a part of
337 performing control SA-2: *Perform the security aspects of systems analyses*, as well as in Appendix
338 G: *Engineering and Security Fundamentals*.
339 • NIST SP 800-161, *Supply Chain Risk Management Practices for Federal Information Systems and*
340 *Organizations,* mentions the concept of criticality several times:
341   o "Baseline Criticality" is mentioned in Task 1-1 to be determined as a part of the Frame step
342   of the Risk Management Process
343   o In Task 2-0, *Criticality Analysis,* a task is to be performed at the beginning of the Assess step
344   in the Risk Management Process.
345   o In the supplemental guidance for control SA-14: *Criticality Analysis*.

346 Through these publications, criticality analysis is integrated into the broader set of related NIST
347 publications, including:

348 • NIST SP 800-39, *Managing Information Security Risk: Organization, Mission, and Information*
349 *System View*
350 • NIST SP 800-30 Rev 1, *Guide for Conducting Risk Assessments*.

351 The Criticality Analysis Process Model presented in this document can be used in conjunction with ISO
352 standards focused on risk management or information security in supplier relationships:

353 • ISO/IEC 27036 – *Information technology – Security techniques – Information security for supplier*
354 *relationships*
355 • ISO/IEC 27001 – *Information technology – Security techniques – Information security management*
356 *system*
357 • ISO/IEC 27002 – *Information technology – Security techniques – Code of practice for information*
358 *security management.*

359 The Criticality Analysis Process Model can also be used in conjunction with additional standards and
360 publications focused on system and software engineering:

361 • NIST SP 800-160, *Systems Security Engineering*
362 • ISO/IEC/IEEE 15288 – *System and software lifecycle processes*
363 • *International Council on Systems Engineering (INCOSE) System Engineering Handbook.*

364

365

366   **1.5   Structure of this document**

367   The rest of this publication is organized as follows:
368   • Chapter 2 provides an overview of the Model including the methodology used to develop it and tips
369   for how to read the Model diagram itself;
370   • Chapter 3 provides a deep-dive description of the Model and the processes that comprise the Model
371   • Appendix A lists acronyms and abbreviations used in this document
372   • Appendix B provides the bibliography of sources and references used in this document
373   • Appendix C provides a brief overview of methods mentioned in the Model
374   • Appendix D provides an illustrative example of how the Model can be used
375   • Appendix E provides a detailed diagram of the Model

2 Criticality Analysis Process Model Overview

This chapter provides an overview of the Criticality Analysis Process Model including:

- The methodology that was used to develop the Model
- Overview of the top-level processes in the Model
- Guidance on how to read the process diagrams used to depict the Model, including any "rules" for interpreting those diagrams.

2.1 Methodology

The Model was developed by conducting four main activities:

- Environmental scan that included identification and detailed review of publications from different subject areas that describe methods for identifying critical assets.
- Comparative analysis and synthesis of the reviewed methodologies to derive a common set of steps for a criticality analysis.
- Identification of steps relevant to information security and potential steps not described in existing literature.
- Translation and transformation of the steps into the information security practitioner language.

First, the authors identified and collected a number of methodologies that described a process for identifying or prioritizing critical assets. A subset of methodologies were then selected for further research, based on an initial assessment of their potential applicability to an information security criticality analysis, their comprehensiveness, uniqueness, and usability by the intended audience. The authors also contacted a small number of subject matter experts to provide insights into the various methodologies and their usefulness to an information security criticality analysis.

The authors then summarized and conducted a detailed review of each methodology using a structured matrix format. The summaries included but were not limited to the following information:

- Applicability to an information security criticality analysis for projects, systems, and components
- Scalability to large and small projects/organizations;
- Investment/cost considerations;
- Applicability to an existing or new system;
- Inputs and outputs;
- Complexity and difficulty of use; and
- Demonstrated effectiveness in their respective domains.

The authors then conducted a comparative analysis and synthesis of the methodologies using another matrix to derive a common set of steps. In addition, additional steps were described that filled gaps related to the application of existing literature to an information security domain. The authors then validated the Model against relevant information security sources to ensure that the terminology in and the general flow of the Model was consistent with information security concepts and guidance. Next, the Model itself was constructed using commonly accepted process modeling techniques. The finalized set of steps were

415   translated and transformed into information security practitioner language and aligned with existing NIST
416   publications. The Model was then edited and simplified for ease of use.

417   **2.2   Model Overview**

418   The Criticality Analysis Process Model is structured to logically follow how organizations design, acquire,
419   and implement projects and systems. Traditionally, organizations establish projects and programs to
420   accomplish mission and business objectives and to guide the performance of corresponding activities. They
421   design and/or deploy information systems to support those activities. These systems are often a loosely
422   defined, complex mixture of hardware, software, network infrastructure, data, humans, and other elements,
423   and may be composed of numerous subsystems (this architecture is often called "systems of systems"). The
424   IT/OT components and subcomponents used to construct these systems and subsystems typically come from
425   a variety of sources and are often Commercial-off-the-Shelf (COTS) products. Different organizational units
426   – including third parties – naturally have different roles and responsibilities with respect to these projects,
427   systems, and components.

428   The structure of the Model defined in this document accommodates these dynamics and at the same time
429   helps to facilitate a holistic view of criticality for a collection of programs, systems/subsystems, and
430   components/subcomponents.[2] The Model combines top-down and bottom-up analysis approaches. The
431   top-down approach in this model enables the organization to identify critical processes and then to
432   progressively narrow the analysis to critical systems that support those processes, and then to critical
433   components which ensure the critical functions of those systems. It follows an ideal system development
434   process while allowing the flexibility for analyzing systems and components in a less ideal situation. The
435   bottom-up approach progressively traces the impact of a malfunctioning or compromised critical component
436   would have on the system, and then on the program. It allows for the identification of connections and
437   dependencies between components, systems, and programs that are not easily identified in a top-down
438   approach. The combination of using top-down and bottom-up approaches ensures that the Model is
439   thorough and complete.

440   The Model consists of five main processes as depicted in Figure 1:

441   • A. Criticality Analysis Procedure Definition
442   • B. Conduct Program-Level Criticality Analysis
443   • C. Conduct System/Subsystem-Level Criticality Analysis
444   • D. Conduct Component/Subcomponent-Level Criticality Analysis
445   • E. Conduct Detailed Review of Criticality for Processes B, C, and D.

446

---

[2] The model does not require organizations to use standard or identical definitions of program, system, subsystem, component, or subcomponent
in order to allow organizations the flexibility of using their existing definitions; however, the model was developed with the assumption
that the systems and components evaluated would be technological in nature (IT/OT). This is explained further in the Model itself.
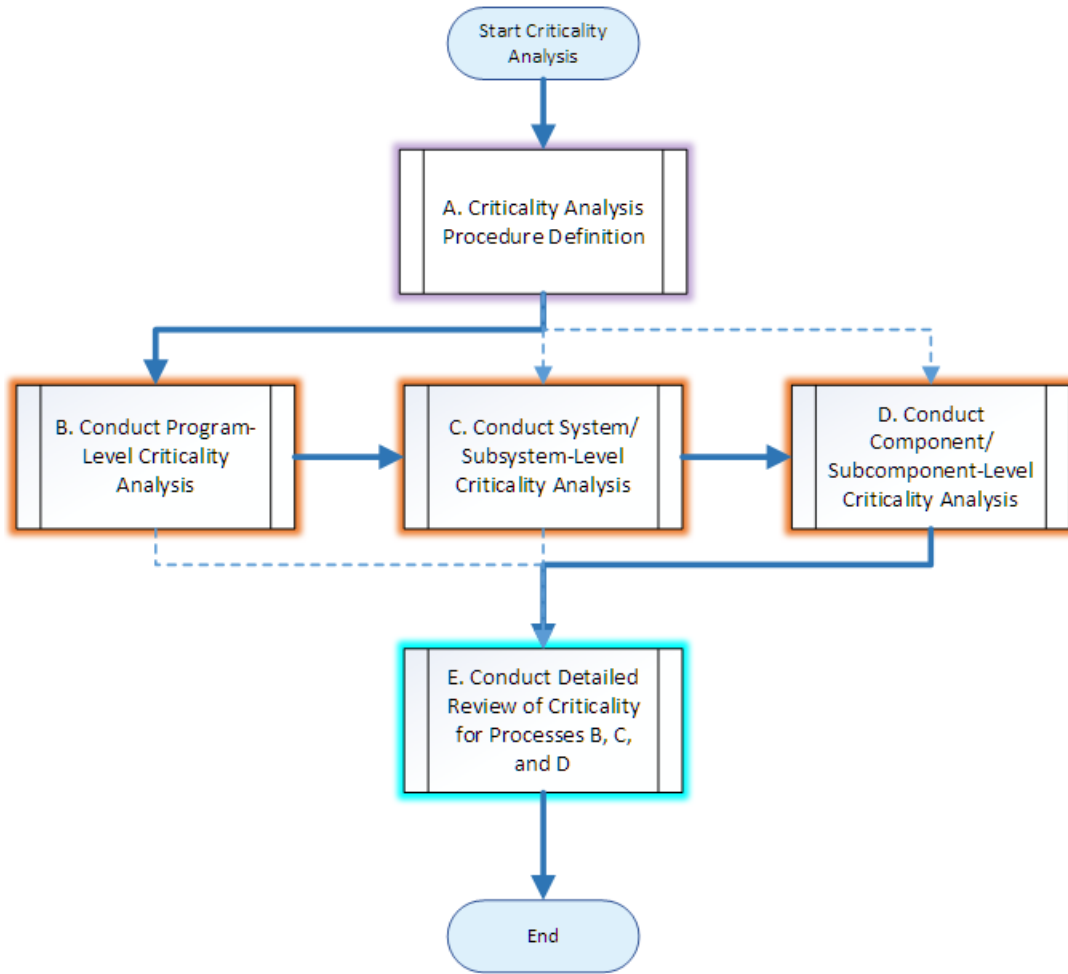
Figure 1 - High Level Criticality Analysis Process Model

Process A is expected to be completed before other processes. Processes B, C, and D ideally will be performed in sequence to provide a comprehensive top-down analysis, but may be performed at the same time or out of sequence (this is shown in the model with dotted lines). Different individuals typically perform these processes; process D in particular is likely to be done partially by a third party in the case of COTS products. These three processes have iterative sub-processes and can be conducted at increasing levels of detail to refine the results and accept additional inputs. Process E is a bottom-up analysis using inputs from, and cutting across, processes B, C, and D. It is performed at the very end to finalize criticality levels for programs, systems/subsystems, and components/subcomponents.

Section 3, Model Process and Sub-process Descriptions, describes each process and sub-process in detail.

459   **2.3   How to Read the Model**

460   Criticality Analysis Process Model was developed using formal process modeling techniques. The
461   following symbols were used in the Model:

462   Documents within/external to the enterprise

463   Output of a sub-process

464   Start/end of process

465   Decision point

466   Sub-process

467   Previous process/sub-process

468   Iterative Feedback loop

469   The Model should be read top-down and left-to-right. Boxes around a series of sub-processes indicate that
470   these sub-processes are iterative and can be performed in a loop until an acceptable output is produced.

471   Each process has "start" and "end" and accepts inputs both from the Model and from other sources. Each
472   process produces outputs that serve as inputs into subsequent processes within the Model.

3   Model Process and Sub-process Descriptions

This chapter provides detailed descriptions of the Model's processes and sub-processes. The five processes
of the Criticality Analysis Process Model are depicted in Figure 1 in Section 2, and a more detailed version
of the Model may be found in Appendix E. Each process consists of one or more sub-processes. Detailed
diagrams of each process, including the associated sub-processes, are included in the sections below.

The first process, *A. Criticality Analysis Procedure Definition,* provides guidance, structure, and continuity
for performing a criticality analysis, necessary due to the number of different people and groups involved in
completing a criticality analysis.

The next three processes, *B. Conduct Program-Level Criticality Analysis, C. Conduct System/Subsystem-
Level Criticality Analysis, and D. Conduct Component/Subcomponent-Level Criticality Analysis*, act as a
top-down means of mapping and prioritizing activities, associated systems/subsystems, and finally,
components/sub-components of those systems. These three processes are very similar to each other
conceptually, but require different methods for completion and are typically done by separate groups of
people with differing areas of expertise. They are iterative and can be conducted at increasing level of detail
to refine the results and accept additional inputs. Ideally, these three processes should be conducted in
sequence. However, it is likely that for many use cases, they will be conducted at least partially out of
sequence or in parallel to each other.

The last process, *E. Conduct Detailed Review of Criticality for Processes B, C, and D*, is performed after
Processes *B*, *C*, and *D* have been completed and cuts across these three processes. This process is performed
in a bottom-up manner for tracing dependencies and impact/risk from sub-components to components,
components to subsystems, subsystems to systems, systems to programs, and programs to higher-level
programs using the information gathered in the previous three processes. It provides connective tissue
between Processes *B*, *C*, and *D*, and ensures that the criticality determination is consistent across all layers
of the Model – program, system/subsystem, and component/subcomponent – in terms of considering
impacts, dependencies, and risks across the entire program. As such, Process E requires a high level of
coordination and collaboration between the actors in those other processes. Baseline Criticality levels
assigned in Processes *C and D* are finalized in process E; the Baseline Criticality levels determined in
Process B are typically sufficient for the program level and so do not need to be finalized in Process E.

Note: Organizations do not need to complete each process or sub-process exactly as described in
this document in order to complete a criticality analysis. Rather, organizations are expected to
tailor this Model to their own needs, capabilities, and operating environment.

The sections below describe inputs, outputs, and methods. These are listed are for informational purposes
only, as examples, and it is expected that users will fill in these items with information specific to their
organization(s) when they tailor the Model. The inputs listed provide examples of the types of documents
that may be useful in completing the process/sub-process. Organizations may not have all the inputs

510 mentioned in this publication. If a specific input does not exist or is unavailable for any reason, the same
511 type of information may exist as part of another document or in another format. Similarly, outputs described
512 in this publication do not need to be stand-alone documents but may be part of an existing document or in
513 another format than is described herein.

514 The methods listed are intended to provide additional guidance on how to complete the sub-processes.
515 These methods are not described in detail, but are briefly described in Appendix C, with references for
516 further guidance where available. In tailoring the Model, it may be useful to more fully describe or provide
517 information on methods to be used.

518 This chapter describes each process and the associated sub-processes in detail. Each process is described
519 using the following template:

Process	Letter designator of the process
Process name	Name of the process
Process summary	Description of the process
Inputs	Documents that may be useful in completing the process.
Outputs	Documents that are created or modified as a result of completing the process. There are two types of outputs: 1. Informal outputs that capture information passed from a sub-process to the next sub-process. Those outputs may include working documents or emails and are not depicted as outputs in the Model. 2. Outputs that produce formal documentation, although the nature of that documentation is flexible. Each process produces at least one piece of formal documentation that is depicted in the Model.
Roles and Responsibilities	List of roles regarding who will be Responsible, Accountable, Consulted, or Informed of the sub-process and its outputs.
Related processes outside of Criticality Analysis	A common security, engineering, business, etc. process that is related to but not directly a part of a criticality analysis. Knowing how these related processes fit into the Criticality Analysis may help identify areas where existing work may be leveraged and provide some context in understanding the Criticality Analysis process.

520

521   Each sub-process will be summarized using the following template:

Sub-process number	Number designator of the sub-process
Sub-process name	Name of the sub-process
Sub-process summary	Description of the sub-process
Inputs	Documents that may be useful in completing the process.
Outputs	Documents that serve as outputs from the process. There are two types of outputs: 1. Informal outputs that capture information passed from a sub-process to the next sub-process. Those outputs may include working documents or emails and are not depicted as outputs in the Model. 2. Outputs that produce formal documentation, although the nature of that documentation is flexible. These outputs are depicted in the Model.
Methods	Methods that may be used in the performance of the sub-process.
Related processes outside of Criticality Analysis	A list of common security, engineering, business, or other processes that are related to or tie in with this sub-process, its inputs, or its outputs. Knowing how these related processes fit into the Criticality Analysis may help identify areas where existing work may be leveraged and provide some context in understanding the Criticality Analysis process.

522

523

524

525 **Process A – Start Criticality Analysis Procedure Definition**
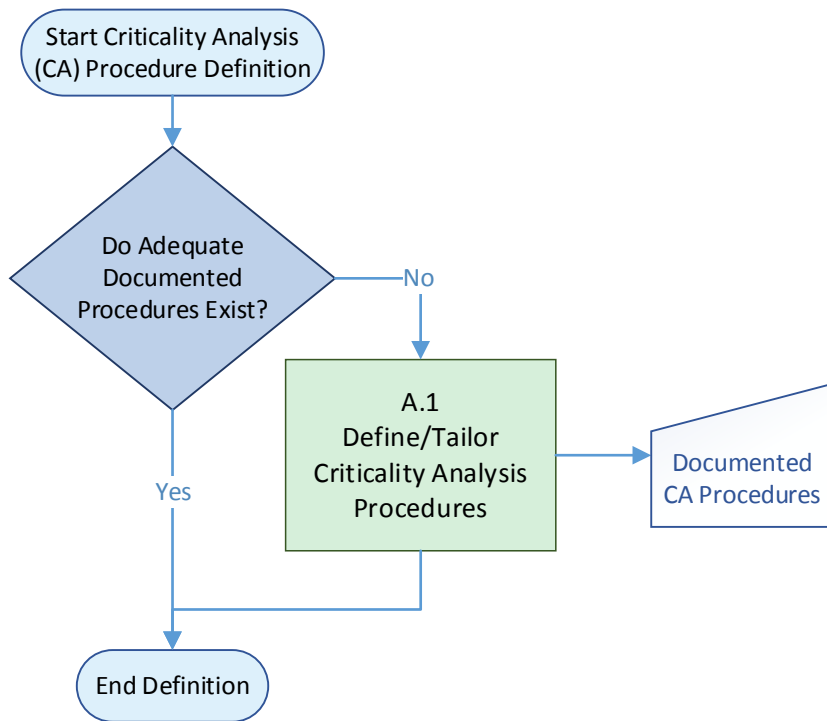
527 **Figure 2 - Start Criticality Analysis Procedure Definition**

528 Process A, *Criticality Analysis Procedure Definition,* depicted in Figure 2, should be completed before the
529 rest of the criticality analysis is performed. This process ensures that there is a set of documented
530 procedures to guide the Criticality Analysis. It helps set up for a successful execution of the Criticality
531 Analysis by providing scoping, framing, and procedural guidance for conducting a Criticality Analysis.

532 Process A consists of the following:

533 • Check if documented procedures already exist and if they are sufficient and appropriate for the
534 needs of the Criticality Analysis;
535 • If procedures already exist, then Process A can end and Process B can begin;
536 • If procedures do not exist or are not sufficient or appropriate for the needs of the Criticality
537 Analysis, sub-process A.1, *Define Criticality Analysis Procedures,* should be performed to develop
538 or tailor Criticality Analysis Procedures.
539 • Once the procedures have been satisfactorily defined or tailored in sub-process A.1, Process A can
540 end and Process B, can begin.

541 The output of this process is "Documented Criticality Analysis Procedures". There does not need to be a
542 document named "Criticality Analysis Procedure", but there needs to be a document that provides guidance
543 on how to conduct a criticality analysis. A project plan, program plan, program implementation plan, or
544 other kind of plan may provide sufficient guidance.

545

Process number	A
Process name	Criticality Analysis Procedure Definition
Process summary	The organization either develops procedures that would guide the Criticality Analysis, or, if such procedures exist, finds them and, if needed, tailors them to the specific needs of the program.
Inputs	None
Outputs	Documented Criticality Analysis Procedures
Responsible persons	Responsible: Project Manager in charge of the Criticality Analysis

Accountable: Program Manager can delegate the execution of this process to another suitable individual, e.g., business analyst.

Consulted: Individuals who have an understanding of the organizations' operational environment, individuals with project management, process management, or criticality analysis experience. Individuals who developed the criticality analysis model being tailored.

Informed: Individuals responsible for conducting any part of the Criticality Analysis. |
| Related processes outside of Criticality Analysis | NIST SP 800-39 – Frame

NIST SP 800-161 – Frame

NIST SP 800-160, Project Planning Process (3.3.1) |

546

547 **A.1 – Define/Tailor Criticality Analysis Procedures**

Sub-process number	A.1
Sub-process name	Define/Tailor Criticality Analysis Procedures
Sub-process summary	Develop procedures for conducting a criticality analysis by adapting this Model to the organization's structure and environment. If a criticality analysis procedure has already been adapted by an organization, tailor or refine that process to the needs and environment of the specific Criticality Analysis being conducted. This includes defining the scope and other requirements for the Criticality Analysis, identifying responsible parties, and detailing the procedures to be used in conducting the analysis.

Include in the procedure, roles and responsibilities, including the individuals |

13

	responsible for, accountable for, consulted on, or informed about each process and sub-process in this Model. Define how these individuals will communicate. This is especially important in cases where different organizational units or third parties will be conducting portions of the analysis. Consider any relevant contract requirements.
Inputs	None
Outputs	Documented Criticality Analysis Procedures
Methods	Project Planning; Document Review
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame

NIST SP 800-161 – Frame

NIST SP 800-160, Project Planning Process (3.3.1) |

548

549 **3.1 Process B – Conduct Program-Level Criticality Analysis**
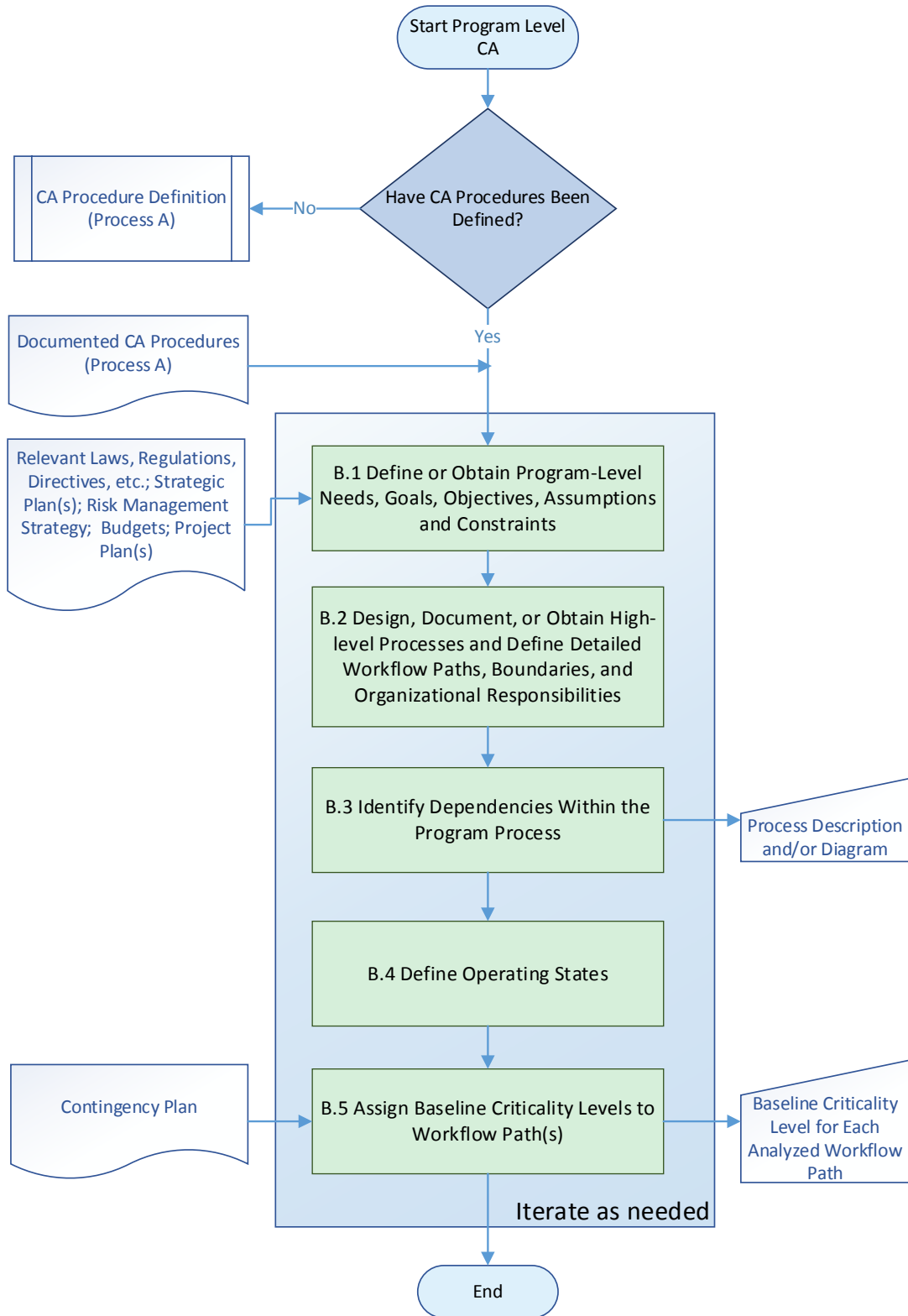
550

551 **Figure 3 - Conduct Program Level Criticality Analysis**

552

Process B, *Conduct Program-Level Criticality Analysis*, depicted in Figure 3, is the first layer of the top-down portion of the Criticality Analysis. For the purpose of this document, a program does not necessarily mean an official government program; it may be a collection of programs, an initiative, or an idea. It is defined by a set of objectives and encompasses the activities that the organization performs in order to accomplish those objectives. The program may be formally defined in a mission statement, project plan or other similar document, it may be a concept under development, or it may be a situation that the organization routinely faces but does not fall under any single program's responsibilities. Ideally, Process B would start at the highest level of an organization and repeat iteratively with increasing granularity until the lowest hierarchy of programs is reached. In most cases, this is impractical, and so this model allows users to begin with any program for which the information required to perform a criticality analysis is available.

Process B helps the user define the program and identify key activities that are necessary to ensure the main goals and objectives of the program are met. It consists of the following:

- Obtain or define program goals and objectives, assumptions, and constraints;
- Obtain, design, or document a high-level process for completing the objectives of the program;
- Identify dependencies within the program process;
- Define how the program will operate normally and if it is impacted by an adverse event, that is referred to as an adverse operating state; and
- Assign Baseline Criticality levels to workflow paths based on gathered information.

Much of this process may already be conducted as part of strategic planning or project planning efforts.

572

Process	B
Process name	Conduct Program-Level Criticality Analysis
Process summary	Define, review, and analyze the program to identify key activities that are vital to reaching the objectives of the program and for reaching the overall goals of the organization. This process ensures that that the criticality determinations for systems/subsystems and components/subcomponents can be directly traced back to the objectives of the program and the goals of the organization.
Inputs	Documented Criticality Analysis Procedures (from Process A); Relevant Laws, Regulations, Directives, etc.; Strategic Plan(s); Risk Management Strategy; Budget; Project Plan(s); Contingency Plan
Outputs	Process Description and/or Diagram; Baseline Criticality Levels of Activity(ies) and/or Workflow Path(s)
Roles and	Responsible: Program Manager should be responsible for the performance of this process. Lead Security Engineer should serve as a co-lead for sub-

Responsibilities	processes B.4, Define Operating States, and B5, Assign Baseline Criticality Levels to Workflow Path(s).
	Accountable: Program Manager can delegate the execution of this process to another suitable individual, e.g. business analyst.
	Consulted: Individuals who have detailed knowledge of the activities identified by this process should participate in this process to contribute to the identification of such activities. These individuals may include system architects and designers, system engineers, security engineers, other security professionals, acquisition/procurement professionals, business leaders, and others, as appropriate. Representatives of each relevant group should be invited to participate in this process.
	Informed: Individuals responsible for conducting any part of the Criticality Analysis.
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame
	NIST SP 800-161 – Frame
	NIST SP 800-160 – Business or Mission Analysis Process (3.4.1)
	NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

573

574   **B.1 – Define or Obtain Program Level Needs, Goals, Objectives, Assumptions, and Constraints**
575   **Sub-Process**

Sub-process number	B.1
Sub-process name	Define or Obtain Program Level Needs, Goals, Objectives, Assumptions and Constraints
Sub-process summary	This process helps define the program being analyzed. It lays the foundation for the criticality analysis, establishes context, and provides a common perspective on the assumptions, constraints, risk tolerances, and priorities/trade-offs used for making investment and operational decisions.
	Define how the success or failure of the program will be measured. Identify a high-level objective or set of related objectives. It is best if there is only one objective or if the set of objectives are closely related in order to focus the analysis. Measurable goals for the objective(s) should be described and include any high-level organizational goals that apply. Consider security, safety, privacy, and other related goals.

	Requirements or constraints should be clearly identified and may include applicable legal regulations, organizational policy, risk tolerance, budgets, and any other constraints that may impact the flexibility of the organization's activities. Finally, any assumptions that may impact the analysis should be defined. These may include environmental, legal, budgetary, or other variables that may have some degree of uncertainty. An example of an assumption could be "assuming no change in budget", or "assuming an operating environment consistent with typical North American weather". Goals, objectives, assumptions, and constraints may be available from current documentation or will need to be developed.
Inputs	Documented Criticality Analysis Procedures (or project plan) from Process A; Relevant Laws, Regulations, Directives (including organizational policies), and other high-level guiding documents that may have the right information; Strategic Plan(s); any documentation that describes organizational mission/vision; needs, goals, objectives, and projects; Risk Management Strategy (including risk tolerance); Budgets and Project Plan(s)
Outputs	Documentation of goals, objectives, assumptions, and constraints
Methods	Project Plan; Document Review; Brainstorming; Process Flow Diagram; Responsible/Accountable/Consulted/Informed (RACI) Charts.
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Business Environment (ID.BE-3, 4), Governance (ID.GV-1, 2, 3) NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

576

577 **B.2 – Design, Document, or Obtain High-level Processes and Define Detailed Workflow Paths,**
578 **Boundaries, and Organizational Responsibilities**

Sub-process number	B.2
Sub-process name	Design, Document or Obtain High-level Processes and Define Detailed Workflow Paths, Boundaries, and Organizational Responsibilities
Sub-process summary	This sub-process defines how the organization accomplishes the objectives defined in B.1. Loosely describe the main activities that will be conducted to reach the objectives and goals defined in B.1. Include any activities that will be conducted in a regular course of events, to measure the performance of the program, and activities that will be conducted in case of an adverse event (i.e. contingency plans). If possible, identify at a high level all activities regularly conducted by the organization(s) responsible for the completion of the program including those responsible for performing these activities. This may include things such as inspections, maintenance, payroll processing, or any other activity that may have an impact on the successful completion of the project. Create a map, diagram, or other representation of all the activities identified. Ensure this representation describes connections between the activities (e.g. a document created by one activity is used in the completion of another), including what activities must be completed before another can begin. The representation should describe a workflow with an identifiable beginning and end. In many cases, the last few activities of the workflow will consist of measuring and reporting how well the objectives of the program were met. Consider including in the representation any outputs or products that will be created and transferred between activities. If the high-level program workflow path is complex or the activities within the workflow are complex, separate the workflow into different processes or workflow paths (also known as "mission threads"). Each workflow path should have an identifiable beginning and end. The more detail that is put into this representation, the more specific and tailored the overall criticality analysis can be. However, very detailed diagrams are also less flexible to changes and can be time-consuming to create; it is important to define what level of detail is necessary for the Criticality Analysis.[3] Once the activities are identified and represented in a workflow or set of

---

[3] The workflow paths will traverse multiple systems that are supporting the program, which means that there will be handoffs between systems, traversing of system boundaries, transfer between different organizations or individuals, and other events that involve transition.

	workflow paths, boundaries of those activities should be defined and described, and individuals who will be performing those activities identified. Boundaries may be defined in terms of time, triggers, functionalities, systems, organizational units, or any system that makes sense to the user. The boundaries and responsible individuals should be documented in sufficient detail in order to provide enough information to support sub-process B.3 and identify which points in the process may be critical to the organization as analyzed in sub-processes B.4 and B.5, and Process E.
Inputs	Documentation of goals, objectives, assumptions, and constraints from B.1.
Outputs	Description of workflow paths, boundaries, and roles and responsibilities to pass to B.3. (This can be done in a document, spreadsheet, or draft process diagram.)
Methods	Project Plan; Document Review; Brainstorming; Process Visualization; RACI Charts; Interviews; Observation; Sequence Diagrams; Scenario/Use-Case
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Asset Management (ID.AM-3, 6) NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

579

580   ***B.3 – Identify Dependencies Within the Program Process***

Sub-process number	B.3
Sub-process name	Identify Dependencies Within the Program Process
Sub-process summary	This sub-process helps identify connection points within the program that may be stressed in an adverse situation. Examine the workflow paths defined in B.2 and determine where they intersect and/or depend on each other. Highlight or otherwise clearly identify any activity or output that multiple workflows depend upon. Consider the boundaries defined in B.2 and identify situations where one activity is dependent on an output or activity located in a different boundary, is outside of the defined workflow path(s), or is outside the scope of the criticality analysis. Also, consider the individuals/ responsibilities defined in B.2 and identify where multiple activities are conducted by a single individual or organizational unit. If appropriate, create or update the process diagram or other representation depicting the workflow paths to include dependencies.
Inputs	Description of workflow paths, boundaries, and roles and responsibilities from B.2.
Outputs	Process Description and/or Diagram; Listing of intersections and dependencies to pass to B.4.
Methods	Document Review; Process Flow Analysis; Interdependency Analysis; Activity Network Diagram; Gantt Chart; Scenario/Use Case; Mission Thread Analysis
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Business Environment (ID.BE-4) NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

581

582 **B.4 – Define Operating States**

Sub-process number	B.4
Sub-process name	Define Operating States
Sub-process summary	This sub-process defines scenarios for regular operation of the program and what might happen if the activities or workflow paths defined in B.2 are compromised. For each activity defined in B.2 and B.3, describe the condition of the activity (i.e. what the activity could look like) and the impact on the workflow if the activity is forced into different conditions, including but not limited to: • Non-operational (i.e. the activity does not occur) • Impaired (i.e. the activity operates at a reduced pace or in an unsafe/insecure manner) • Normal operation (i.e. how the activity operates under typical or ideal circumstances) • Increased operations (i.e. the activity performs quicker or with more output than normal) • Unintended operations (i.e. the activity performs but with additional outputs or actions that are not part of the expected routine) Determine the severity of the operating states other than normal (adverse operating states) on the workflow. This could be a ranking (e.g. low, moderate, high), or measure (e.g. time lost; cost in time/resources). Consider defining what types of scenarios would lead to such situations. Examples of scenarios to consider include reduced performance (e.g., lower bandwidth), security breach, physical accident, or any other similar event. Consider security, safety, and privacy ramifications. For example, what information is made vulnerable if the activity performs slower than normal? Could there be physical damages if an activity is performed too quickly? Then, using the Process Description and/or Diagram created in B.3, identify specific intersections and dependencies in the workflow paths that would be impacted by each adverse operating state. The question to try to answer here is how do the adverse operating states of the activity impact the other activities in the process and in turn identify which intersections or dependencies have the most influence over whether the program continues operating normally and thus are more critical.
Inputs	Process Description and/or Diagram; listing of intersections and dependencies.

Outputs	Description of Operating States
Methods	Document Review; Brainstorming; Interviews; Group Decision Making
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

583

584   **B.5 – Assign Baseline Criticality Levels to Workflow Path(s)**

Sub-process number	B.5
Sub-process name	Assign Baseline Criticality Levels to Workflow Path(s)
Sub-process summary	This sub-process determines criticality levels of workflow paths defined in B.3 against operating states defined in B5. Using the Process Description and/or Diagram created in B.2 and B.3, consider how the program would be affected by each of the operating states defined in B.4. Rank the activities, workflow paths and/or bounded areas according to how vital they are to the success of the objectives/goals defined in B.1 and how strongly an adverse operating state will affect the program objectives and goals. The user should create a way to measure or rank the workflow paths according to how important they are to the success of the program. This could be a ranking (e.g. low, moderate, high), or measure (e.g. time lost; cost in time/resources; probability of being able to complete activity). The user could also use ranges and thresholds to define such rankings.[4]
Inputs	Process Description and/or Diagram from B.3; Listing of intersections and dependencies from B.3; Listing of operating states from B.4; Contingency Plan
Outputs	Baseline Criticality for each analyzed workflow path.

---

[4] For further guidance on example security measures and metrics see Appendix A of NIST SP 800-55 Revision 1, *Performance Measurement Guide for Information Security.*

Methods	Document Review; Group Decision Making
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2) FIPS 199

585

586 **3.2 Process C– Conduct System/Subsystem-Level Criticality Analysis**
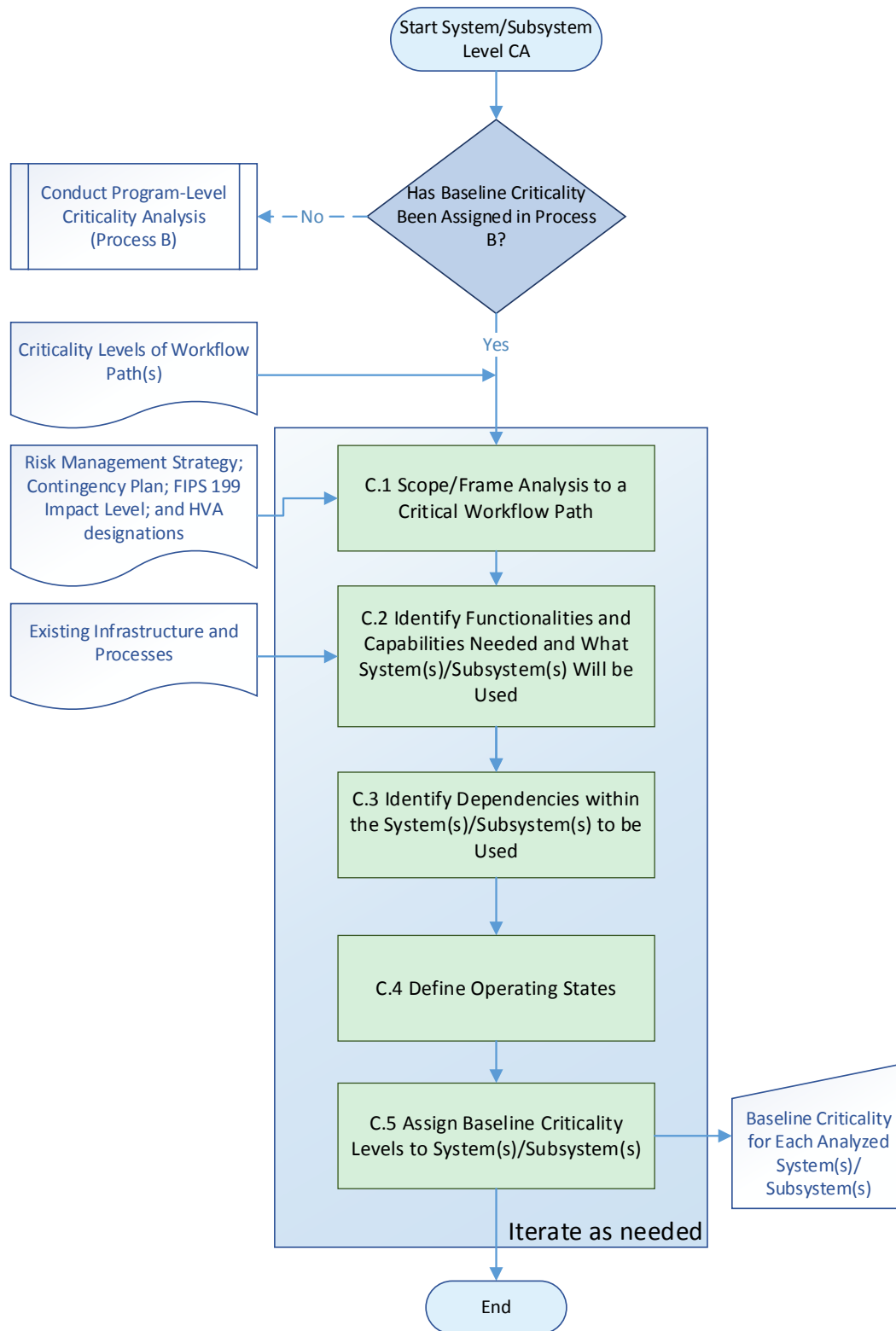
587

588 **Figure 4 - System/Subsystem-Level Criticality Analysis**

589 Process C, *Conduct System/Subsystem-Level Criticality Analysis,* depicted in Figure 4 is ideally is
590 performed after Process B, *Conduct Program-Level Criticality Analysis, is complete.* The process may be
591 repeated at increasingly granular levels in order to break a complex system down into its smallest parts, until
592 the lowest hierarchical level of subsystem is analyzed. A system or subsystem[5] may include multiple
593 components or subcomponents, often COTS products, each of which may require its own Criticality
594 Analysis to be performed in Process D. Similarly, one system may support numerous programs; this will be
595 discussed and analyzed in Process E.

596 Process C consists of:

597 • Scoping or framing the analysis to a critical workflow path or paths;
598 • Identifying functionalities/capabilities needed;
599 • Identifying systems/subsystems to be used;
600 • Defining what the system/subsystem will look like when it is operating normally or impacted by an
601   adverse event, that is referred to as an adverse operating state;
602 • Assigning Baseline Criticality to the workflow paths identified earlier.
603

604 Much of this process may be conducted as part of project planning, system design, and acquisition
605 processes.

Process	C
Process name	Conduct System/Subsystem-Level Criticality Analysis
Process summary	This process reviews and analyses the system or subsystem from the point of view of its criticality to the overall organizational goals.
Inputs	Documented Criticality Analysis Procedures (Process A); Final Criticality Levels of Activity(ies) and/or Workflow Path(s) of Program (Process B); Risk Management Strategy; Contingency Plan; FIPS 199 Impact Level; HVA designations; Existing Infrastructure and Processes

---

[5] System: Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions. See also information system. (SOURCE: CNSSI-4009)

Subsystem: A major subdivision or component of an information system consisting of information, information technology, and personnel that perform one or more specific functions. (SOURCE: SP 800-53; SP 800-53A; SP 800-37)

System: Combination of interacting elements organized to achieve one or more stated purposes.
   Note 1: There are many types of systems. Examples include: general and special-purpose information systems; command, control, and communication systems; crypto modules; central processing unit and graphics processor boards; industrial/process control systems; flight control systems; weapons, targeting, and fire control systems; medical devices and treatment systems; financial, banking, and merchandising transaction systems; and social networking systems.
   Note 2: The interacting elements in the definition of system include hardware, software, data, humans, processes, facilities, materials, and naturally occurring physical entities.
   Note 3: System of systems is included in the definition of system. (SOURCE: ISO/IEC/IEEE 15288)

Outputs	Baseline Criticality for Each Analyzed System/Subsystem
Roles and Responsibilities	Responsible: Lead System Architect or a similar role should be responsible for the performance of this sub-process. Lead Security Engineer should serve as a co-lead for sub-processes C.4, Define Operating States, and C.5, Assign Baseline Criticality Levels to System(s)/Subsystem(s). Accountable: Lead System Architect can delegate the execution of this process to another suitable individual, e.g. business analyst or systems analyst. Consulted: Individuals who have detailed knowledge of the activities identified by this process should participate in this process to contribute to the identification of critical activities. These individuals may include system architects and designers, system engineers, security engineers, other security professionals, acquisition/procurement professionals, business leaders, and others, as appropriate. Representatives of each relevant group should be invited to participate in this process. Informed: Individuals responsible for conducting any part of the Criticality Analysis.
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

606

607 **C.1** – **Scope/Frame Analysis to a Critical Workflow Path**

Sub-process number	C.1
Sub-process name	Scope/Frame Analysis to Critical Workflow Path
Sub-process summary	This sub-process involves identifying which critical process path will be examined further and is necessary for performing a criticality analysis for systems/subsystems. Ideally, it is performed once Baseline Criticality Levels of Activity(ies) and/or Workflow Path(s) have been determined. If criticality levels have not been determined, strongly consider returning to Process B, *Conduct Program-Level Criticality Analysis*. Using the Criticality Levels determined in Process B, identify which critical

	activities or workflows should be further analyzed. If more than one workflow path or set of related activities are determined critical in Process B, they should be analyzed separately in Process C unless they are very similar. If there are many activities in one workflow, identify similar types of activities, or activities grouped by the boundaries defined in Process B. Ensure the scope of process C is limited to a set of closely related activities. If Process B was not completed, if the organization wishes to focus on a certain type of system, or if the organization wishes to focus on a particular function, documents such as the Risk Management Strategy, Contingency Plans, FIPS 199 Impact Level, and High Value Asset (HVA) designations, may be useful to help identify systems that should be further analyzed and scope the analysis. Ensure the scope has definitive boundaries. Define any assumptions or constraints that will help limit the analysis.
Inputs	Criticality Levels of Activity(ies) and/or Workflow Path(s) from Process B. Other inputs: Risk Management Strategy; Contingency Plan; FIPS 199 Impact Level; and HVA designations
Outputs	Scope of analysis to pass to C.2.
Methods	Document Review; Context Diagram; Decision Analysis
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2)

608

609 **C.2 – Identify Functionalities and Capabilities Needed and What System(s)/Subsystem(s) Will be**
610 **Used**

Sub-process number	C.2
Sub-process name	Identify Functionalities and Capabilities Needed and What System(s)/Subsystem(s) Will be Used
Sub-process summary	This sub-process defines those functionalities and capabilities that are critical for successful operation of the system/subsystem.

	Review the activities identified in C.1; list the functionalities and capabilities that are needed to support that activity. Consider how the activity or related activities are currently conducted and identify any capabilities or tools that are used. Consider assigning initial values to each of the functionalities and capabilities to determine whether they are necessary to the successful completion of the activity, supportive to the activity, or if they are useful but not critical. Specifically identify functionalities and capabilities that are required by law, regulation, or policy. Also, identify any functionalities that directly support any security, safety, privacy, or similar goals. Finally, identify systems and subsystems that will be used to support required functionalities and capabilities.
Inputs	Scope from C.1; Existing Infrastructure and Processes
Outputs	List of functionalities and capabilities to pass to C.3.
Methods	Document Analysis; Brainstorming; Requirements Definition; Architecture Definition; Data Modeling; Data Flow Diagrams; Survey/Questionnaire.
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Asset Management (ID.AM-1, 2, 4) NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – System Requirements Definition Process (3.4.3) NIST SP 800-160 – Architecture Definition Process (3.4.4)

611

612 **C.3 – Identify Dependencies within the System(s)/Subsystem(s) to be Used**

Sub-process number	C.3
Sub-process name	Identify Dependencies within the System(s)/Subsystem(s) to be Used
Sub-process summary	This sub-process identifies the systems/subsystems to be used in the analysis. Determine whether there is available existing infrastructure sufficient to support the functions and capabilities described in C.2. Identify any functions or capabilities that are not supported by existing infrastructure. Determine what (if any) functions or capabilities will be supported by new systems/subsystems.

	Consider selecting a range of systems/subsystems that meet the functions and capabilities needed for the program, and rank them according to systems/subsystems that best provide the functions and capabilities noted as necessary.
Inputs	List of functionalities and capabilities from C.2; Existing Infrastructure and Processes
Outputs	List of dependencies to pass to C.4.
Methods	Document Review; Brainstorming; Questionnaire; Observation
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Asset Management (ID.AM-3)

NIST SP 800-160 – System Requirements Definition Process (3.4.3)

NIST SP 800-160 – Architecture Definition Process (3.4.4) |

613

614 **C.4 – Define Operating States**

Sub-process number	C.4
Sub-process name	Define Operating States
Sub-process summary	This sub-process defines scenarios for how the system/subsystem would operate normally and what would constitute abnormal operations of the system/subsystem.

Review the functions and capabilities defined in C.2. Describe the condition of the functions and capabilities (i.e., how will it operate). Determine the impact on both the system and the activity the system is intended to support (defined in C.2 and C.3) in each of the following conditions:

- Non-operational
- Impaired (i.e. the function or capability operates at a reduced pace or in an unsafe/insecure manner)
- Normal operation
- Increased operations (i.e. the function or capability performs quicker or with more output than normal)
- Unintended operations (i.e. the function or capability performs but with additional outputs or actions that are not part of the expected routine)

Consider defining what types of scenarios would lead to such situations. |

	Examples of scenarios to consider include reduced performance (e.g., lower bandwidth), security breach, physical accident, or any other similar event. Consider the security, safety, and privacy ramifications of these situations; for example, what information is made vulnerable if the function/capability performs slower than normal? Could there be physical damages if a function is performed too quickly?
Inputs	Process Description and/or Diagram; Listing of intersections and dependencies.
Outputs	Description of operating states to pass to C.5.
Methods	Document Review; Brainstorming; Interviews; Group Decision Making; Scenario/Use Case
Related processes outside of Criticality Analysis	NIST SP 800-39 – Assess NIST SP 800-161 – Assess NIST SP 800-160 – Architecture Definition Process (3.4.4)

615

616 **C.5 – Assign Baseline Criticality Levels to System(s)/Subsystem(s)**

Sub-process number	C.5
Sub-process name	Assign Baseline Criticality Levels to System(s)/Subsystem(s)
Sub-process summary	This sub-process determines criticality levels of system(s) and subsystems identified in C.3 against adverse states defined in C.4. Determine the severity of the operating states on the activity that the function/capability is intended to support. The user can rank systems and subsystems that are on the critical workflow path, perform vital functions and capabilities, and would be most impacted by adverse states. This could be a ranking (e.g. low, moderate, high), or measure (e.g. time lost; cost in time/resources; probability of being able to complete activity). The user could also use ranges and thresholds to define such rankings.[6]

---

[6] For further guidance on example security measures and metrics see Appendix A of NIST SP 800-55 Revision 1, *Performance Measurement Guide for Information Security.*

Inputs	Listing of intersections and dependencies from C.3; Listing of adverse states from C.4
Outputs	Baseline Criticality for each analyzed system/subsystem.
Methods	Document Review; Group Decision Making; Root Cause Analysis; Scenario/Use Case
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2) FIPS 199

617

618

619   **3.3   Process D – Conduct Component/Subcomponent-Level Criticality Analysis**

620
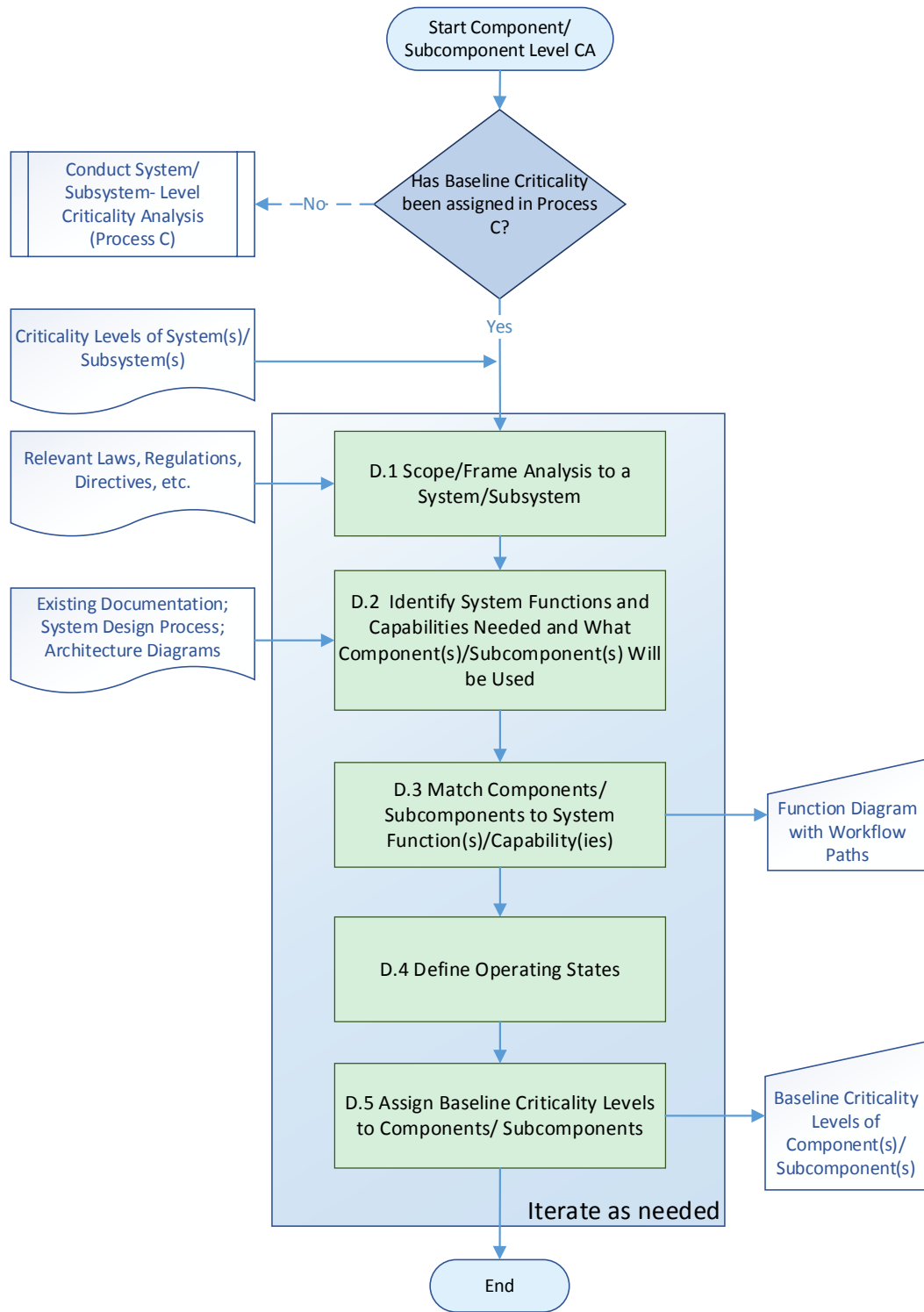
621

622   **Figure 5 - System/Subsystem-Level Criticality Analysis**

623 Process D, *Conduct Component/Subcomponent-Level Criticality Analysis*, depicted in Figure 5, is ideally
624 performed after Process C, *Conduct System/Subsystem-Level Criticality Analysis*, is complete. It is
625 performed at the component and subcomponent levels, as defined by the user. Oftentimes, the
626 components/subcomponents will be COTS products; as a result, this process will likely be performed out of
627 sequence and, at least partially, by a third party. The process may be repeated at increasingly granular levels
628 in order to break a complex set of components down into their smallest parts, until the lowest hierarchical
629 level of component is analyzed. As the components could be decomposed into extremely fine detail (e.g.
630 raw materials), it is important to define what level of granularity is necessary for this analysis.

631 Process D consists of:

632 • Scoping/framing the analysis to a specific system or subsystem
633 • Identifying system functionalities, capabilities, and pathways needed to fulfil functional
634   requirements;
635 • Matching components and subcomponents to the identified system functionalities, capabilities, and
636   pathways;
637 • Defining normal operating conditions and those conditions that system/subsystem will be operating
638   sub-optimally, referred to as adverse operating states; and
639 • Assigning Baseline Criticality to the components and subcomponents identified earlier
640

641 Much of this process is conducted as part of system architecture and design processes.

Process	D
Process name	Conduct Component/Subcomponent-Level Criticality Analysis
Process summary	This process reviews and analyzes a specific system to identify critical components and/or subcomponents.
Inputs	Criticality Levels of System(s)/Subsystem(s); Relevant Laws, Regulations, Directives, etc.; Existing Documentation; System Design Process; Architecture Diagrams
Outputs	Function Diagram with Workflow Pathways; Baseline Criticality Levels of Component(s)/Subcomponent(s)
Roles and Responsibilities	Responsible: Lead System Engineer or a similar role should be responsible for the performance of this sub-process. Lead Security Engineer should serve as a co-lead for sub-processes D.4, Define Operating States, and D.5, Assign Baseline Criticality Levels to Components/Subcomponents. Accountable: Lead System Engineer or a similar role can delegate the execution of this process to another suitable individual, e.g. system analyst. Consulted: Individuals who have detailed knowledge of the activities identified by this process should participate in this process to contribute to the

	identification critical activities. These individuals may include system architects and designers, system engineers, security engineers, other security professionals, acquisition/procurement professionals, business leaders, and others, as appropriate. Representatives of each relevant group should be invited to participate in this process. Informed: Individuals responsible for conducting any part of the Criticality Analysis.
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-39 – Assess NIST SP 800-161 – Assess NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

642

643 ### **D.1 – Scope/Frame Analysis to a System/Subsystem**

Sub-process number	D.1
Sub-process name	Scope/Frame Analysis to a System/Subsystem
Sub-process summary	This sub-process narrows the scope of the analysis to a specific system or subsystem. Ideally, it is performed once Baseline Criticality Levels of systems/subsystems have been determined. If criticality levels have not been determined, strongly consider returning to Process C, *Conduct System/Subsystem-Level Criticality Analysis.* In the case of COTS products, the process will likely be performed out of sequence. Using the criticality levels determined in Process C, identify which critical system/subsystem should be further analyzed. For the purposes of this Criticality Analysis, the system/subsystem should be an IT/OT product, device, or solution, although the Model will support the analysis of any well-defined system. Separate analyses should be conducted for all critical systems identified in Process C, if possible. This is because the components comprising systems are often varied even if the systems seem identical. Ensure the scope has definitive boundaries. Define any assumptions or constraints, which will help, limit the analysis. Components and subcomponents are sometimes guided by specific legal and

	regulatory requirements, such as sourcing requirements (where those can/cannot come from); take those into account. If the analysis is conducted by a third-party, such as in the case of COTS, work with the COTS provider(s) to define what information is available which may serve to inform a Baseline Criticality determination, including system documentation, risk analyses performed, operating constraints, and assumptions. If a system does not exist, but is being designed or is under development, bear in mind that the system design may change frequently. It may be best to perform this analysis from a theoretical viewpoint and use the result to inform the design and development process. Then repeat the process when the system development has been completed.
Inputs	Criticality Levels of Systems/Subsystems from Process C; Relevant Laws, Regulations, Directives, and other documents that may contain requirements that describe anything to do with the components that are being used in this system.
Outputs	Determination of the system/subsystem to focus analysis.
Methods	Document Review; Survey; Interviews
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

644

645 **D.2 – Identify System Functions and Capabilities Needed and What Components/Subcomponents**
646 **Will be Used**

Sub-process number	D.2
Sub-process name	Identify System Functions and Capabilities Workflow Needed and What Components/Subcomponents Will be Used
Sub-process summary	This sub-process analyzes the system to identify the components and subcomponents required to ensure the system functions as intended. Review the activities identified in D.1; list the functions and capabilities of the system being analyzed. If the system is complex, consider scoping the analysis to only the functions and capabilities determined as critical. Define

	the processes that are activated or that the system uses in order to perform these functions and capabilities. Those can be extracted from existing system documentation, such as functional requirements, system diagrams, process flow diagrams, system concept of operations, or any other documentation that describes what the system does. The way the system executes a specific function or capability by handing them from one process to another is a workflow path. Identify all workflow paths required to execute each function or, if the system is complex, each critical function. Finally, identify components and subcomponents that will be used to support required functionalities and capabilities.
Inputs	Determination of the system/subsystem to focus analysis from D.1; Existing documentation, system design process, architecture diagrams.
Outputs	Listing of capabilities and pathways needed to pass to D.3.
Methods	Document Review; Process Analysis; Systems Analysis; Workflow Analysis; Data Flow Diagrams; Functional Decomposition; Interface Analysis
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

647

648 **D.3 – Match System Components/Subcomponents to System Function(s)/Capability(ies)**

Sub-process number	D.3
Sub-process name	Match System Components/Subcomponents to System Function(s)/Capability(ies)
Sub-process summary	This sub-process identifies specific components and subcomponents that support the Workflow paths identified in D.2 and the associated system functions and capabilities, defined in process C. For each workflow path identified in D.2, identify the components and subcomponents that are or would be required for each workflow path to be executed. In many cases, a single component, or identical components, will be used to support multiple workflow paths. Document these components matched to workflow paths in a matrix, spreadsheet, database, function

	diagram, or a similar tool.
Inputs	Listing of capabilities and pathways from D.2. Existing documentation, system design, lists of components, bill of materials, other documentation that somehow describes components and subcomponents.
Outputs	Listing of components and subcomponents matched to workflow paths to pass to D.4 and D.5; Function Diagram with Workflow Paths.
Methods	Document Review; Systems Analysis; Brainstorming
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

649

650   **D.4 – Define Operating States**

Sub-process number	D.4
Sub-process name	Define Operating States
Sub-process summary	This sub-process defines normal operational states, as well as the states in which the system will be operating abnormally. Review the outputs of D.3. Describe operation of the components and/or workflow paths. Then determine the impact of the various operating states of the component on the workflow path(s) and consequently on the system function/capability that path supports. Consider each of the following operating states: • Non-operational • Impaired (i.e. the component/subcomponent operates at a reduced capability or in an unsafe/insecure manner) • Normal operation • Increased operations (i.e. the function or capability performs quicker or with more output than normal)

	• Unintended operations (i.e. the function or capability performs but with additional outputs or actions that are not part of the expected routine) Consider defining what types of scenarios would lead to such situations. Examples of scenarios to consider include reduced performance (e.g., lower bandwidth), security breach, physical accident, or any other similar event. Using the Function Diagram with Workflow Paths, identify specific points within the workflow paths where the system will be particularly stressed as a result of any of these operating states. This would include any points that would exacerbate the situation. Define the severity of the impact. This may be a ranking (e.g. low, moderate, high), or measure (e.g. processing speed; downtime; percentage of remaining functionality). Consider the security, safety, and privacy ramifications of these situations; for example, what information is made vulnerable if the components/subcomponents fails?
Inputs	Listing of components and subcomponents matched to workflow paths from D.3; Function Diagram with Workflow Paths.
Outputs	Description of operating states to pass to D.5.
Methods	Document Review; Systems Analysis; Workflow Analysis; Brainstorming; Group Decision Making; Scenario/Use Case
Related processes outside of Criticality Analysis	NIST SP 800-39 – Assess NIST SP 800-161 – Assess NIST SP 800-160 – Design Definition Process (3.4.5)

651

652 **D.5 – Assign Baseline Criticality Levels to Components/Subcomponents**

Sub-process number	D.5
Sub-process name	D.5 Assign Baseline Criticality Levels to Components/Subcomponents
Sub-process summary	This sub-process assigns criticality levels to components and subcomponents identified in D.3 based on the impact of the operating states defined in D.4. Rank the components and subcomponents in a way that gives the highest ranking to components and subcomponents that are on the critical workflow

	path, perform vital functions and capabilities, and would be most impacted by adverse operating states. The user can create a ranking schema that would, for example, rank those activities and workflow paths that are impacted by the highest number of scenarios as High Criticality and those that are impacted by the lowest number of scenarios as Low Criticality. The user could also use ranges and thresholds to define such rankings.
Inputs	Listing of components and subcomponents matched to workflow paths from D.3; Function Diagram with Workflow Paths; Description of operating states from D.4.
Outputs	Baseline Criticality Levels of Component(s)/Subcomponent(s).
Methods	Document Review; Systems Analysis; Process Flow Analysis; Group Decision Making; Root Cause Analysis; Scenario/Use Case
Related processes outside of Criticality Analysis	NIST SP 800-39 – Frame NIST SP 800-161 – Frame FIPS 199

653

654

655 **3.4 Process E – Conduct Detailed Review of Criticality for Processes B, C, and D**
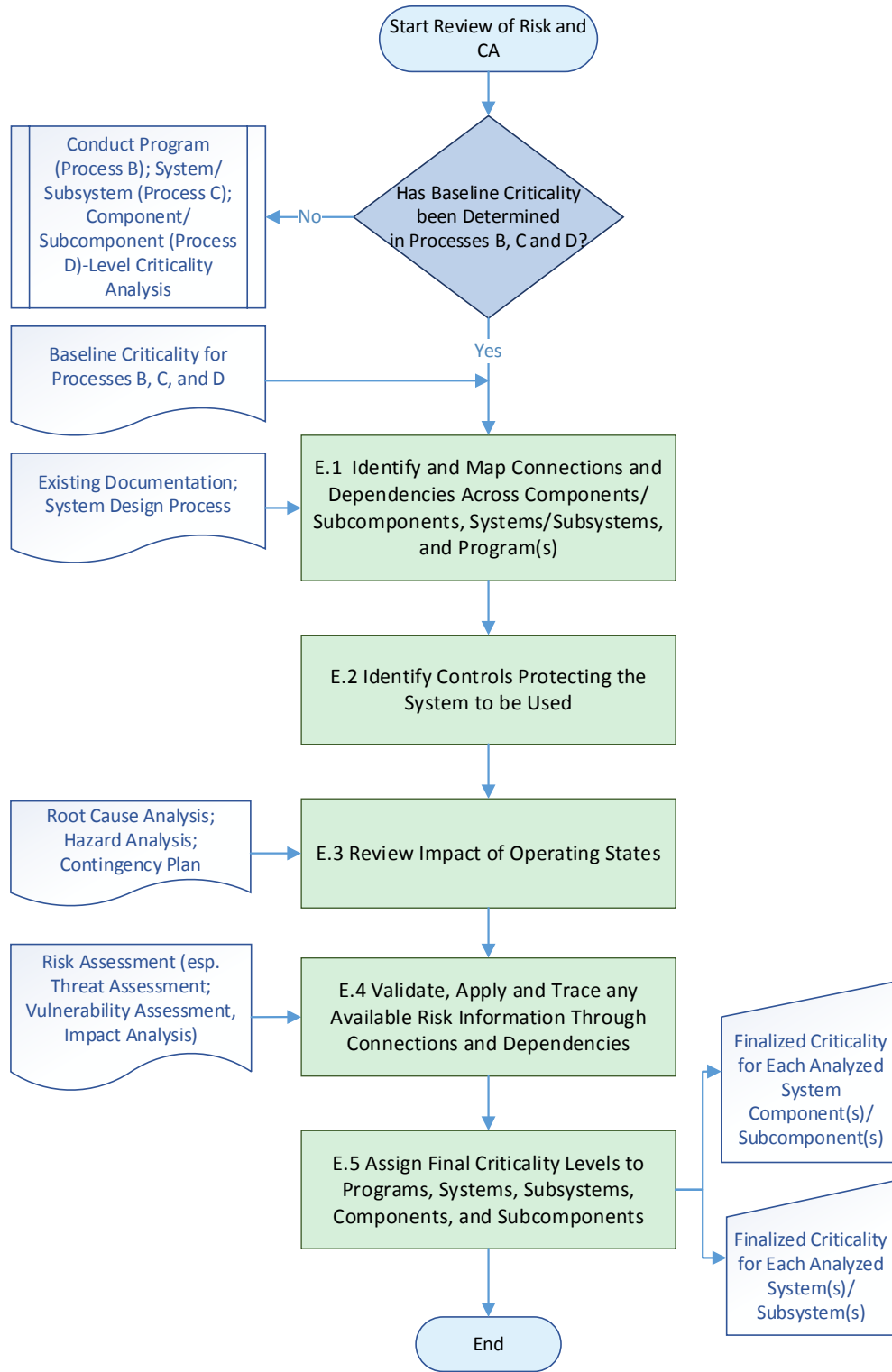
656

657 **Figure 6 - Conduct Detailed Review of Criticality for Processes B, C, and D**

658 Process E*, Conduct Detailed Review of Criticality for Processes B, C, and D* depicted in Figure 6, provides
659 a bottom-up review of impacts and ensures cross-process interaction and collaboration. Process E consists
660 of:

661 • Identifying and mapping connections, and dependencies across Program; System/Subsystem;
662 or Component/Subcomponent;
663 • Identifying controls protecting the system to be used;
664 • Reviewing impact of Operating States; and
665 • Validating, applying and tracing any available risk information through connections and
666 dependencies.

667 Process E is a bottom-up process where information is iteratively validated across the entire Model. Process
668 E is performed after sub-processes B.5, C.5, and D.5. The output of these sub-processes, Baseline
669 Criticality, is used as an input to Process E.

670 The information on Baseline Criticality for components/subcomponents is used to validate criticality for
671 systems/subsystems. Please note that program level baseline criticality does not need to be revised here and
672 is instead an input only. This process iterates until the three criticalities are consistent and harmonized.
673 When the validation is complete, then the user can finalize criticality levels for systems/subsystem, and
674 components/subcomponents.

Process	E
Process name	Conduct Detailed Review of Criticality for Processes B, C, and D
Process summary	This is a bottom-up sub-process conducted after Baseline Criticality levels have been defined under Processes B, C and D. It is used to create final criticality levels for Systems/Subsystems and Components/subcomponents. This process involves identifying connections and dependencies across Processes B, C, and D. It considers any available risk information, including any existing mitigation strategies, to create a more precise criticality score.
Inputs	Baseline Criticality for B, C, and D; Existing Documentation; System Design Process; Security Requirements; Functional Requirements; Root Cause Analysis; Hazard Analysis; Risk Assessment (esp. Threat Assessment; Vulnerability Assessment, Impact Analysis).
Outputs	Criticality Levels of Programs, Systems/Subsystems, and Component(s)/ Subcomponent(s).
Roles and Responsibilities	Responsible: Lead System Architect or a similar role should be responsible for the performance of this sub-process. Lead Security Engineer should serve as a co-lead for this process. Accountable: Lead System Engineer should work in partnership with Lead Security Engineer and Program Manager to ensure appropriate communication

	and collaboration across Processes B, C, and D. Consulted: Individuals who have detailed knowledge of the activities identified by this process should participate in this process to contribute to the identification of critical activities. These individuals may include system architects and designers, system engineers, security engineers, other security professionals, acquisition/procurement professionals, business leaders, and others, as appropriate. Representatives of each relevant group should be invited to participate in this process. Informed: The outputs of this process should be shared with individuals performing any part of Criticality Analysis process.
Related processes outside of Criticality Analysis	NIST SP 800-39 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) NIST SP 800-161 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) FIPS 199 NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

675

676   **E.1 – Identify and Map Connections and Dependencies Across Components/ Subcomponents,**
677   **Systems/Subsystems, and Program(s)**

Sub-process number	E.1
Sub-process name	Identify and Map Connections and Dependencies Across Components/ Subcomponents, Systems/Subsystems, and Program(s)
Sub-process summary	This sub-process uses the process diagrams, design documents or other artifacts created in processes B, C, and D, to trace sub-components through to program goals and objectives. One system component or type of system component may be used in multiple sub-systems. Identify these by reviewing the system design documentation that was created in Process D for each system or sub-system that was identified in Process C. Similarly, one system may support multiple programs. Identify these by reviewing the systems identified in Process C for each workflow described in Process B.

	Identify identical or similar types of components used for critical functions of multiple systems. Also, identify components or sub-systems that originate from a single supplier. Look for any other connection or dependency that may impact the success of the objective or goals if stretched by maintenance, supply chain, security, or other concerns.
Inputs	Baseline Criticality Levels of Program, System(s)/Subsystem(s), and Component(s)/ Subcomponent(s); Existing Documentation and System Design Process.
Outputs	Identification and maps of connections and dependencies across Program, System/Subsystem, and Component/Subcomponent to pass to E.2.
Methods	Document Review; Mission Thread Analysis; Impact Analysis; Hazard Analysis
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Asset Management (ID.AM-3) NIST SP 800-39 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) NIST SP 800-161 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) FIPS 199 NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

678

679 **E.2 – Identify Controls Protecting the System to Be Used**

Sub-process number	E.2
Sub-process name	Identify controls protecting the system to be used
Sub-process summary	This sub-process is used to identify components, system functions, processes, or other measures that are used to ensure the system operates within acceptable parameters. Beginning at the sub-component level, identify all controls that ensure the program, systems, and components operate within acceptable parameters. Identify system components that monitor or protect critical subcomponents.

	Then review critical components and identify any system functions that provide those same assurances. Next, identify external systems, programmatic activities, processes, procedures, and practices that serve to monitor or protect the system. Identify any programmatic activities that serve to monitor or protect the program itself. Using the connections and dependencies identified in E.1, *Identify and Map Connections and Dependencies across Program, System/Subsystem, and Component/Subcomponent*, identify controls that monitor and protect those connections and dependencies.
Inputs	Identification and maps of connections and dependencies across Program, System/Subsystem, and Component/Subcomponent from E.1; Security Requirements; Functional Requirements
Outputs	Listing of controls protecting the system to be used to pass to E.3.
Methods	Document Review; Security Control Selection and Allocation (Risk Management)
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Governance (ID.GV); Risk Assessment (ID.RA-6); Risk Management Strategy (ID.RM) NIST SP 800-39 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) NIST SP 800-161 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) FIPS 199 NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

680

681 **E.3 – Review Impact of Operating States**

Sub-process number	E.3
Sub-process name	Review Impact of Operating States
Sub-process summary	Beginning at the sub-component level, trace the impact of each operating state each adverse operating state at the system level (defined in C.4) would have on the operations of the activity or workflow path it is meant to support (defined in

	B), and what each adverse operating state at the activity level would have on the success of the program (defined in B.4). Using the controls identified in E.2, review the likelihood of the adverse operating states and associated impact(s) occurring. For example, if there are no controls monitoring and protecting a vital component, this may need to be reflected in the criticality level of the component.
Inputs	Listing of controls protecting the system to be used from E.2; Descriptions of operating states from B.4, C.4, and D.4; Results of Root Cause Analysis; Results of Hazard Analysis; Contingency Plans
Outputs	Refined list of operating states to pass to E.4.
Methods	Document Analysis; Scenario/Use Case; Hazard Analysis.
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Risk Assessment (ID.RA-4) NIST SP 800-39 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) NIST SP 800-161 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) FIPS 199 NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

682

683 **E.4 – Validate, Apply and Trace any Available Risk Information Through Connections and**
684 **Dependencies**

Sub-process number	E.4
Sub-process name	Validate, Apply and Trace any Available Risk Information Through Connections and Dependencies
Sub-process summary	If available, apply any threat, vulnerability, or other risk information to the connections and dependencies mapping and increase or decrease the criticality level of the system or component as appropriate.
Inputs	Refined list of adverse operating states from E.3; Risk Assessment (esp. Threat Assessment; Vulnerability Assessment, Impact Analysis).

Outputs	Detailed review results to pass to E.5, *Assign Final Criticality Levels to Systems, Subsystems, Components, and Subcomponents*
Methods	Document Review; Risk Analysis; Brainstorming
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Risk Assessment (ID.RA) NIST SP 800-39 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) NIST SP 800-161 – Assess (Risk Assessment, Threat Assessment; Vulnerability Assessment, Impact Analysis) FIPS 199 NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

685
686   **E.5 – Assign Final Criticality Levels to Systems, Subsystems, Components, and Subcomponents**

Sub-process number	E.5
Sub-process name	Assign Final Criticality Levels to Systems, Subsystems, Components, and Subcomponents
Sub-process summary	This sub-process finalizes Baseline Criticality levels determined in processes C and D. The critical nature of components and systems may be influenced by the dependencies, connections, controls, and impacts identified in Process E. Review the Baseline Criticality levels defined in C.5 and D.5. In light of the outputs of Process E, refine the rankings. Consider ranking: - Components and subcomponents by their importance in keeping the system from entering adverse operating states or keeping the system operational while in adverse operating states; and - Systems and subsystems by their importance in keeping the program from entering adverse operating states or keeping the program running while in adverse operating states. Avoid reducing the criticality scores of systems and components without carefully considering how this may impact purchasing and management decisions. A scoring system may be defined that uses the information from Process E to refine but not alter the Baseline Criticality levels. For example,

	Baseline Criticality levels could be given a digit identifier from 1-5 while results of the review conducted in Process E add a digit 0-9 to that identifier, so the final identifier would be a two-digit ranking from 10 to 59. Whatever method is used to score the criticality levels, ensure the method is sufficiently detailed so that a reasonably small number of components are given a high criticality score. Using the process described in this publication, a large number of components should not be given a criticality score – it should be assumed that these components either are outside the scope or control of the program or do not have high criticality.
Inputs	Detailed Review Results from Sub-Process E, *Conduct Detailed Review of Risk and Criticality Analysis.*
Outputs	Finalized criticality for each analyzed component/subcomponent, system/subsystem, etc.
Methods	Document Review; Brainstorming; Group Decision Making
Related processes outside of Criticality Analysis	NIST Framework for Improving Critical Infrastructure – Asset Management (ID.AM-5) NIST SP 800-39 – Assess NIST SP 800-161 – Assess NIST SP 800-160 – Business or Mission Analysis Process (3.4.1) NIST SP 800-160 – Stakeholder Needs and Requirements Definition Process (3.4.2) FIPS 199 NIST SP 800-160 – Architecture Definition Process (3.4.4) NIST SP 800-160 – Design Definition Process (3.4.5)

687

688 **Appendix A—Acronyms**

689 Selected acronyms and abbreviations used in this paper are defined below.

BABOK®	Business Analysis Body of Knowledge®
CA	Criticality Analysis
COTS	Commercial Off the Shelf
DoD	Department of Defense
FIPS PUB	Federal Information Processing Standard Publication
FMECA	Failure Mode Effects and Criticality Analysis
HVA	High Asset Value
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
ISO/IEC/IEEE	International Organization for Standardization/International Electrotechnical Commission/ Institute of Electrical and Electronics Engineers
IT	Information Technology
IT SCRM	Information Technology Supply Chain Risk Management
ITL	Information Technology Laboratory
IT/OT	Information Technology/Operations Technology
NIST	National Institute of Standards and Technology
NIST SP	National Institute of Standards and Technology Special Publication
NISTIR	National Institute of Standards and Technology Interagency Report
OT	Operations Technology
PMBOK®	Project Management Body of Knowledge®

SP   Special Publication

SCRM   Supply Chain Risk Management

SOS   System of Systems

US   United States

690

691 **Appendix B—References**

B.1   Sources for the Model

[1]   International Institute of Business Analysis, *A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) v 3,* April 15, 2015,

[2]   Warwick Manufacturing Group, Product Excellence Using Six Sigma, *Section 12: Failure Modes, Effects & Criticality Analysis, University of Warwick, Coventry, CV4 7AL, UK,* 32pp, http://www2.warwick.ac.uk/fac/sci/wmg/ftmsc/modules/modulelist/peuss/slides/section_12a_fmeca_Notes.pdf [accessed 6/5/2017].

[3]   H. Salim and S. Madnick, *Cyber Safety: A Systems Thinking and Systems Theory Approach to Managing Cyber Security Risks,* Working Paper CISL# 2014-12, Massachusetts Institute of Technology, Cambridge, MA, September 2014, 157 pp, http://web.mit.edu/smadnick/www/wp/2016-09.pdf [accessed 6/5/2017].

[4]   H. Salim and S. Madnick, *Cyber Safety: A Systems Theory Approach to Managing Cyber Security Risks – Applied to TJX Cyber Attack,* Working Paper CISL# 2016-09, Massachusetts Institute of Technology, Cambridge, MA, August 2016, 17pp, http://web.mit.edu/smadnick/www/wp/2016-09.pdf [accessed 6/5/2017]

[5]   N. Leveson, "An STPA Primer, Version 1,*"* August 2013*,* 80pp http://sunnyday.mit.edu/STPA-Primer-v0.pdf [accessed 6/5/17].

[6]   N. Leveson, "Engineering a Safer and More Secure World," *MIT Press*, June 2011, 72pp http://psas.scripts.mit.edu/home/wp-content/uploads/2016/04/STAMP-Intro-2016.pdf [accessed 6/5/17].

[7]   Dan Reddy, "Criticality Analysis & Supply Chain: Providing "Representational Assurance," paper presented at the RSA Conference, Moscone Center, San Francisco, February 24-28, 2014, https://www.rsaconference.com/writable/presentations/file_upload/str-w04a-criticality-analysis-supply-chain_v2.pdf [accessed 6/5/17].

[8]   Dan Reddy, "Criticality analysis and the supply chain: Leveraging representational assurance," *Science Direct*, May 9, 2014, accessed June 5, 2017, doi.org/10.1016/j.technovation.2014.01.009

[9]   National Electric Sector Cybersecurity Organization Resource (NESCOR), *Electric Sector Failure Scenarios and Impact Analyses – Version 3.0,* December 2015. http://smartgrid.epri.com/doc/NESCOR%20Failure%20Scenarios%20v3%2012-11-15.pdf [accessed 6/5/2017]

[10]   INCOSE, *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities,* 4th Edition, [New York: Wiley, 2015], Chapters 4, 6, 10.

[11]   Department of Defense, *Defense Acquisition Guidebook: Chapter 13 – Program Protection Plan,* March 2012, 70pp

[12]   M. Reed, "System Security Engineering and Comprehensive Program Protection," presented at the 16th Annual NDIA Systems Engineering Conference, Arlington, VA, October 30, 2013 (Revised 4/17/2104), http://www.acq.osd.mil/se/briefs/2013_10_30_NDIA-SEC-Reed-ProgramProtection-Approved.pdf [accessed 6/5/2017].

[13]   M. Reed, "Comprehensive Program Protection Planning for the Materiel Solution Analysis (MSA) Phase*,*" presented at the 15th Annual NDIA Systems Engineering Conference, San Diego, CA, October 24, 2012, http://www.acq.osd.mil/se/briefs/14761-2012_10_24-NDIA-SEC-Reed-SSE-PP-MSA-Phase.pdf [accessed 6/5/2017].

[14]   Dr. Jean-Claude Franchitti, *Session 1 – Sub-Topic 1Requirements Definition & Management Processes and Tools,* Software Engineering, G22.2440-001*,* New York University, 21 pp*,* http://www.nyu.edu/classes/jcf/g22.2440-001_sp09/slides/session2/g22_2440_001_c22.pdf [accessed 6/5/2017]

[15]   A Guide to Fault Detection and Diagnosis, Greg Stanley and Associates [Web site] http://gregstanleyandassociates.com/whitepapers/FaultDiagnosis/faultdiagnosis.htm [accessed 6/5/2017].

[16]   SAE International, *ARP4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment,* 331 pp, December 1, 1996, http://standards.sae.org/arp4761/ [accessed 6/5/2017].

[17]   Creating A Value Stream Map, Lean Manufacturing Tool [Web site], http://leanmanufacturingtools.org/551/creating-a-value-stream-map/ [accessed 6/5/2017].

[18]   Value-Stream Mapping for Manufacturing, Lean Enterprise Institute, [Web site] https://www.lean.org/Workshops/WorkshopDescription.cfm?WorkshopId=7 [accessed 6/5/2017

[19]   Factor Analysis of Information Risk, FAIR Institute [Web site] *http://www.fairinstitute.org* [accessed 6/5/2017].

[20]   Merriam-Webster [Web site], https://www.merriam-webster.com/dictionary/ [accessed 6/14/2017].

[21]   McAllister, Richard K., and James L. Coyle. "Interdependency analysis." In
         Proceedings of 22nd NIST-NCSC national information systems security
         conference Arlington, VA NIST, pp. 403-414. 1999,
         http://csrc.nist.gov/nissc/1999/proceeding/papers/p31.pdf , [accessed
         6/14/2017].

[22]   M. Gagliardi, W. Wood and T. Morrow, Introduction to the Mission Thread
         Workshop, Carnegie Mellon University, Software Engineering Institute,
         CMU/SEI-2013-TR-003 Software Engineering Institute, 44 pp, October
         2013,
         http://repository.cmu.edu/cgi/viewcontent.cgi?article=1762&context=sei
         [accessed 6/14/2017]

[23]   Project Management Institute, *Project Management Body of Knowledge®
         (PMBOK® Guide) 5th edition,* 2014

[24]   American Society for Quality (ASQ), [Web site], http://asq.org/learn-about-
         quality/process-analysis-tools/overview/flowchart.html [accessed 6/14/2017]

[25]   Six Sigma Daily, [Web site], http://www.sixsigmadaily.com/ [accessed
         6/14/2017]

[26]   U.S. Department of Transportation Pipeline and Hazardous Materials Safety
         Administration, [Web site], https://www.phmsa.dot.gov/ [accessed
         6/15/2017]

[27]   The Open Group, *The Open Group Architecture Framework TOGAF^TM* ,
         Version 9, 2009

[28]   U.S. Department of Health and Human Services, Enterprise Performance
         Life Cycle Framework Practices Guide: Requirements Definition, [Web site]
         https://www.hhs.gov/ocio/eplc/Enterprise%20Performance%20Lifecycle%2
         0Artifacts/eplc_artifacts.html [accessed 6/15/2017]

[29]   Healthcare Information and Management Systems Society (HIMSS), [Web
         site], http://www.himss.org/ [accessed 6/15/2017]

[30]   Kfir Eliaza, Debraj Rayb and Ronny Razinc, "Group decision-making in the
         shadow of disagreement Journal of Economic Theory," Journal of Economic
         Theory, 0022-0531 (2005), accessed July 15, 2017,
         doi:10.1016/j.jet.2005.07.008.

B.2   Related Standards & NIST Publication

[FIPS PUB   Federal Information Processing Standards Publication (FIPS PUB) 199,
199]   *Standards for Security Categorization of Federal Information and
                 Information Systems,* National Institute of Standards and Technology,
                 Gaithersburg, Maryland, February 2004, 13pp,

[1]   NIST Framework for Improving Critical Infrastructure Cybersecurity.
                 Version 1.0. February 12, 2014. https://www.nist.gov/cyberframework

[IR7298]   NIST Interagency Report (IR) 7298, Revision 2, *Glossary of Key
                 Information Security Terms,* National Institute of Standards and Technology,
                 Gaithersburg, Maryland, May 2013, 222pp,
                 http://nvlpubs.nist.gov/nistpubs/ir/2013/NIST.IR.7298r2.pdf

[SP800-161]   NIST Special Publication (SP) 800-161, *Supply Chain Risk Management
                 Practices for Federal Information Systems and Organizations*, National
                 Institute of Standards and Technology, Gaithersburg, Maryland, April 2015,
                 282pp, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-
                 161.pdf

[SP800-160]   NIST Special Publication (SP) 800-160 *Systems Security Engineering
                 Considerations for a Multidisciplinary Approach in the Engineering of
                 Trustworthy Secure Systems,* National Institute of Standards and
                 Technology, Gaithersburg, Maryland, September 2016, 262pp,
                 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-160.pdf

[SP800-55]   NIST SP 800-55 Revision 1 (Rev 1), *Performance Measurement Guide for
                 Information   Security,*   July   2008,   80pp,
                 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-
                 55r1.pdf

[SP800-53]   NIST Special Publication (SP) 800-53 Revision 4 (Rev 4), *Security and
                 Privacy Controls for Federal Information Systems and Organizations,*
                 National Institute of Standards and Technology, Gaithersburg, Maryland,
                 September 2016, 462pp,
                 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[SP800-39]   NIST Special Publication (SP) 800-39, *Managing Information Security Risk:
                 Organization, Mission, and Information System View,* National Institute of
                 Standards and Technology, Gaithersburg, Maryland, March 2011, 88pp,
                 http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf

[SP800-37]   NIST Special Publication (SP) 800-37, *Guide for Applying the Risk
                 Management Framework to Federal Information Systems: A Security
                 Lifecycle Approach,* National Institute of Standards and Technology,
                 Gaithersburg, Maryland, February 2010, 102pp,
                 http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-37r1.pdf

[SP800-30]   NIST Special Publication (SP) 800-30, *Guide for conducting risk assessments*, National Institute of Standards and Technology, Gaithersburg, Maryland, September 2012, 95pp, http://csrc.nist.gov/publications/drafts/800-160/sp800_160_final-draft.pdf

[2]   International Organization for Standardization/International Electrotechnical Commission, *Information technology – Security techniques – Information security for supplier relationships – Part 1: Overview and concepts,* ISO/IEC 27036-1:2014, 2014. https://www.iso.org/standard/59648.html

[3]   International Organization for Standardization/International Electrotechnical Commission, *Information technology – Security techniques – Information security management system – Requirements*, ISO/IEC 27001:2013, 2013. https://www.iso.org/standard/54534.html

[4]   International Organization for Standardization/International Electrotechnical Commission, *Information technology – Security techniques – Information security management system – Requirements*, ISO/IEC 27001:2013, 2013. https://www.iso.org/standard/54534.html

[5]   International Organization for Standardization/International Electrotechnical Commission, *Information technology – Security techniques – Code of practice for information security controls*, ISO/IEC 27002:2013, 2013. https://www.iso.org/standard/54533.html

[6]   International Organization for Standardization/International Electrotechnical Commission/ Institute of Electrical and Electronics Engineers, *Information technology – System and software lifecycle processes*, ISO/IEC/IEEE 15288:2015, 2015. https://www.iso.org/standard/63711.html

692

693 **Appendix C—Methods**

694 Activity Network Diagram – "an activity network diagram also known as arrow diagram, pert chart,
695 and critical path method is used to show activities that are in parallel and/or in series. It will show
696 the most likely times, the most pessimistic times, and the most likely times for the completion of
697 projects" (For more information: Six Sigma Daily).

698 Architecture Definition – "provides a formal model of the Baseline Architecture, Target
699 Architecture, and the gaps between the two states" (For more information: The Open Group
700 Architecture Framework TOGAF^{TM}).

701 Brainstorming or Brainstorming of Activities – "a team activity that seeks to produce a broad or
702 diverse set of options through the rapid and uncritical generation of ideas" (For more information:
703 BABOK®).

704 Context Diagram – "an analysis model that illustrates product scope by showing the system in its
705 environment with the external entities (people and systems) that give to and receive from the
706 system" (For more information: BABOK®).

707 Security Control Selection and Allocation –A process for identifying what security measures are or
708 will be taken. (For more information: NIST SP 800-37 Rev 1)Critical Function Identification –
709 method to identify critical functions in a system/subsystem (For more information: Defense
710 Acquisition Guidebook, Chapter 13 – Program Protection Plan).

711 Data Flow Diagrams – "an analysis model that illustrates processes that occur, along with the flows
712 of data to and from those processes" (For more information: BABOK®).

713 Data Modeling – describes the concepts and relationships relevant to the solution or business domain
714 (For more information: BABOK®).

715 Decision Analysis – "an approach to decision-making that examines and models the possible
716 consequences of different decisions. Decision analysis assists in making an optimal decision under
717 conditions of uncertainty" (For more information: BABOK®).

718 Document Analysis – "a means to elicit requirements of an existing system by studying available
719 documentation and identifying relevant information" (For more information: BABOK®).

720 Document Review – data collection method for the review of documentation received throughout the
721 criticality analysis process(es)/sub-process(es) (For more information: BABOK®).

722 Functional Decomposition – to decompose processes, functional areas, or deliverables into their
723 component parts and allow each part to be analyzed independently (For more information:
724 BABOK®).

725 Gantt Chart – a visual representation of a project schedule. A Gantt chart is a type of bar chart in
726 which a series of horizontal lines shows the amount of work done or production completed in certain
727 periods of time in relation to the amount planned for those periods (For more information:
728 PMBOK®).

729   Group Decision Making (also known as collaborative decision-making) – "process by which a
730   collective of individuals attempt to reach a required level of consensus on a given issue (For more
731   information: Journal of Economic Theory)

732   Hazard Analysis – "the identification of material properties, system elements, or events that lead to
733   harm or loss. The term hazard analysis may also include evaluation of consequences from an event
734   or incident." (For more information: U.S. Department of Transportation Pipeline and Hazardous
735   Materials Safety Administration)

736   Interdependency Analysis – "a technique for evaluating security service strengths of combinations of
737   security mechanisms employed to protect information. Such a technique can provide a valuable tool
738   for assessing the security architectures and implementations of information systems." (For more
739   information: Interdependency Analysis)

740   Interface Analysis – elicitation technique used "to identify interfaces between solutions and/or
741   solution components and define requirements that describe how they will interact" (For more
742   information: BABOK®).

743   Interviews – "a systematic approach designed to elicit information from a person or group of people
744   in an informal or formal setting by talking to an interviewee, asking relevant questions and
745   documenting the responses" (For more information: BABOK®).

746   Mission Thread Analysis – analysis of mission threads (For more information: Defense Acquisition
747   Guidebook, Chapter 13 – Program Protection Plan)

748   "Mission Thread – A sequence of end-to-end activities and events that take place to
749   accomplish the execution an SoS capability. The context of a mission thread is defined by a
750   vignette. A mission thread is given as a series of steps. There are three main types of mission
751   thread: operational, development, and sustainment. Chairman of the Joint Chiefs of Staff
752   6212.01F defines a Joint Mission Thread (JMT) as an operational and technical description
753   of the end-to-end set of activities and systems that accomplish the execution of a joint
754   mission [CJCSI 2012]." (For more information: CMU/SEI-2013-TR-003)

755   Observation – "a means to elicit requirements by conducting an assessment of the stakeholder's
756   work environment." (For more information: BABOK®).

757   Procedure Development – system of creating defined steps and tasks in order to complete a task
758   performed.

759   Process Analysis – See workflow analysis.

760   Process Flow Analysis – analysis of the process flow or workflow diagram.

761   Process Flow Diagram – Also called process flowchart. "A flowchart is a picture of the separate
762   steps of a process in sequential order. Elements that may be included are: sequence of actions,
763   materials or services entering or leaving the process (inputs and outputs), decisions that must be
764   made, people who become involved, time involved at each step and/or process measurements. (For
765   more information: American Society for Quality)

766 Project Plan – "a formal, approved document used to guide both project execution and project
767 control. The primary uses of the project plan are to document planning assumptions and decisions, to
768 facilitate communication among stakeholders, and to document approved scope, cost, and schedule
769 baselines. A project plan may be summary or detailed." (For more information: PMBOK®)

770 Project Planning – "development and maintenance of the project plan" (For more information:
771 PMBOK®).

772 Process Visualization – See Visualization.

773 Questionnaire – See Survey.

774 Requirements Definition – "often the main practice that serves as a bridge between project teams
775 and business stakeholders. The practice should define both product and project requirements as well
776 as related functional and non-functional requirements. Requirements definition should begin early in
777 the analysis phase" (For more information: U.S. Department of Health and Human Services,
778 Enterprise Performance Life Cycle Framework Practices Guide: Requirements Definition]

779 Responsible/Accountable/Consulted/Informed (RACI) – "describes the roles of those involved in"
780 activities. "It describes stakeholders as having one or more of the following responsibilities for a
781 given task or deliverable:

782 [R]esponsible - does the work,

783 [A]ccountable - is the decision maker (only one)

784 [C]onsulted - must be consulted prior to the work and gives input

785 [I]nformed - means that they must be notified of the outcome" (For more information:
786 BABOK®).

787 Review and Analysis of Process Description and/or Diagram – a way of collecting data (or
788 additional data) by reviewing and conducting further analysis on (e.g., existing process descriptions
789 or diagrams, listing of intersections and dependencies, relevant project plans, strategic plans,
790 implementation plans, or any documentation that can point to critical or limiting activities for this
791 program; process description and/or diagram against the listing of chokepoints and bottlenecks that
792 could degrade process's ability to fulfill the mission.)

793 Risk Analysis – the process of identifying the risks to system security and determining the likelihood
794 of occurrence, the resulting impact, and the additional safeguards that mitigate this impact. Part of
795 risk management and synonymous with risk assessment (For more information: NIST SP 800-27).

796 Root Cause Analysis – "a structured examination of an identified problem to understand the
797 underlying causes" (For more information: BABOK®).

798 Scenario – "an analysis model that describes a series of actions or tasks that respond to an event.
799 Each scenario is an instance of a use case" (For more information: BABOK®).

800   Scope Modeling – "used to define the scope of the analysis or the scope of the solution" (For more
801   information: BABOK®).

802   Sequence Diagrams – "type of diagram that shows objects participating in interactions and the
803   messages exchanged between them" (For more information: BABOK®).

804   Survey – "administers a set of written questions to stakeholders in order to collect responses from a
805   large group in a relatively short period of time." (For more information: BABOK®).

806   Systems Analysis – "the act, process, or profession of studying an activity (such as a procedure, a
807   business, or a physiological function) typically by mathematical means in order to define its goals or
808   purposes and to discover operations and procedures for accomplishing them most efficiently" (For
809   more information: Merriam-Webster).

810   Use Case – "an analysis model that describes the tasks that the system will perform for actors and
811   the goals that the system achieves for those actors along the way" (For more information:
812   BABOK®).

813   Visualization – "1: formation of mental visual images; or 2: the act or process of interpreting in
814   visual terms or of putting into visible form" (For more information: Merriam-Webster).

815   Workflow Analysis – "entails reviewing all processes in an organization with a view toward
816   identifying inefficiencies and recommending improvements" (For more information: Healthcare
817   Information and Management Systems Society (HIMSS)).

818

819

820

821

822 **Appendix D—Illustrative Example of Using Criticality Analysis Process Model**

823 **Exposition/Situation**

824 An organization is engaged in a large system integration effort. The system will process a variety of data
825 including critical mission data. It will consist of custom-built and COTS components.

826 The organization does not have a set of standardized security requirements for acquisition. Criticality levels
827 for systems and components have not been established. A FIPS 199 Impact Level for the system has been
828 established.

829 The Program Manager for the effort decides to use Criticality Analysis Process Model to prioritize different
830 subsystems, components, and subcomponents within the system to ensure appropriate levels of care are
831 applied to those subsystems, components, and subcomponents.

832 **Action**

833 The Program Manager starts with Process A as described in the Model. Since Criticality levels have not yet
834 been assigned, the Program Manager decides to use the Model in the order of the processes, as presented, A,
835 B, C, D, and E.

836 *Process A – Criticality Analysis Procedure Definition*

837 The Program Manager reviews the organization's strategic plan, risk management plan, and other
838 documentation and determines that the organization does not have any formal Criticality Analysis
839 Procedures. It is early enough in the process that a procedure can be written into the project plan for system
840 procurement and implementation.

841 **A.1** The Program Manager delegates to a business analyst[7] the process of integrating Criticality Analysis
842 activities into the project plan. The business analyst uses the Criticality Analysis Process Model in this
843 publication to determine appropriate tasks, roles, and responsibilities based on Processes B, C, D, and E of
844 the Model. The business analyst uses guidance in the *Related Processes Outside of Criticality Analysis* in
845 each Process and Sub-process description to ensure appropriate integration points with overall security
846 engineering and risk management processes have been appropriately integrated and planned.

847 *Process B - Conduct Program-Level Criticality Analysis*

848 The Program Manager checks to ensure that procedures for conducting a Criticality Analysis have been
849 defined and are appropriate to be used for this system integration effort. The Program Manager also checks
850 with the individual responsible for the program for which the system is being developed, to see if Criticality
851 Levels have been assigned to that program – they have not. The two Program Managers then work together
852 to execute Process B.

853 **B.1** To ensure that criticality levels appropriately reflect organizational and program priorities, the two

---

[7]In the context of this document, any role mentioned, such as "business analyst" is not a specific title but a role of a person engaged in the
program.

854 Program Managers collect documentation that contains organizational and program goals and objectives,
855 such as organizational strategic plan, mission and vision statement, shareholder report, as well as relevant
856 laws and regulations (e.g., FISMA, NERC CIP, FFIEC IT Handbook). They define several program goals
857 related to information security and safety that were not already documented. They work together to define
858 how much detail is needed for this analysis to be usable and define certain assumptions about the project,
859 including the expected budget and timeline.

860 **B.2** The Program Managers use the information from B.1 to develop a high-level description for how the
861 program goals and objectives are accomplished. They interview several personnel within the project for
862 which the system will be used on how they perform their duties and how they expect to use the system. The
863 Program Managers use the gathered information to develop a visual mapping of the processes. This
864 information will also be used in designing the system.

865 **B.3** The Program Managers delegates to a business analyst the task of identifying dependencies within the
866 processes. This person conducts brainstorming discussions with relevant program stakeholders asking the
867 following questions (among others):

868 • Which processes store information or data from another process?
869 • Which processes must be completed before another process takes over?
870 • Which organizational unit is responsible for ensuring information or data is transferred between
871 processes?
872 • What is required to ensure the information or data is transferred in a timely manner?
873

874 As a result of this effort, certain workflow paths are identifiable and areas of concern can be highlighted.
875 The business analyst documents this work in the visual mapping created in B.2.

876 **B.4** The business analyst conducts interviews and brainstorming sessions with key stakeholders of each
877 process along with an individual responsible for security to define how the process operates under both
878 normal and abnormal conditions. The following questions (among others) are used to help the discussion:

879 • What will happen to the process if a key person takes an unexpected leave of absence?
880 • What will happen to the process if an input is not received on time and as expected?
881 • What will happen to the process if information is stolen?
882

883 The business analyst then broadens the discussion to define how the process impacts the overall project.
884 Using the mapping developed in B.3, the business analyst uses the following questions to help guide the
885 discussion:

886 • What other processes will be impacted if a processes output is compromised?
887 • How much of a delay in a process can occur before other processes are impacted?
888 • How will a delay or compromise in one process impact the overall goals of the organization
889 (including safety and security goals)?
890

891 **B.5** Next, the business analyst and program managers use this information to rank each of the processes

892   from most important to least important by how vital they are to the success of the objectives / goals defined
893   in B.1, and how strongly an adverse operating state will affect the program objectives and goals. They use
894   organization's contingency plan or a similar document to inform their discussion. They group those
895   rankings into High, Moderate, and Low, which becomes the Baseline Criticality of those activities and
896   associated workflow paths.

897   *Process C*

898   With the program-level Baseline Criticality Levels completed, the Program Manager focuses the analysis on
899   the system that is currently under development (in the design phase). This analysis is delegated to the
900   individual in charge of systems architecture (the lead systems architect). He or she reviews the artifacts
901   produced so far. This person participated in much of the work done in Process B and is therefore somewhat
902   familiar with it already.

903   **C.1** The lead systems architect starts from validating how the system being designed will support the
904   workflow paths that were assigned High Baseline Criticality Level in Process B. The lead systems architect
905   decides to make the analysis more detailed and scope the analysis to only those subsystems that support
906   activities and workflow paths that have been assigned High Baseline Criticality in Process B. The lead
907   systems architect collects and reviews the relevant artifacts of Process B, plus more detailed system
908   documentation that was not included in Process B, including backup plans, the FIPS 199 impact level for
909   the system, and relevant High Value Asset (HVA) designations.

910   **C.2** Using this and any other relevant information about the current system design and existing
911   infrastructure (e.g., system mission statement, stakeholder requirements, functional requirements, system
912   architecture and design, proposed network topology, etc.), the lead systems architect then works to identify
913   functionalities and capabilities needed to support highly critical processes defined in Process B. The lead
914   systems architect identifies those subsystems – both existing and under development - which will be used to
915   provide those functionalities and capabilities. For existing subsystems, the lead systems architect initially
916   identifies those functions, which directly support a highly critical process, and those functions, which do,
917   not, based on information from Process B. The lead systems architect documents the resulting dependencies
918   in a simple diagram.

919   **C.3** Next, the lead systems architect identifies dependencies between the subsystems. They ask questions
920   such as:

921   •   Which subsystem processes accept data inputs, process and/or store that data, and then present the
922   data when requested as an output?
923   •   What is required for providing the data in a timely manner?
924   •   Which subsystem processes have to be completed before the next process takes over? And where do
925   these processes intersect?
926

927   The lead systems architect documents the resulting dependencies in a process flow diagram.

928   **C.4** The lead systems architect uses this information to document how the subsystems will function when
929   operating normally and abnormally. The lead systems architect works with a security engineer to brainstorm
930   what might cause an abnormal operating condition. They consider situations such as:

931   • A network component fails due to incorrectly applied patch
932   • A vulnerability is discovered in a critical module that makes the system vulnerable to a data breach
933   • The system produces abnormal results including incorrect data
934   • A system administrator resigns due to a dispute with the supervisor.
935

936   The lead systems architect documents the results of this analysis and the brainstorm in a series of
937   descriptions of operating states.

938   **C.5** Next, the lead systems architect works with a group of relevant program stakeholders to rank the
939   subsystems from most important to least important. They ask the following questions (among others):

940   • What will happen to the functions/capabilities each subsystem is delivering when each of the
941   abnormal operating states occurs?
942   • What will be the impact to system and subsystem operations?
943   • Which of the systems and subsystems are most important for the functions and capabilities to be
944   continually delivered, even at a reduced state or slower speed?
945

946   The lead systems architect then decides on the thresholds for grouping the subsystems into High, Moderate,
947   and Low, which becomes the Baseline Criticality level of each subsystem.

948   **Process D**

949   Based on the Baseline Criticality Levels for the subsystems, the Program Manager narrows the scope to the
950   most critical subsystems. This analysis is delegated to the individual in charge of systems design and
951   engineering (the lead systems engineer). He or she reviews the artifacts produced so far. This person
952   participated in much of the work done in Process B and is therefore somewhat familiar with it already.

953   **D.1** The lead systems engineer starts with identifying those subsystems that were assigned High Baseline
954   Criticality Level in Process C. This includes a mix of COTS and custom subsystems, both existing and to-
955   be-developed. The analysis for each subsystem is delegated to a systems engineer. For each subsystem, the
956   responsible system engineer reviews the artifacts of Processes B and C and relevant laws and regulations
957   that may help provide additional information about any requirements for components and subcomponents.
958   For each subsystem, the responsible system engineer determines the level of detail that is necessary and
959   possible for the analysis.

960   **D.2** Each systems engineer then works to identify functionalities and capabilities needed within the
961   subsystem. The responsible system engineer requests any documentation available related to the design of
962   the subsystem, including a description of processes, functions, and components (e.g., bills of materials,
963   component manuals, or anything else that is available). For some of the COTS products, this information is
964   very limited. Using this information, each systems engineer creates a listing of the functionalities and
965   capabilities that will be performed by the subsystem. For existing subsystems, the responsible system
966   engineer identifies the components and subcomponents supporting their subsystem.

967   **D.3** The systems engineers identify dependencies within their respective subsystems. The following
968   questions could be useful:

969   • Which components accept data inputs, process and/or store that data, and then present the data when
970   requested as an output?
971   • What is required for the subsystem to function as expected?
972   • Which component must be operational or completed before another can begin?
973   • Who supplies the components?
974   • When are the components expected to fail?
975

976 For existing subsystems, this information is provided in architecture diagrams and similar documents. For
977 systems yet to be developed, the system engineers develop this information as part of their design process.

978 **D.4** For custom subsystems, the systems engineers, working with security engineers, document how each
979 component of the subsystem will function when operating normally and/or abnormally. The following
980 situations are considered (among others):

981   • Not enough power
982   • An overload of data
983   • Is given incorrect data
984   • Operates in extreme temperatures
985   • A microchip component fails or malfunctions
986   • A firmware update goes wrong
987   • Unexpected shortage of subcomponents
988

989 For COTS subsystems, the system engineers review the manufacturer-provided documentation to identify
990 how the subsystems will respond to these situations.

991 Systems engineers document results of each of their analyses in a series of descriptions of operating states.

992 **D.5** Next, each systems engineer works with a group of relevant stakeholders to assign Baseline Criticality
993 levels to components. The following questions may be useful:

994   • What will happen to the functions/capabilities delivered by the subsystem when components or
995   subcomponents fail, resulting in an adverse operating state?
996   • What will be the impact to subsystem operations?
997   • Which of the components are most important for the subsystem to continue operating, even at a
998   reduced state or slower speed?
999

1000 Each systems engineer ranks the components from most important to least important. They then decide on
1001 the thresholds grouping the components and subcomponents into High, Moderate, and Low groups, which
1002 becomes the Baseline Criticality level for each component.

1003 In a few cases, the systems engineers decided to do further analysis to identify critical subcomponents of
1004 highly critical components. They repeat Process D, focusing on these highly critical components and their
1005 associated subcomponents.

1006 **Process E**

1007 With Baseline Criticalities assigned across the program activities/workflow paths, subsystems of the system
1008 under development, and components/subcomponents, the lead systems architect, lead systems engineer, and
1009 lead security engineer begin to review baseline criticalities for consistency, interdependencies, and develop
1010 final subsystem and component/subcomponent criticality levels.

1011 **E.1** Those persons responsible for conducting processes B, C, and D, meet to review the artifacts and
1012 results from their respective Processes. They identify any components, which are very similar in
1013 functionality; they identify which components or component types support more than one subsystem; they
1014 identify groups of components supplied by the same manufacturer; they identify groups of components
1015 expected to fail around the same timeframe. They perform this same analysis for subsystems and the system
1016 functions they support. The tracing of components all the way through the program activities and workflows
1017 helps identify any interdependencies that have not been considered in the analysis up to this point. The
1018 group documents results of this analysis using diagrams of interdependencies.

1019 **E.2** The group then identifies what existing controls have been designated to monitor and protect the
1020 system, subsystems, and components. These controls include traditional security controls such as access
1021 control, configuration management, secure design principles, network and system activity monitoring
1022 functions, software switches, etc.; they may be automated, technical, or manual. These controls can be
1023 documented in a variety of places, including security requirements, security plans, risk treatment plans, etc.

1024 **E.3** The group then traces the impact of operating states that were defined in B.4, C.4, and D.4 to determine
1025 what adverse operating states may have a cascading effect across the subsystems, system, and project
1026 processes. The group reviews the controls in place at each level and what would happen to the program if
1027 the control(s) failed.

1028 **E.4** Next, the group reviews available security risk documentation to see components/ subcomponents, the
1029 subsystems, or the system itself should be assigned a higher criticality level than what has been assigned
1030 during the process thus far. If it is discovered that existing controls are not appropriate for the level of
1031 criticality assigned the component or subsystem, the group increases the criticality level of that component
1032 or subsystem. Using existing risk assessment, threat assessment, impact analysis, or any similar
1033 documentation the group evaluates the controls identified in E.2, the impact of operating states on those
1034 controls in E.3, and the Baseline Criticality Levels assigned in processes B, C, and D. The results of this
1035 analysis will help finalize Criticality Levels in E.5.

1036 **E.5** Finally, the group reviews analysis results from Process E to determine how Baseline Criticality Levels
1037 that were assigned to the system, its subsystems, and components/subcomponents should be revised to
1038 assign final Criticality Levels. This analysis takes into account identified interdependencies, controls, and
1039 any aspects of the system, subsystem, and component/subcomponent operations that may be vulnerable due
1040 to systems architecture and design, reliance on a single supplier, or any other factors that were discovered in
1041 the overall analysis.

1042 Once Criticality Levels have been finalized, the program manager distributes the results to the groups
1043 performing risk analysis, threat analysis, impact analysis, contingency planning, and systems engineering
1044 activities. Criticality Levels are then used to inform these activities and help refine how they are planned
1045 and performed in the future. Criticality Levels also provide valuable inputs into the design and refinement

1046   of security requirements and controls, help shape system and component (hardware and software) testing,
1047   determine if any components should be bought in advance and stockpiled, and to inform supplier
1048   diversification decisions.

1049   Later, Criticality Levels are used to inform future system development and integration efforts, as well as
1050   future procurements and modernization efforts.

1051 **Appendix E—Criticality Analysis Process Model**

1052 Detailed version of high level criticality process. Please note that this image is split into two parts for ease of printing. For a .pdf of the entire
1053 image, please see the Supplemental Content section of this publication.
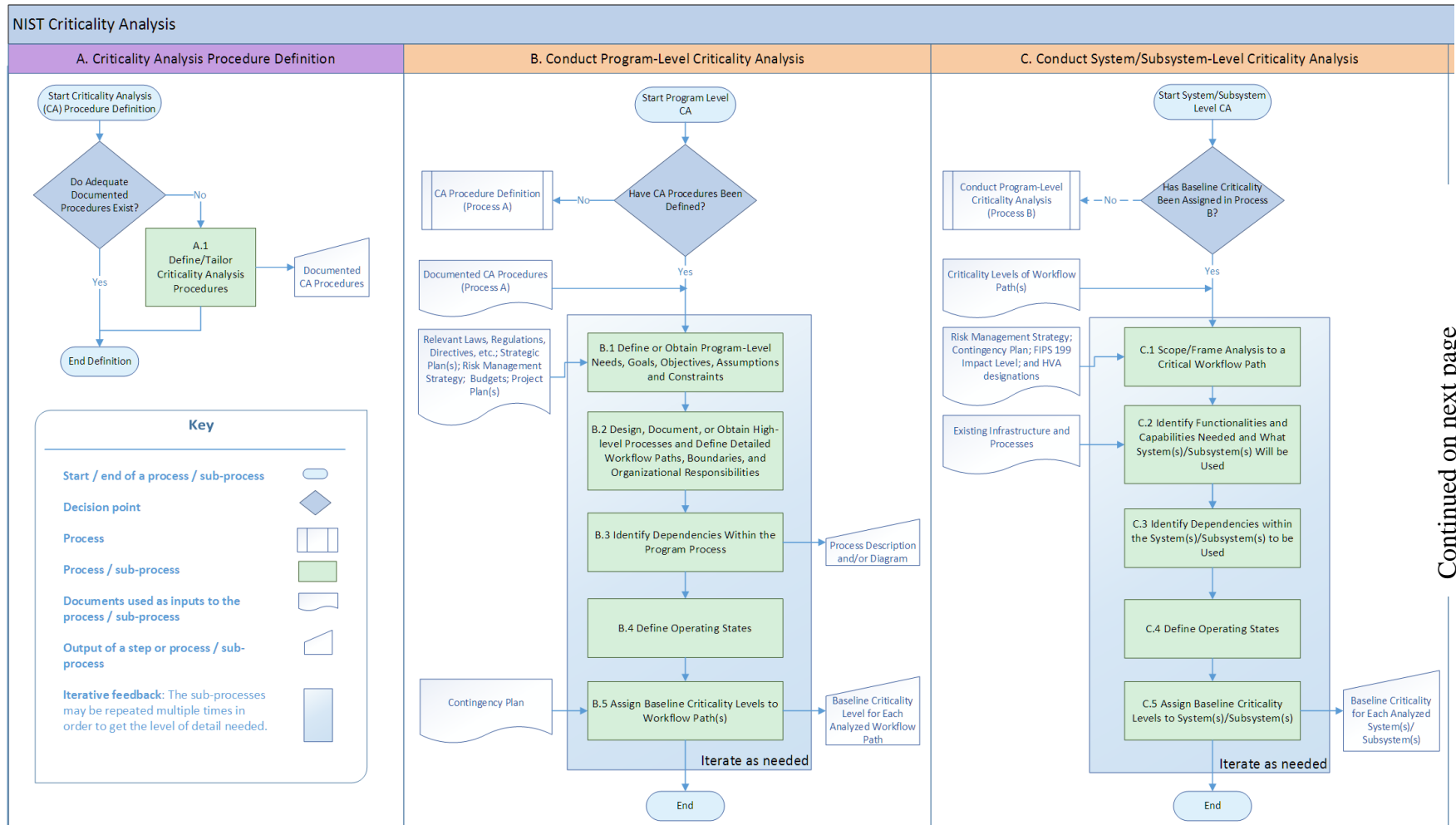
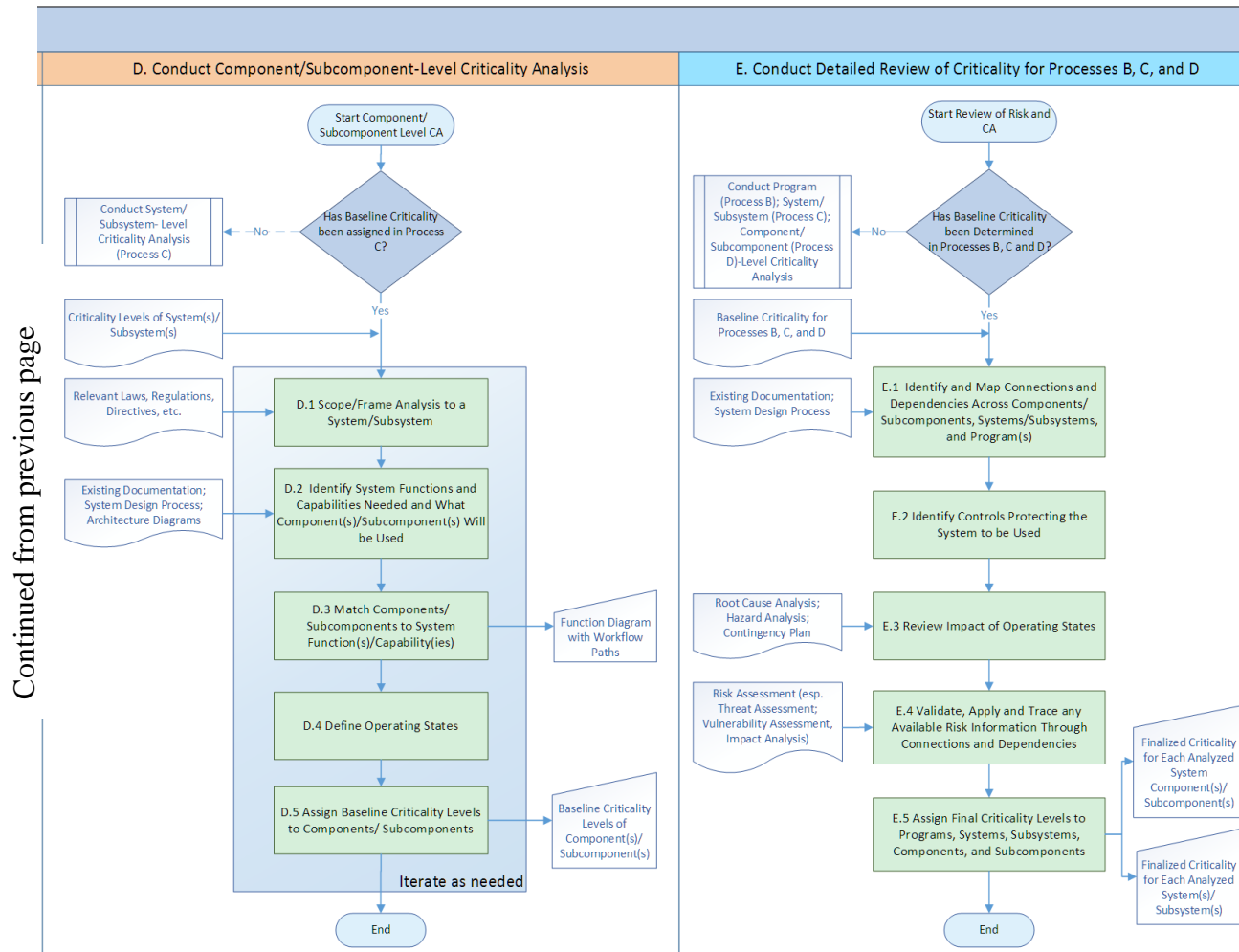1054

1055 **Figure 7 - NIST Criticality Analysis Process Model Part 1**

67

1056

1057

Figure 8 - NIST Criticality Analysis Process Model Part 2

1058

1059