Software Engineering Institute

CERT® Resilience Management Model, Version 1.2

Resilient Technical Solution Engineering (RTSE)

Richard A. Caralli
Julia H. Allen
David W. White
Lisa R. Young
Nader Mehravari
Pamela D. Curtis

February 2016

CERT Program

http://www.cert.org/resilience/

Carnegie Mellon

This material is based upon work funded and supported by various entities under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Various or the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0003234

RESILIENT TECHNICAL SOLUTION ENGINEERING

Engineering

Purpose

The purpose of Resilient Technical Solution Engineering is to ensure that software and systems are developed to satisfy their resilience requirements.

Introductory Notes

Software and systems are pervasive organizational assets that automate services and support business processes to help organizations meet their missions. The importance of resilient technical solutions—software and systems that resist threats, function satisfactorily in the face of adversity, and continue to help services meet their missions during times of stress—cannot be overstated.

Resilient software and systems do not become survivable and resistant to threat without an organizational commitment to address resilience throughout the development process. These assets must be specifically designed and developed with consideration of the types of threats they will face, the operating conditions and changing risk environment in which they will operate, and the priority and sustainment needs of the services they support. Typical software and system development life cycles understandably focus on identifying and satisfying functional requirements; that is, most of the effort goes into defining and designing what the software or system must do to fulfill its use case, purpose, objectives, and ultimately its mission. However, requirements for quality attributes such as security, availability, performance, reliability, and the ability to sustain software and system assets can in the long run be equally important to the usability and longevity of software and system assets and require considerable resources to address in the operations phase if they are not considered early in the development life cycle.

Unfortunately, quality attribute requirements can be harder to define, design, and implement and in many cases require significant business impact and cost analysis up front to ensure that they are worth investing in. This leads to a tendency to ignore these requirements early in the development life cycle and to bolt on solutions to address them later in the design and implementation phases, when they are more costly, less effective, and typically harder to manage and sustain in an operational mode. The failure to consider requirements for quality attributes is a primary reason why software and systems in operation are subject to high levels of operational risk resulting from failed technology and processes. This expands an already complex operational risk environment brought about by the integration of software and systems with other technology assets such as information, hardware, networks, and telecommunications. In essence, ignoring quality attributes creates additional security, continuity, and other related operational risks that must be managed in the operations phase of the life cycle, typically at higher cost, lower efficacy, and potentially increased consequences to the organization. In some cases, these problems may be so significant as to shorten the expected life of the software and systems, diminish their overall operational resilience, and result in cumulatively lower than expected return on investment.

The functional aspects of software and systems do not have meaning if they are not resistant to disruption or cannot be sustained under degraded conditions. High-quality software and systems cannot be produced and sustained without addressing these issues early in the development life cycle. The controls necessary to demonstrate that integrity and availability requirements are met must be identified as early as the needs determination phase. Controls can then be designed to fit the architecture and functionality of the software and systems in their expected operating environment and can be implemented and made operable to ensure that they achieve the desired effect. This process cannot be shortchanged; it must be wholly integrated into the organization's development process and must be measured, managed, and improved in the same manner as highly effective and mature software and system development processes.

Developing or acquiring resilient technical solutions such as software and systems requires a dedicated process that encompasses the asset's life cycle. The process begins with establishing a plan for addressing resilience as part of the organization's regular development life cycle and the integration of the plan into the organization's corresponding development process. The identification, development, and validation of quality attribute requirements are performed alongside similar processes for functional requirements. Resilient software and systems are designed through the elicitation and identification of resilience requirements and the design of architectures that reflect a resilience focus, including security, operations controls, and the ability to sustain software and system assets. Resilient software and systems are developed through processes that include secure coding of software, software defect detection and removal, and the development of resilience controls based on design specifications. The resilience controls for software and systems are tested, and issues are referred back to the design and development cycle for resolution. Reviews are conducted throughout the development life cycle to ensure that resilience is kept in the forefront and given adequate attention and consideration. System-specific continuity planning is performed and integrated with service continuity planning to ensure that software, systems, hardware, networks, telecommunications, and other technical assets can be sustained. A post-implementation review of deployed systems is performed to ensure that resilience requirements are being satisfied as intended.

In operation, software and systems are monitored to determine if there is variability that could indicate the effects of threats or vulnerabilities and to ensure that controls are functioning properly. Configuration management and change control processes are implemented to ensure software and systems are kept up-to-date to address newly discovered vulnerabilities and weaknesses (particularly in vendor-acquired products and components) and to prevent the intentional or inadvertent introduction of malicious code or other exploitable vulnerabilities.

To effectively integrate resilience considerations, the organization must establish guidelines for developing resilient software and systems, develop a plan for selecting, tailoring, and integrating selected guidelines into existing development life cycles and processes for any given development project, and then execute the plan. Plan development and execution include identifying and mitigating risks to the success of the development project.

The Resilient Technical Solution Engineering process area is strongly influenced by two Capability Maturity Model Integration (CMMI) process areas [CMMI Product Team 2006]:

- Requirements Development, the purpose of which is to produce and analyze customer requirements and software and system product and product component requirements

- Technical Solution, the purpose of which is to design, develop, and implement solutions to software and system requirements (Solutions, designs, and implementations encompass software and system products, product components, and product-related life-cycle processes, either singly or in combination as appropriate.)

There are a growing number of reputable sources to consider when identifying and selecting candidate guidelines for the development of resilient software and systems across the life cycle, particularly for software security and assurance.

These are examples of sources of guidelines:

- Building Security In Maturity Model (BSIMM2) v6, www.bsimm.com/
- Open Web Applications Security Project (OWASP) Software Assurance Maturity Model (SAMM) v1.0, www.owasp.org/index.php/Category:Software_Assurance_Maturity_Model
- Microsoft's Security Development Lifecycle, Version 4.1, www.microsoft.com/security/sdl/
- Department of Homeland Security Assurance for CMMI Process Reference Model, https://buildsecurityin.us-cert.gov/swa/procwg.html

The Resilient Technical Solution Engineering process area assumes that the organization has one or more existing, defined processes for software and system development into which resilience controls and activities can be integrated. If this is not the case, the organization should not attempt to implement the goals and practices identified in this process area.

Note: This process area does not address the unique aspects of the resilience of embedded systems or the resilience of hardware that is part of a software-intensive system.

Related Process Areas

Resilience requirements for software and system technology assets in operation, including those that may influence quality attribute requirements in the development process, are developed and managed in the Resilience Requirements Development and Resilience Requirements Management process areas respectively.

Identifying and adding newly developed and acquired software and system assets to the organization's asset inventory are addressed in the Asset Definition and Management process area.

The management of resilience for technology assets as a whole, particularly for deployed, operational assets, is addressed in the Technology Management process area. This includes, for example, asset fail-over, backup, recovery, and restoration.

Acquiring software and systems from external entities and ensuring that such assets meet their resilience requirements throughout the asset life cycle are addressed in the External Dependencies Management process area. That said, RTSE-specific goals and practices should be used to aid in evaluating and selecting external entities that are developing software and systems (EXD:SG3.SP3), formalizing relationships with such external entities (EXD:SG3.SP4), and managing an external entity's performance when developing software and systems (EXD:SG4).

Monitoring for events, incidents, and vulnerabilities that may affect software and systems in operation is addressed in the Monitoring process area.

Service continuity plans are identified and created in the Service Continuity process area. These plans may be inclusive of software and systems that support the services for which planning is performed.

Summary of Specific Goals and Practices

Goals	**Practices**
RTSE:SG1 Establish Guidelines for Resilient Technical Solution Development	RTSE:SG1.SP1 Identify General Guidelines
	RTSE:SG1.SP2 Identify Requirements Guidelines
	RTSE:SG1.SP3 Identify Architecture and Design Guidelines
	RTSE:SG1.SP4 Identify Implementation Guidelines
	RTSE:SG1.SP5 Identify Assembly and Integration Guidelines
RTSE:SG2 Develop Resilient Technical Solution Development Plans	RTSE:SG2.SP1 Select and Tailor Guidelines
	RTSE:SG2.SP2 Integrate Selected Guidelines with a Defined Software and System Development Process
RTSE:SG3 Execute the Plan	RTSE:SG3.SP1 Monitor Execution of the Development Plan
	RTSE:SG3.SP2 Release Resilient Technical Solutions into Production

Specific Practices by Goal

RTSE:SG1 Establish Guidelines for Resilient Technical Solution Development

Guidelines are developed to ensure proper consideration of resilience activities and controls in all phases of the life cycle.

Resilient technical solution development requires the integration of security and business continuity considerations into the organization's software and system development life cycle, methodologies, and processes. During all phases of development, resilience requirements must be considered and correspondingly translated into requirements, design, and implementation actions.

Integrating resilience into the life cycle cannot be performed as a bolt-on activity; resilience development activities must be fully incorporated into the development process to ensure that

- resilience requirements that are relevant for the software or system are identified *(Refer also to the Resilience Requirements Development and the Resilience Requirements Management process areas.*)
- resilience requirements are analyzed and planned for in the development life cycle
- software and system designs and architectures include appropriate levels of controls and satisfy resilience requirements
- resilience requirements are addressed early in and throughout the life cycle and maintained over the useful life of the asset
- to the extent possible and practical, operational software and systems do not contain vulnerabilities and weaknesses that arise from poor or inadequate consideration of resilience in prior life-cycle phases
- software and systems can be effectively and efficiently maintained through monitoring, change control, and configuration management

Integration of resilience requirements into the life cycle may require the expansion of life-cycle phases and activities. For example, consideration of protective controls during development requires expanded requirements definition and analysis activities and the development of assurance cases to provide evidence that resilience requirements are met. In some cases, additional activities have to be added to phases such as the design of specific security architectures, the use of misuse/abuse cases and attack patterns to anticipate risks to continuity, and secure coding practices to reduce known vulnerabilities. Incorporation of these additional considerations must be part of the organization’s software and system development plan; otherwise, there is risk that the additional considerations may be viewed as optional and, as a result, be underfunded or underresourced.

Guidelines are a means for documenting the organization’s requirements for considering and institutionalizing resilience in development projects. Guidelines are an important tool for enforcing (and reinforcing) a resilience viewpoint in development projects because they provide a consistent foundation. This is particularly important because organizations often

- execute many different development projects with different development teams simultaneously
- deploy more than one life-cycle methodology, some of which may be proprietary and depend on the involvement of external entities
- assemble software and systems from existing or legacy assets, open-source assets, or commercial off-the-shelf (COTS) assets
- outsource development projects

Resilience guidelines ensure consistency across projects (and external entities) and provide for expected outcomes regardless of the project under development, methodology deployed, or entities involved.

Guidelines for the acquisition of resilient software and systems, comparable to those used for development performed within the organization, may be established in this process area. *(Their use in evaluating and selecting external entities, in developing contractual agreements with external entities, and in managing ongoing relationships is addressed in the External Dependencies Management process area.)*

RTSE:SG1.SP1 Identify General Guidelines

General guidelines for building resilience into software and systems are identified.

General guidelines apply to all life-cycle phases. Practices and controls that implement general guidelines will differ by life-cycle phase, becoming more detailed and precise as development progresses.

General guidelines include such topics as understanding the operational production environment within which the software and system will be deployed, performing trade-off analyses that balance resilience needs and requirements against costs and benefits, identifying and analyzing resilience project risks throughout the life cycle, addressing issues relevant to continuity of operations for the service or services that the software and systems are intended to support, conducting threat analysis, and collecting and reporting appropriate measures of progress and satisfaction of life-cycle phase exit criteria, particularly with respect to resilience.

Guidelines that address software and system interoperability, including establishing standards, developing an interoperability management strategy, and analyzing risks related to the interoperability of software and systems, are addressed in the Technology Management process area, specifically TM:SG5.SP4.

Typical work products

1. General guidelines for resilient software and systems

Subpractices

1. Identify general guidelines for the development of resilient software and systems.

   Guidelines are project-specific but should address topics such as the following:

   - project management, including
     - defining project objectives for resilience
     - defining the scope of resilience for the software or system (levels of required resilience based on risk thresholds)
     - understanding the operating environment and defining the operating constraints for resilience for the environments in which software and systems will be deployed
     - identifying operational concepts and associated scenarios for resilience
     - balancing resilience needs against costs and benefits
     - defining criteria (with respect to resilience) for approval to proceed from one project life-cycle phase to the next
   - risk management, including
     - identifying and analyzing resilience project risks *(Refer to the Risk Management process area.)*
     - identifying and analyzing resilience software and system risks during all life-cycle phases
   - threat analysis, including modeling, assessment, attack models and patterns, and misuse/abuse cases
   - interconnectivity and interoperability (Refer to TM:SG5.SP4.)
   - control identification and prioritization, including
     - controls for protecting and sustaining the service or services that the software and systems are intended to support
     - controls for protecting and sustaining the software and systems
     - software supply chain resilience controls, such as chain of custody, least privilege access, separation of duties, tamper resistance (such as code signing) and evidence of tampering, persistent protection of high-value

information, compliance management, and code inspection, testing, and verification *(Refer to the Controls Management process area.)*

- quality assurance, including methods for validating and verifying desired or attained levels of software and system resilience, sometimes referred to as “assurance cases”
- measurement (Refer to the Measurement and Analysis process area.)
- reviews and documentation necessary to demonstrate successful completion of each life-cycle phase
- training for software engineers and project managers (Refer to the Organizational Training and Awareness process area.)

RTSE:SG1.SP2 Identify Requirements Guidelines

Guidelines for determining software and systems resilience requirements are identified.

Requirements are the basis for architecture and design. Resilience requirements must be defined early in the development life cycle so that the resulting software or system can be evaluated in terms of its ability to support service continuity and other operational resilience requirements.

Resilience requirements derive from analyzing the needs of service owners for each software or system associated with a high-value service, relevant stakeholders, the operational environment, and information security risk assessments and/or business impact analyses. Requirements need to reflect confidentiality, integrity, and availability requirements as well as those for accountability, non-repudiation, correctness, predictability, and reliability.

Resilience requirements are identified and refined throughout the phases of the software and system life cycle. Design decisions, subsequent corrective actions, and feedback during each phase of the life cycle are analyzed for impact on derived and allocated requirements.

During requirements engineering, an important perspective is that of the attacker. An attacker typically seeks defects and other conditions outside of normal operations that will allow for a successful intrusion. Thus it is important for requirements engineers to think about the attacker’s objectives and not just the software or system functional requirements.

Constructing threat models and operational usage scenarios for high-value services and assets can be useful in demonstrating satisfaction of resilience requirements. Scenarios and misuse/abuse cases can also aid in validating that the software or system responds correctly during times of stress and disruption.

Typical work products

1. Requirements guidelines for resilient software and systems

Subpractices

1. Identify requirements guidelines for the development of resilient software and systems.

   Guidelines are project-specific but should address topics such as the following:

   - resilience requirements elicitation (from service owners, stakeholders, and other entities dependent upon the software and system) (Resilience requirements for software and systems in operation are defined and managed in the Resilience Requirements Management process area and the Resilience Requirements Definition process area respectively. Requirements resulting from these two process areas that are relevant for developed and acquired software and system assets also have to be considered.)
   - risk analysis during requirements engineering (Risk analysis results inform requirements prioritization.)
   - threat analysis during requirements engineering
   - requirements trade-off analyses (service owner needs, stakeholder needs, operational environment considerations, etc.)
   - assumptions, decisions, and rationales
   - methods for representing defender and attacker perspectives (such as misuse/abuse cases and scenarios)
   - access control (Refer to the Access Management process area.)
   - identity management (Refer to the Identity Management process area.)
   - data security, including the use of encryption and credentials management (Refer to the Knowledge and Information Management process area.)
   - control identification and prioritization during requirements engineering (Refer to the Controls Management process area.)
   - analysis of any open-source, COTS, and legacy software that will be part of the system, including specification of resilience requirements to be met by such software
   - requirements specification reviews, including means for validating desired or attained levels of software and system resilience
   - quality assurance during requirements engineering
   - monitoring and audit (for example, of system logs, for intrusion prevention and detection) during requirements engineering (Refer to the Monitoring process area.)
   - measurement during requirements engineering (Refer to the Measurement and Analysis process area.)
   - training for software requirements engineers (Refer to the Organizational Training and Awareness process area.)

RTSE:SG1.SP3 Identify Architecture and Design Guidelines

Guidelines for designing resilience into software and systems are identified.

Architecture and design may be the most important phases for the development of resilient software and systems. Effective architecture and design choices not only will produce a structure that is more resilient and resistant to disruption but will also help prescribe and guide better decision making and guideline selection during implementation and assembly and integration. Poor, ill-informed decisions made during architecture and design can lead to design flaws that can never be overcome in later development phases or in operations.

A resilient architecture and its supporting design satisfy specified resilience requirements, reflect the defined operational production environment, and anticipate how best to adapt to changing conditions. A resilient architecture accounts for issues of interconnectivity, interoperability, service continuity, scale, and complexity. The design contains minimal to no detectable weaknesses that could be exploited when translated into implemented software and systems. A resilient architecture ensures that high-value services are operational during times of stress and the software and systems that support them are able to recover and return to operation in a reasonable period of time after a disruptive event.

Architecture and design guidelines for developing resilient software and systems cover design concepts, architecture, component design, detailed design, and design review and assessment.

Typical work products

1. Architecture and design guidelines for resilient software and systems

Subpractices

1. Identify architecture and design guidelines for the development of resilient software and systems.

   Guidelines are project-specific but should address topics such as the following:

   - risk analysis during architecture and design
   - threat analysis during architecture and design
   - design assumptions, decisions, and rationales
   - methods for representing defender and attacker perspectives (such as use scenarios)
   - attack surface
   - secure design patterns at the architecture and design level
   - access control (Refer to the Access Management process area.)
   - identity management (Refer to the Identity Management process area.)
   - data security, including the use of encryption and credentials management

- control identification and prioritization during architecture and design (Refer to the Controls Management process area.)
- analysis of open-source, COTS, and legacy software, including verification of required functional and resilience behavior and absence of malicious content
- service-oriented architectures, virtualization, and cloud computing (software as a service)
- integration with existing architectures (interconnectivity and interoperability)
- analysis for system complexity and scale, including end-to-end business process and service vulnerability analysis and failure analysis
- inspections and architectural and design reviews, including validating desired or attained levels of software and system resilience
- quality assurance during architecture and design
- monitoring and audit (for example, of system logs, for intrusion prevention and detection) during architecture and design (Refer to the Monitoring process area.)
- measurement during architecture and design (Refer to the Measurement and Analysis process area.)
- training for software architects and designers (Refer to the Organizational Training and Awareness process area.)

RTSE:SG1.SP4 Identify Implementation Guidelines

Guidelines for implementing resilient software and systems are identified.

Implementation includes the life-cycle phases of software coding and software and system testing. The purpose of implementation is to ensure that all resilience requirements are met, as reflected in the software and system architecture and design. Resilient software and systems are predictable in execution during both normal operations and times of stress. They are as free from exploitable vulnerabilities as possible.

Coding guidelines for resilient software include, for example, the use of secure coding standards and static and dynamic code analysis tools to ensure that standards are met and that identifiable software vulnerabilities have been eliminated. Standards may address, for example, input and output validation, exception handling, and the use of logging and tracing for debugging and diagnosis.

Testing guidelines for resilient software and systems include a wide range of testing techniques such as white box, black box, fuzz, and penetration testing, all designed to demonstrate the satisfaction of resilience requirements. In addition, software and systems are tested to produce evidence that attained levels of software and system resilience are as expected (sometimes referred to as “assurance cases”). Testing guidelines include the identification of approaches and cases that will be used during final inspection and when software needs to undergo regression testing as the result of a change that is made after the software is released into production, such as a patch to address a software vulnerability.

Typical work products

1. Coding guidelines for resilient software
2. Testing guidelines for resilient software
3. Testing guidelines for resilient systems

Subpractices

1. Identify coding guidelines for the development of resilient software.

   Guidelines are project-specific but should address topics such as the following:

   - risk analysis during coding
   - threat analysis during coding
   - attack surface evaluation and mitigation
   - secure design patterns at the implementation level
   - secure coding standards (language-specific)
   - code checklists, reviews, inspections, and static and dynamic code analysis, including tools to support these, which can be used to verify
     - that resilience requirements are satisfied
     - that architecture and design guidelines were followed
     - the absence of banned functions
     - the absence of commonly known vulnerabilities
     - that desired or attained levels of software resilience are present
   - conducting more in-depth reviews for the highest-risk, highest-value code
   - control identification and prioritization during coding (Refer to the Controls Management process area.)
   - quality assurance during coding
   - monitoring and audit during coding
   - measurement during coding
   - training for software developers

2. Identify testing guidelines for the development of resilient software.

   Guidelines are project-specific but should address topics such as the following:

   - risk analysis during software testing
   - threat analysis during software testing
   - attack surface reevaluation and mitigation
   - at the software level, methods for
     - resilience requirements functional testing
     - unit testing (commonly referred to as “white box testing”), including code coverage analysis

  - black box testing that focuses on the software’s externally visible behavior
  - fuzz testing
  - penetration testing, including assessment by external entities (*Refer to the External Dependencies Management process area.*)
  - testing for specific vulnerabilities as well as vulnerability regression testing
  - application of threat and attack models
  - testing open-source, COTS, and legacy software, including verification of required functional and resilience behavior and absence of malicious content
  - inspection testing in support of approval to release
  - regression testing
- automation of software test methods and tools to support automation
- software testing reviews, which can be used to verify
  - that resilience requirements are satisfied
  - that architecture and design guidelines were followed
  - the absence of banned functions
  - the absence of commonly known vulnerabilities
  - that desired or attained levels of software resilience are present
- code integrity and handling (including strong configuration management, verifiable chain of custody, anti-tampering, monitoring and analysis of event and audit logs, and code signing)
- conducting more in-depth testing for the highest-risk, highest-value software
- demonstrating compliance with interoperability standards (Refer to TM:SG5.SP4.)
- testing controls during software testing (Refer to the Controls Management process area.)
- quality assurance during software testing, including criteria for releasing tested software into production
- monitoring and audit during software testing
- measurement during software testing
- training for software test engineers

3. Identify testing guidelines for the development of resilient systems.

Guidelines are project-specific but should address topics such as the following:

- risk analysis during system testing
- threat analysis during system testing
- attack surface reevaluation and mitigation
- at the system level, methods for
  - resilience requirements functional testing

  - black box testing that focuses on the system's externally visible behavior
  - fuzz testing
  - penetration testing
  - testing for specific vulnerabilities as well as vulnerability regression testing
  - application of threat and attack models
  - integration testing
  - testing open-source, COTS, and legacy software in a system environment, including verification of required functional and resilience behavior
  - testing for system complexity and scale, including end-to-end business process and service vulnerability analysis and failure analysis
  - inspection testing in support of approval to release
  - regression testing
- automation of system test methods and tools to support automation
- system testing reviews, which can be used to verify
  - that resilience requirements are satisfied
  - that architecture and design guidelines were followed
  - that desired or attained levels of system resilience are present
- conducting more in-depth testing in a system environment for the highest-risk, highest-value business processes, services, and software
- demonstrating compliance with interoperability standards (Refer to TM:SG5.SP4.)
- testing controls during system testing (Refer to the Controls Management process area.)
- quality assurance during system testing, including criteria for releasing a tested system into production
- monitoring and audit during system testing
- measurement during system testing
- training for system test engineers

RTSE:SG1.SP5 Identify Assembly and Integration Guidelines

Guidelines for the assembly and integration of resilient software into resilient systems are identified.

During assembly and integration, the logical design assumptions for software and systems meet the physical, business, technical, organizational, and individual user realities of the operational production environment. Vulnerabilities (and their exploitation) can increase significantly based on assembly-integration design errors, architectural mismatches among software assets, insecure identity management and services, false assumptions about an asset's properties, an overreliance on perimeter-based network security mechanisms, and the use of assets in environments not envisioned by the assets' designers.

Business pressures for increased efficiency and flexibility are moving applications toward "just-in-time" service creation and delivery (for example, through dynamic assembly in a web services environment) and are therefore stressing the limits of resilience even further. User privacy concerns regarding the use of their identifiable information for tracking and tracing may create constraints and conflict with resilience goals. During assembly and integration, the potential system-wide effects of the emergent behavior of large numbers of software components and services have to be addressed in the operational production environment.

The assembly and integration of software and system assets to the end objective of ensuring resilience are not robust, well-understood disciplines, so they are subject to organizational interpretation and tailoring.

Typical work products

1. Assembly and integration guidelines for resilient systems

Subpractices

1. Identify assembly and integration guidelines for resilient systems.

   Guidelines are project-specific but should address topics such as the following:

   - for legacy software, COTS, and open-source software, analysis of existing software and system artifacts such as requirements specifications, architectures, designs, threat environment, code, test results, monitoring results, and vulnerabilities
   - use of analysis technologies such as reverse engineering, function abstraction, correctness verification, flow analysis, statistical analysis, test design and evaluation, and assurance auditing
   - analyzing interfaces with untrusted systems for vulnerabilities
   - end-to-end analysis of cross-software, cross-system work flows that support high-value business processes and services
   - containment and recovery from failures (failure analysis) in the context of service continuity (Refer to the Service Continuity process area.)
   - demonstrating compliance with interoperability standards (Refer to TM:SG5.SP4.)
   - testing of controls during assembly and integration (Refer to the Controls Management process area.)
   - at the assembled system level, methods for
     - resilience requirements functional testing
     - black box testing that focuses on the system's externally visible behavior
     - fuzz testing
     - penetration testing
     - testing for specific vulnerabilities, as well as vulnerability regression testing

  - application of threat and attack models
  - integration testing
  - testing open-source, COTS, and legacy software in a system environment, including verification of required functional and resilience behavior
  - testing for system complexity and scale, including end-to-end business process and service vulnerability analysis and failure analysis
  - inspection testing in support of approval to release
  - regression testing
- quality assurance, including criteria for releasing an assembled system into production
- measurement during system assembly and integration
- training for assembly and integration engineers

RTSE:SG2 Develop Resilient Technical Solution Development Plans

Plans for addressing resilience in the development life cycle are based on documented guidelines.

Planning for the incorporation of resilience into the development life cycle ensures that resilience activities and controls are included as a required part of the production of software and systems.

Planning involves first identifying which software and system technology assets warrant the integration of resilience activities into their development life cycles, and at what level. The plan describes the selection and incorporation of appropriate guidelines for addressing resilience in the life-cycle phases. These guidelines ensure consistency of resilience activities across projects and when using different life-cycle methodologies. The plan details how resilience activities will be incorporated and how they will be monitored and measured to ensure resilience requirements are appropriately considered in preliminary and detailed design, implementation, and assembly and integration. The plan also calls for interim reviews of resilience activities at all key milestones and decision points of the development life cycle.

The plan should be communicated to all staff involved in the development life cycle so that there is broad awareness of the organization's mandate to address resilience as a project, software, and system requirement. This should extend to external entities as well, particularly when projects have been outsourced or when proprietary life-cycle methodologies are being used by projects. (*Refer to the External Dependencies Management process area.)*

The identification, definition, management, and control of technology assets are addressed in the Asset Definition and Management process area. This includes the inventory of high-value technology assets, those that are essential to the successful operation of organizational services.

The prioritization of technology assets relative to their importance in supporting the delivery of key services is addressed in the Technology Management process area, specifically TM:SG1.SP1, Prioritize Technology Assets. While the Technology Management process assumes that technology assets already exist via, for example, *the organization's asset inventory, TM:SG1.SP1 can be effectively used to prioritize software and system assets that are yet to be developed or require significant upgrades or assets that are to be acquired.*

The management of technology asset risk and the maintenance of operational technology assets via monitoring, configuration management, and change control are described in the Technology Management process area.

RTSE:SG2.SP1 Select and Tailor Guidelines

Guidelines are determined for a specific software or system development project using selection criteria.

Organizations need to have well-established, business-driven criteria to determine which guidelines to incorporate into the development life cycle for a specific software or system technology asset.

Determining which guidelines to incorporate is based, in large part, on the relative value of the asset and its resilience requirements *(as defined in the Resilience Requirements Development process area)*.

Once criteria are established, they are used to select and tailor resilience guidelines for each life-cycle phase for a defined software or system development project *(refer to RTSE:SG1)*.

Typical work products

1. Selection criteria
2. Selected requirements guidelines
3. Selected architecture and design guidelines
4. Selected implementation guidelines
5. Selected assembly guidelines

Subpractices

1. Identify selection criteria for resilience guidelines.

Selection criteria may include the following:

- the value of services that the software or system is intended to support
- the relative value of the software or system to services that it is intended to support
- the extent to which the software or system addresses actions called for in service risk mitigation plans (along with the corresponding risk impacts and valuations)
- the priority of resilience objectives and requirements that must be satisfied by the software or system
- cost/benefit trade-off analyses, such as the relative importance of identifying software flaws early in the software development life cycle versus the cost to implement the guidelines
- make versus buy trade-off analyses
- the availability of adequately trained staff
- staff training costs

2. Select and tailor guidelines for a specific software or system asset.

   Prioritize guideline selection criteria based on discussion with key stakeholders, such as service owners. Determine which guidelines to include in the software or system development plan by applying the most important selection criteria for a specific software or system development project. Tailor guidelines as appropriate for the project.

RTSE:SG2.SP2 Integrate Selected Guidelines with a Defined Software and System Development Process

Selected resilience guidelines are integrated with a defined software and system development process and a documented plan.

Many organizations use documented approaches such as process models to define, manage, and improve software and system development processes that may even extend beyond life-cycle methodologies. Such approaches help ensure high-quality products that satisfy their requirements. However, these approaches do not typically address or incorporate resilience considerations. As a result, resilience as a property or quality of software and systems is absent.

Methods and models for improving software and system development processes must be extended to include resilience as a foundational element in process definition and as an expected attribute of software and systems that are produced through this process. Failure to include resilience as part of the development process may result in software and system assets that are unable to resist, tolerate, and recover from adverse or disruptive events. Such failures will likely affect the ability of key services and business processes to fulfill their mission.

Similarly, the plan for a specific software or system development project must be enhanced and updated to reflect resilience requirements and guidelines in the following areas:

- development process definitions
- tasks, progress measures, milestones, deliverables, and the assignment of resources (staff, funding, capital equipment, etc.) to implement resilience guidelines
- new risks introduced by resilience guidelines and the elevation of currently identified risks to a higher priority
- stakeholder involvement
- commitment to the updated plan
- decision criteria and authority at key project milestones

This specific practice assumes that

- defined development processes exist and are in use for any software or system that is required to meet resilience requirements

- documented plans exist and are used to develop any software or system that is required to meet resilience requirements

Typical work products

1. Updated development process definitions
2. Updated development plan

Subpractices

1. Update process definitions.

   The defined development process that is being used as the basis for a specific software or system development project is updated to reflect selected resilience guidelines *(refer to RTSE:SG1)*.

   Process definitions are periodically reviewed and updated to reflect new requirements, new understanding, and new guidelines throughout the development process.

2. Update the development plan.

   The documented plan that is being used to conduct a specific software or system development project is updated to reflect tasks, progress measures, milestones, deliverables, and the assignment of resources necessary to implement selected resilience guidelines *(refer to RTSE:SG1)*. In addition, updates to the plan should reflect any new risks introduced by the integration of resilience guidelines, as well as stakeholder involvement and necessary commitments to execute the updated plan.

   Development plans are periodically reviewed and updated throughout the development process.

RTSE:SG3 Execute the Plan

Progress against the plan for developing resilient software and systems is monitored throughout the development life cycle.

Progress against the development plan is monitored on an ongoing basis by the development team, and status against the plan is periodically measured and reported to key stakeholders at project milestones identified in the development plan. The purpose of monitoring, measurement, and review is to ensure that software and systems satisfy their resilience requirements.

Upon successful completion of the development plan, software and systems are formally reviewed to ensure they have met specified resilience requirements. The result of a successful formal review is organizational approval to release the asset into production.

RTSE:SG3.SP1 Monitor Execution of the Development Plan

Execution of the development plan is monitored to ensure that software and system resilience requirements are satisfied.

The organization uses the development plan as the basis and criteria for monitoring project performance in satisfying resilience requirements. Any deviations from the plan with respect to resilience must be analyzed to understand the potential impact on the project, the software, the system, and the organization.

The monitoring process must include requirements, design, implementation, and assembly and integration reviews at identified milestones to ensure that software and systems satisfy all stated resilience requirements at the level appropriate to the life-cycle phase. In the case where requirements cannot be satisfactorily demonstrated, development plans must be renegotiated with owners and stakeholders and updated. Any new and residual risks must be identified and managed (*refer to the Risk Management process area*).

To ensure that monitoring progress against the plan is performed on a timely and consistent basis, the organization should establish procedures that specify the frequency, protocol, and responsibility for monitoring a specific project's progress and performance in satisfying resilience requirements. (Responsibility is typically assigned to the organizational owner of the software or system or the applicable service owner.) These procedures should be consistent with development plan tasks and activities. It may be appropriate to adjust the monitoring frequency in response to changes in the risk environment, changes to resilience requirements, and changes in project staff.

Typical work products

1. Project review procedures
2. Project measures, reports, and review results
3. Updated project plans
4. Updated resilience guidelines
5. Updated process definitions

Subpractices

1. Monitor project performance against the development plan to ensure that resilience requirements are satisfied.

   This may include, for example, collecting, analyzing, and reporting

   - the effectiveness of resilience guidelines in satisfying resilience requirements
   - status toward meeting planned milestones that demonstrate the satisfaction of resilience requirements
   - actual expenditures against budgeted expenditures for implementing resilience guidelines
   - identified risks and risk mitigation plans
   - impacts to service continuity plans for the software or system in development
   - impacts to controls for protecting and sustaining services, software, and systems
   - improvements to resilience guidelines and process definitions that address resilience

2. Update development project plans, resilience guidelines, and process definitions as appropriate.

Updates may include normal, expected changes and improvements. Updates may also include the results of having to renegotiate resilience requirements, failure to meet milestone review criteria, failure to implement resilience guidelines, cost and schedule deviations beyond established thresholds, and reassignment of vital staff. All of these conditions should be identified and managed as project risks.

RTSE:SG3.SP2 Release Resilient Technical Solutions into Production

Software and systems that demonstrate satisfaction of resilience requirements are released into production.

Prior to releasing software or system assets into an operational production environment, these assets should undergo a formal inspection against documented criteria to ensure they have met specified resilience requirements.

The result of satisfying inspection criteria is approval to release software and system assets into production (sometimes referred to as "authority or authorization to operate").

Typical work products

1. Inspection criteria
2. Inspection procedures
3. Inspection results
4. Production-ready software and system assets

Subpractices

1. Establish inspection criteria.

   Inspection criteria should be sufficient to provide an acceptable level of assurance and confidence to release software and systems into operational production environments. Such criteria will likely include

   - documented evidence in support of selected assurance cases (Refer to RTSE:SG1.SP1.)
   - results of testing approaches and test cases used during implementation and assembly as called for in resilience guidelines reflected in development plans (Refer to RTSE:SG1.SP2 and SP3.)
   - demonstrated satisfaction of resilience requirements overall and in support of service continuity plans for services supported by the assets being inspected
   - the availability of complete and thorough asset documentation, including updated asset inventories (Refer to the Asset Definition and Management process area.)
   - demonstrating that controls for protecting and sustaining services, software, and systems are implemented and adequate

   Some criteria or subcriteria may be selected based on the priority of the software or system such that higher value assets are subjected to more stringent or more comprehensive inspection.

2. Inspect software and systems to ensure they have satisfied inspection criteria.

To ensure that asset inspections are performed in a predictable, repeatable manner, the organization should establish procedures that specify the protocol and responsibility for performing an asset inspection against established criteria. Inspection procedures should include documentation of inspection results, including any actions that have to be closed prior to release and the identification of new and residual risks that have to be managed (*refer to the Risk Management process area*).

Responsibility for assembling a qualified inspection team and conducting inspections may be assigned to managers who have direct responsibility for the service or services that the asset will be supporting as well as the asset's performance in the operational production environment. Quality assurance and internal audit staff may also fill this role. Staff conducting inspections should be sufficiently knowledgeable and experienced to verify and validate the satisfaction of all established inspection criteria.

3. Approve assets for release.

Assets are approved for release into the operational production environment upon demonstrating that they have met all established inspection criteria. The approval process may allow for waivers of specific criteria based on high-level manager approvals.

Elaborated Generic Practices by Goal

Refer to the Generic Goals and Practices document in Appendix A for general guidance that applies to all process areas. This section provides elaborations relative to the application of the Generic Goals and Practices to the Resilient Technical Solution Engineering process area.

RTSE:GG1 Achieve Specific Goals

The operational resilience management system supports and enables achievement of the specific goals of the Resilient Technical Solution Engineering process area by transforming identifiable input work products to produce identifiable output work products.

RTSE:GG1.GP1 Perform Specific Practices

Perform the specific practices of the Resilient Technical Solution Engineering process area to develop work products and provide services to achieve the specific goals of the process area.

Elaboration:

Specific practices RTSE:SG1.SP1 through RTSE:SG3.SP2 are performed to achieve the goals of the resilient technical solution engineering process.

RTSE:GG2 Institutionalize a Managed Process

Resilient technical solution engineering is institutionalized as a managed process.

RTSE:GG2.GP1 Establish Process Governance

Establish and maintain governance over the planning and performance of the resilient technical solution engineering process.

Refer to the Enterprise Focus process area for more information about providing sponsorship and oversight to the resilient technical solution engineering process.

Subpractices

1. Establish governance over process activities.

   Elaboration:

   Governance over the resilient technical solution engineering process may be exhibited by

   - establishing a higher level position, often the chief information officer, responsible for the resilience of the organization’s software and system assets
   - developing and publicizing higher level managers’ objectives and requirements for developing resilient software and systems
   - oversight over the development, acquisition, operations, and management of high-value software and system assets
   - sponsoring and providing oversight of policies, procedures, standards, and guidelines for the development of software and system assets and for establishing asset ownership and custodianship
   - making higher level managers aware of applicable compliance obligations related to the development of resilient software and systems, and regularly reporting on the organization’s satisfaction of these obligations to higher level managers
   - oversight over the establishment, implementation, and maintenance of the organization’s internal control system for software and system assets, including those for protection, continuity, and sustainment during the development life cycle
   - sponsoring and funding process activities
   - implementing a technology steering committee that includes software and systems under development
   - providing guidance for prioritizing software and system assets relative to the organization’s high-priority strategic objectives
   - providing guidance on identifying, assessing, and managing operational risks related to software and system assets in development
   - providing guidance for resolving gaps or shortfalls in the satisfaction of software and system resilience requirements during development
   - verifying that the process supports strategic resilience objectives and is focused on the assets and services that are of the highest relative value in meeting strategic objectives
   - regular reporting from organizational units with responsibility for development projects to higher level managers on process activities and results

- creating dedicated higher level management feedback loops on decisions about the process and recommendations for improving the process
- conducting regular internal and external audits and related reporting to appropriate committees on software and system asset controls and the effectiveness of the process
- creating formal programs to measure the effectiveness of process activities, and reporting these measurements to higher level managers

2. Develop and publish organizational policy for the process.

Elaboration:

The resilient technical solution engineering policy should address

- responsibility, authority, and ownership for performing process activities
- integrating resilience guidelines with a defined software development process
- procedures, standards, and guidelines for
  - developing software and systems that meet their resilience requirements during all life-cycle phases
  - describing and identifying software and system owners and custodians
  - developing and documenting resilience requirements for software and system assets (*Refer to the Resilience Requirements Development process area.*)
  - establishing, implementing, and maintaining an internal control system for software and systems, and controls to sustain services and the systems and software on which they depend
  - maintaining environmental conditions for physical components of systems (hardware and infrastructure)
  - managing software and system asset risk, in development and in operations
  - establishing software and system asset service continuity plans and procedures
  - retiring software and system assets at the end of their useful life
  - architectural interoperability
  - project reviews
  - formal inspections prior to releasing software and system assets into production
- the association of software and system assets to core organizational services, and the prioritization of assets for service continuity
- requesting, approving, and providing access to software and system assets to persons, objects, and entities, including type and extent of access and requests that originate externally to the organization *(Refer to the Access Management process area for more information about granting access [rights and privileges] to software and system assets. Refer to the Identity Management process area for more information about creating and maintaining identities for persons, objects, and entities.)*
- methods for measuring adherence to policy, exceptions granted, and policy violations

RTSE:GG2.GP2 Plan the Process

Establish and maintain the plan for performing the resilient technical solution engineering process.

Elaboration:

A plan for performing the resilient technical solution engineering process is created to ensure that resilience is considered in the development process for all software and systems. The plan must address the inclusion of resilience guidelines in the software and system development plans and development life-cycle process definitions, as well as consideration of multiple asset owners, custodians, and stakeholders at various levels of the organization. In addition, because software and system assets may be developed and deployed in more than one geographical location by more than one development organization, the plan must extend to external stakeholders that can enable or adversely affect software and system resilience during development.

The plan for the resilient technical solution engineering process should not be confused with software and system development plans *(refer to RTSE:SG2.SP2 and RTSE:SG3)*. The plan for the resilient technical solution engineering process details how the organization will ensure that software and system assets are developed to satisfy their resilience requirements, including updating software and system development plans and life-cycle process definitions to reflect resilience guidelines.

Subpractices

1. Define and document the plan for performing the process.

   Elaboration:

   Special consideration in the plan may have to be given to establishing, implementing, and maintaining an internal control system for software and system assets, including planning to ensure these assets are protected, sustained, and continue to operate as intended. These activities are determined commensurate with software and system resilience requirements and the extent to which they support high-value services.

2. Define and document the process description.
3. Review the plan with relevant stakeholders and get their agreement.
4. Revise the plan as necessary.

RTSE:GG2.GP3 Provide Resources

Provide adequate resources for performing the resilient technical solution engineering process, developing the work products, and providing the services of the process.

Subpractices

1. Staff the process.

   Elaboration:

   These are examples of staff required to implement and support the resilient technical solution engineering process:

   - staff responsible for

  - software and system security during development and acquisition
  - business continuity and disaster recovery for software and system assets
  - implementing and maintaining software and system asset security controls (such as software architects, designers, developers, and testers trained in resilience requirements and guidelines)
  - configuration management, change management, and release management of software and system assets
  - trade-off analyses in support of guideline selection and prioritization
  - software and system development processes
  - project reviews
  - quality assurance
- staff involved in identifying and managing risk for software and systems in development
- external entities responsible for developing, implementing, and maintaining software and system assets
- owners and custodians of software and system assets (to identify and enforce resilience requirements)

Refer to the Organizational Training and Awareness process area for information about training staff for resilience roles and responsibilities.

Refer to the Human Resource Management process area for information about acquiring staff to fulfill roles and responsibilities.

2. Fund the process.

Elaboration:

At a minimum, funding must be available to execute software and system development plans that incorporate selected resilience guidelines.

Refer to the Financial Resource Management process area for information about budgeting for, funding, and accounting for the development of resilience software and system assets.

3. Provide necessary tools, techniques, and methods to perform the process.

Elaboration:

Keep in mind that many of the automated tools used to support the resilient technical solution engineering process are themselves software assets that have to be managed according to the process.

These are examples of tools, techniques, and methods to support the resilient technical solution engineering process:

- project management tools
- threat analysis methods, techniques, and tools
- methods for representing defender and attacker perspectives such as misuse/abuse cases
- quality assurance methods such as vulnerability analysis
- methods and techniques for conducting resilience guidelines trade-off analyses and prioritizing resilience guidelines

- tools, techniques, and methods for
  - supporting and automating the guidelines that have been selected for each development life-cycle phase (requirements, architecture and design, implementation, and assembly and integration)
  - identifying and managing risks to software and system assets by life-cycle phase, including tracking open risks to closure and monitoring the effectiveness of asset risk mitigation plans
  - maintaining software and system assets, including asset configuration management, change control, release management, and monitoring and logging of modification activities
  - ensuring software and system asset integrity during development, such as code signing
  - controlling access to software and system assets
  - analyzing open-source, COTS, and legacy software
  - measuring, reviewing, testing, monitoring, auditing, and inspecting software and systems at key milestones in their development life cycle
  - software and system asset backup, retention, and restoration throughout the development life cycle
  - managing software and system assets that are provided by external entities
- methods for establishing, implementing, and maintaining the internal control system for software and system assets throughout the development life cycle
- methods for the proper retirement and disposal of software and system assets

RTSE:GG2.GP4 Assign Responsibility

Assign responsibility and authority for performing the resilient technical solution engineering process, developing the work products, and providing the services of the process.

Elaboration:

Of paramount importance in assigning responsibility for the resilient technical solution engineering process is the establishment of software and system asset owners *(which is described in ADM:SG1.SP3)*. Owners are responsible for establishing asset resilience requirements, ensuring these requirements are met throughout the development life cycle, and identifying and remediating gaps and risks where requirements are not being met. Owners may also be responsible for establishing, implementing, and maintaining an internal control system commensurate with meeting asset resilience requirements.

Refer to the Human Resource Management process area for more information about establishing resilience as a job responsibility, developing resilience performance goals and objectives, and measuring and assessing performance against these goals and objectives.

Refer to the Asset Definition and Management process area for more information about establishing ownership and custodianship of software and system assets.

Subpractices

1. Assign responsibility and authority for performing the process.

Elaboration:

Responsibility and authority may extend not only to staff inside the organization but to those with whom the organization has a contractual agreement for developing, implementing, and managing software and system assets (including implementation and management of controls and ensuring assets are sustained).

2. Assign responsibility and authority for performing the specific tasks of the process.

   Elaboration:

   Responsibility and authority for performing resilient technical solution engineering tasks can be formalized by

   - defining roles and responsibilities in the process plan and in software and system development plans
   - including process tasks and responsibility for those tasks in specific job descriptions
   - developing policy requiring organizational unit managers, line of business managers, project managers, and asset and service owners and custodians to participate in the process for assets under their ownership or custodianship
   - including process tasks in staff performance management goals and objectives, with requisite measurement of progress against those goals
   - developing and implementing contractual instruments (as well as service level agreements) with external entities to establish responsibility and authority for outsourced and acquired software and system assets
   - including process tasks in measuring performance of external entities against service level agreements *(Refer to the External Dependencies Management process area for additional details about managing relationships with external entities.)*

3. Confirm that people assigned with responsibility and authority understand it and are willing and able to accept it.

RTSE:GG2.GP5 Train People

Train the people performing or supporting the resilient technical solution engineering process as needed.

Refer to the Organizational Training and Awareness process area for more information about training the people performing or supporting the process.

Refer to the Human Resource Management process area for more information about inventorying skill sets, establishing a skill set baseline, identifying required skill sets, and measuring and addressing skill deficiencies.

Subpractices

1. Identify process skill needs.

Elaboration:

These are examples of skills required in the resilient technical solution engineering process:

- software and system requirements engineering, architecture and design, implementation, and assembly and integration
- developing and selecting resilience guidelines that are used throughout the development life cycle
- managing the development of software and systems
- conducting effective trade-off analyses to inform decision making
- knowledge of tools, techniques, and methods that can be used to identify, analyze, mitigate, and monitor operational risks to software and system assets during their development life cycle
- establishing, implementing, and maintaining the internal control system for software and system assets during their development life cycle
- protecting and sustaining software and system assets to meet their resilience requirements

2. Identify process skill gaps based on available resources and their current skill levels.
3. Identify training opportunities to address skill gaps.

Elaboration:

These are examples of training topics:

- software and system asset risk management concepts and activities (e.g., risk identification, analysis, mitigation, and monitoring) during development
- software and system asset resilience requirements development
- establishing, implementing, and maintaining an internal control system for protecting and sustaining software and system assets
- cross-training to ensure adequate knowledge and coverage of all software and system assets that are part of a specific development project
- proper techniques for inspecting software and system assets, including approval to release them into production
- software and system asset configuration, change, and release management
- supporting software and system asset owners and custodians in understanding the process and their roles and responsibilities with respect to its activities
- working with external entities that have responsibility for process activities
- using process methods, tools, and techniques, including those identified in RTSE:GG2:GP3 subpractice 3

4. Provide training and review the training needs as necessary.

RTSE:GG2.GP6 Control Work Products

Place designated work products of the resilient technical solution engineering process under appropriate levels of control.

Elaboration:

These are examples of resilient technical solution engineering work products placed under control:

- guidelines for the development of resilient software and systems, including general, requirements, architecture and design, implementation (coding and testing), and assembly and integration
- resilience guideline selection criteria
- development process definitions, updated to reflect resilience guidelines
- development plans, updated to reflect resilience guidelines
- administrative, technical, and physical controls for software and systems
- new and residual risks associated with resilience requirements and guidelines
- project review procedures
- project measures, reports, and results of reviews
- inspection criteria, procedures, and results
- production-ready software and system assets
- modification logs and audit reports
- baseline configuration items and configuration control logs and reports
- configuration management, change management, and release management systems
- baseline archives and backup media
- release builds, testing procedures, and release-build test results
- updated service continuity plans
- process plan
- policies and procedures
- contracts with external entities

Refer to the Technology Management process area for more information about managing software and system assets during operations.

RTSE:GG2.GP7 Identify and Involve Relevant Stakeholders

Identify and involve the relevant stakeholders of the resilient technical solution engineering process as planned.

Subpractices

1. Identify process stakeholders and their appropriate involvement.

   Elaboration:

   Because software and system assets may reside in a wide range of physical locations and be developed and maintained by internal and external entities, a substantial number of stakeholders are likely to be external to the organization.

   These are examples of stakeholders of the resilient technical solution engineering process:

- owners and custodians of software and system assets
- service and business process owners
- organizational unit and line of business managers responsible for high-value software and system assets and the services they support
- staff responsible for managing development risks to software and system assets
- staff responsible for establishing, implementing, and maintaining an internal control system for software and system assets, including those responsible for configuration, change, and release management
- staff required to develop, test, implement, and execute service continuity plans for software and system assets
- staff responsible for maintaining process definitions for software and system development
- staff in other organizational support functions, such as accounting or general services administration (particularly as related to software and system inventory valuation and retirement)
- staff responsible for reviewing and approving assets prior to their release into production, including internal and external auditors

Stakeholders are involved in various tasks in the resilient technical solution engineering process, such as

- planning for software and system development, including the selection and tailoring of resilience guidelines
- adding new software and systems to the technology asset inventory
- creating software and system asset profiles and asset risk and vulnerability profiles
- associating software and system assets with services and analyzing service dependencies
- assigning resilience requirements for software and system assets
- establishing, implementing, and maintaining controls for software and system assets
- developing service continuity plans for software and system assets
- managing development risks to software and system assets
- managing software and system configurations, changes, and releases
- controlling the development environment in which software and system assets reside
- managing software and system asset external dependencies for assets developed and maintained by external entities
- managing relationships with external entities that develop software and system assets (or components thereof)
- reviewing and appraising the effectiveness of process activities
- resolving issues in the process

2. Communicate the list of stakeholders to planners and those responsible for process performance.
3. Involve relevant stakeholders in the process as planned.

RTSE:GG2.GP8 Measure and Control the Process

Measure and control the resilient technical solution engineering process against the plan for performing the process and take appropriate corrective action.

Refer to the Monitoring process area for more information about the collection, organization, and distribution of data that may be useful for measuring and controlling processes.

Refer to the Measurement and Analysis process area for more information about establishing process metrics and measurement.

Refer to the Enterprise Focus process area for more information about providing process information to managers, identifying issues, and determining appropriate corrective actions.

Subpractices

1. Measure actual performance against the plan for performing the process.
2. Review accomplishments and results of the process against the plan for performing the process.

Elaboration:

These are examples of metrics for the resilient technical solution engineering process:

- percentage of software assets that have been developed without resilience guidelines, by guideline type (general, requirements, architecture and design, implementation, assembly and integration)
- percentage of software assets that have been acquired without consideration of resilience guidelines, by guideline type (general, requirements, architecture and design, implementation, assembly and integration)
- percentage of software development staff trained in the tailoring and use of resilience guidelines, by guideline type (general, requirements, architecture and design, implementation, assembly and integration)
- life-cycle costs associated with implementing each resilience guideline (time, staff resources, and funding, including training) or some meaningful collection of guidelines
- percentage of resilience requirements not satisfied by a specific software or system asset by life-cycle phase and ranked in priority order (refer to RRD)
- percentage of resilience requirements not satisfied by a specific software or system asset, where lack of satisfaction has been identified as a residual risk to be managed
- number of defects and vulnerabilities above threshold for a specific software or system asset by life-cycle phase
- number of defects and vulnerabilities above threshold for a specific software or system asset where such defects and vulnerabilities have documented mitigation plans
- number of defects and vulnerabilities above threshold for a specific software or system asset where such defects and vulnerabilities have been identified as residual risks to be managed

- number of defects and vulnerabilities above threshold for a specific software or system assets where the presence of such defects and vulnerabilities is a result of not implementing a resilience guideline
- percentage of software assets for which some form of risk assessment has not been performed and documented (per policy or other resilience guidelines) and within the specified time frame, by life-cycle phase
- percentage of system assets for which some form of risk assessment has not been performed and documented (per policy or other resilience guidelines) and within the specified time frame, by life-cycle phase
- number of unauthorized changes to software assets, by life-cycle phase
- number of unauthorized changes to sys-tem assets, by life-cycle phase
- inspection yield: defects found during the inspection / (defects found during the inspection + those that escaped the inspection)
- inspection removal rate: effort spent in inspection / number of defects found in inspection
- planned versus actual number of inspections
- percentage of software assets released into production without consideration of resilience guidelines
- percentage of system assets released into production without consideration of resilience guidelines
- elapsed time between the identification of a newly released software or system asset and its inclusion in the asset inventory
- number of software and system development risks referred to the risk management process
- percentage of software and system development policies that are met
- test defect density (number of vulnerabilities found in test / size of software asset)usage defect density (number of vulnerabilities found while using software or number of incidents that occurred while using software / size of software asset)

3. Review activities, status, and results of the process with the immediate level of managers responsible for the process and identify issues.

Elaboration:

Periodic reviews of the resilient technical solution engineering process are needed to ensure that

- software and system assets have stated resilience requirements
- software and system assets are being developed in accordance with selected resilience guidelines
- software and system assets are satisfying their resilience requirements appropriate to the life-cycle phase under review
- administrative, technical, and physical controls are implemented during development
- controls are meeting the stated intent of software and system resilience requirements
- software and system process definitions and development plans are updated as required

- newly developed and acquired software and system assets are included in the asset inventory
- asset-service mapping is accurate and current
- ownership and custodianship of software and system assets are established and documented
- status reports are provided to appropriate stakeholders in a timely manner
- software and system development issues are referred to the risk management process when necessary
- actions requiring management involvement are elevated in a timely manner
- the performance of process activities is being monitored and regularly reported
- key measures are within acceptable ranges as demonstrated in governance dashboards or scorecards and financial reports
- actions resulting from inspections and internal and external audits are being closed in a timely manner

4. Identify and evaluate the effects of significant deviations from the plan for performing the process.
5. Identify problems in the plan for performing and executing the process.
6. Take corrective action when requirements and objectives are not being satisfied, when issues are identified, or when progress differs significantly from the plan for performing the process.

   Elaboration:

   For software and system assets, corrective action may require the revision of existing resilience requirements (*refer to the Resilience Requirements Management process area*). Corrective action may also require the revision of existing administrative, technical, and physical controls, development and implementation of new controls, or a change in the type of controls (preventive, detective, corrective, compensating, etc.).
7. Track corrective action to closure.

RTSE:GG2.GP9 Objectively Evaluate Adherence

Objectively evaluate adherence of the resilient technical solution engineering process against its process description, standards, and procedures, and address non-compliance.

Elaboration:

These are examples of activities to be reviewed:

- identifying and prioritizing software and system assets that are to be developed
- identifying software and system asset resilience requirements
- identifying, selecting, and implementing software and system asset resilience guidelines
- establishing and implementing software and system asset controls
- identifying and managing software and system asset risks
- developing service continuity plans for newly developed software and system assets
- identifying and managing software and system asset dependencies
- identifying and managing changes to software and system assets
- decisions to not release software and system assets into production

- aligning stakeholder requirements with the process plan
- assigning responsibility, accountability, and authority for process activities
- determining the adequacy of process reports and reviews in informing decision makers regarding the performance of operational resilience management activities and the need to take corrective action, if any
- verifying controls on software and system assets
- using process work products for improving strategies for protecting and sustaining software and system assets

These are examples of work products to be reviewed:

- resilience guidelines and costs to implement them
- project review results by life-cycle phase
- software and system asset internal controls documentation
- software and system asset risk statements and mitigation plans
- service continuity plans for newly developed software and system assets
- business impact analysis results for newly developed software and system assets
- results of inspections for releasing software and system assets into production
- contracts with external entities
- process plan and policies
- software and system development issues that have been referred to the risk management process
- process methods, techniques, and tools
- metrics for the process *(Refer to RTSE:GG2.GP8 subpractice 2.)*

RTSE:GG2.GP10 Review Status with Higher Level Managers

Review the activities, status, and results of the resilient technical solution engineering process with higher level managers and resolve issues.

Elaboration:

Status reporting on the resilient technical solution engineering process may be part of the formal governance structure or may be performed through other organizational reporting requirements (such as through the chief risk officer or the chief resilience officer level). Audits of the process, particularly the validation and verification of asset resilience requirements satisfaction and the internal control system at points in time, may be escalated to higher level managers through the organization's audit committee of the board of directors or similar construct in private or non-profit organizations.

Refer to the Enterprise Focus process area for more information about providing sponsorship and oversight to the operational resilience management system.

RTSE:GG3 Institutionalize a Defined Process

Resilient technical solution engineering is institutionalized as a defined process.

RTSE:GG3.GP1 Establish a Defined Process

Establish and maintain the description of a defined resilient technical solution engineering process.

Elaboration:

Managing the development of resilient software and systems is typically carried out by a project at the organizational unit or line of business level for convenience and accuracy and may have to be geographically focused (because of the location of specific assets and the skilled staff to develop them). However, to achieve consistent results in developing software and system assets, the activities at the project level must be derived from an enterprise definition of the resilient technical solution engineering process. Resilience guidelines and the selected software development process may be inconsistent across projects, particularly when assets support multiple services and thus have shared ownership across organizational lines, but the defined process remains consistent. The development of software and system assets by multiple organizational units and lines of business may affect asset management at the enterprise level and impede operational resilience.

In addition, a variable mix of administrative, technical, and physical controls may be used across the organization to meet the resilience requirements for software and system assets, but the process is consistent with the enterprise definition.

Establishing and tailoring process assets, including standard processes, are addressed in the Organizational Process Definition process area.

Establishing process needs and objectives and selecting, improving, and deploying process assets, including standard processes, are addressed in the Organizational Process Focus process area.

Subpractices

1. Select from the organization’s set of standard processes those processes that cover the resilient technical solution engineering process and best meet the needs of the organizational unit or line of business.
2. Establish the defined process by tailoring the selected processes according to the organization’s tailoring guidelines.
3. Ensure that the organization’s process objectives are appropriately addressed in the defined process, and ensure that process governance extends to the tailored processes.
4. Document the defined process and the records of the tailoring.
5. Revise the description of the defined process as necessary.

RTSE:GG3.GP2 Collect Improvement Information

Collect resilient technical solution engineering work products, measures, measurement results, and improvement information derived from planning and performing the process to support future use and improvement of the organization's processes and process assets.

Elaboration:

These are examples of improvement work products and information:

- updates to software and system development process definitions
- updates to resilience guidelines
- updates to software and system development plans
- resilience requirements that are not being satisfied by software and system assets or are being exceeded
- reports on the effectiveness and weaknesses of controls
- improvements based on risk identification and mitigation
- software and system test and inspection results
- metrics and measurements of the viability of the process *(Refer to RTSE:GG2.GP8 subpractice 2.)*
- changes and trends in operating conditions, risk conditions, and the risk environment that affect process results
- lessons learned in post-event review of software and system asset incidents and disruptions in continuity
- conflicts and risks arising from dependencies on external entities
- lessons learned in updating, replacing, and retiring software and system assets from active use

Establishing the measurement repository and process asset library is addressed in the Organizational Process Definition process area. Updating the measurement repository and process asset library as part of process improvement and deployment is addressed in the Organizational Process Focus process area.

Subpractices

1. Store process and work product measures in the organization's measurement repository.
2. Submit documentation for inclusion in the organization's process asset library.
3. Document lessons learned from the process for inclusion in the organization's process asset library.
4. Propose improvements to the organizational process assets.