

Software Assurance Objectives Hierarchy – Top Level

Top Objective: Software performs what is intended, only what is intended, and only in the intended manner

Strategy: Plan and execute Software Assurance throughout the software lifecycle

Objective: Software development and assurance processes are necessary and sufficient to achieve the project's desired levels of safety, quality, security, and reliability
(1)

SA Planning
Section 9.1

Objective: Software conforms to functional intent and performs as planned
(2)

SW Quality
Section 9.2

Objective: Software does not adversely impact safety and contributes to system safety
(3)

SW Safety
Section 9.3

Objective: Software system is robust and tolerant to failure & off nominal conditions
(4)

SW Reliability &
Maintainability
Section 9.4

Objective: Software is secure and does not adversely impact safety and functionality of the system.
(5)

SW Security
Section 9.5

Objective: Software Verification and Validation Processes provide confidence in the interim and end products
(6)

SW V&V/IV&V
Section 9.6/9.8

Sub-obj. **1** SA Planning

Objective: Software development and assurance processes are necessary and sufficient to achieve the project's desired levels of safety, quality, security and reliability (1)

Strategy: Assure a software development process (1.A)

Objective: Software development planned processes have rigor appropriate to the project's risk posture and conditions (1.A.1)

Strategy: Assess the software level and criticality (1.A.1.A)

Strategy: Assure a comprehensive and mature software development process is planned (1.A.1.B)

Strategy: Assure development organization(s) have the capability to perform as planned (1.A.1.C)

Strategy: Assure a software acquisition process (1.B)

Objective: Software acquisition planned processes have rigor appropriate to the project's risk posture and conditions (1.B.1)

Strategy: Assure appropriate level SW development & assurance plans and processes are levied on providers (1.B.1.A)

Strategy: Assure acquired SW meets the project's risk posture and conditions (1.B.1.B)

Strategy: Plan software assurance activities (1.C)

Objective: Software assurance planned activities are appropriate to the project's risk posture and conditions (1.C.1)

Strategy: Estimate the cost of each applicable software assurance activity and the risk reduction it would provide (1.C.1.A)

Strategy: Create the software assurance plan (1.C.1.B)

Objective: Software assurance is supported to the appropriate level (1.C.2)

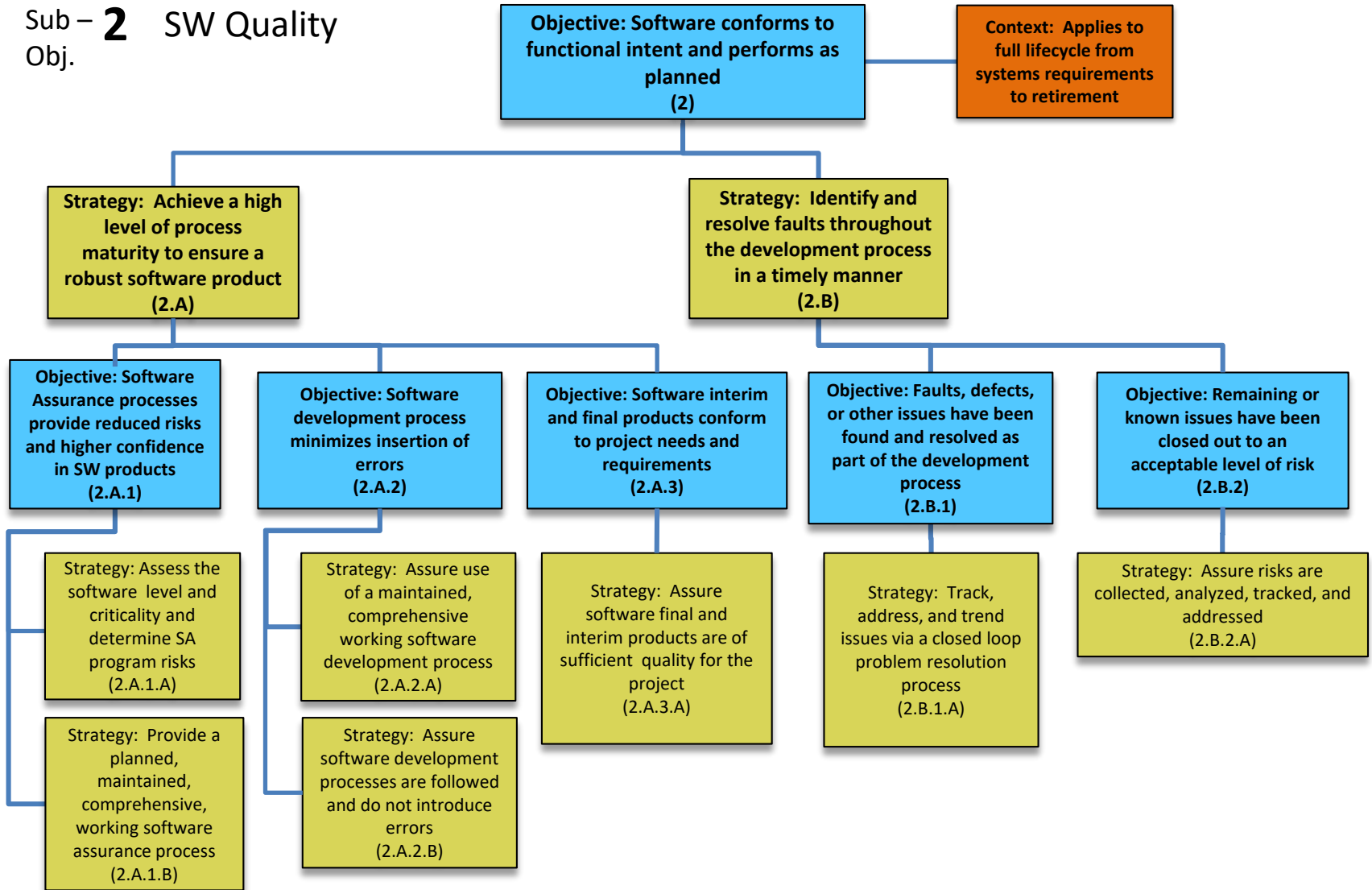
Strategy: Allocate assurance personnel sufficient to execute SA plan (1.C.2.A)

Strategy: Train assurance personnel for needed skills & knowledge of project (1.C.2.B)

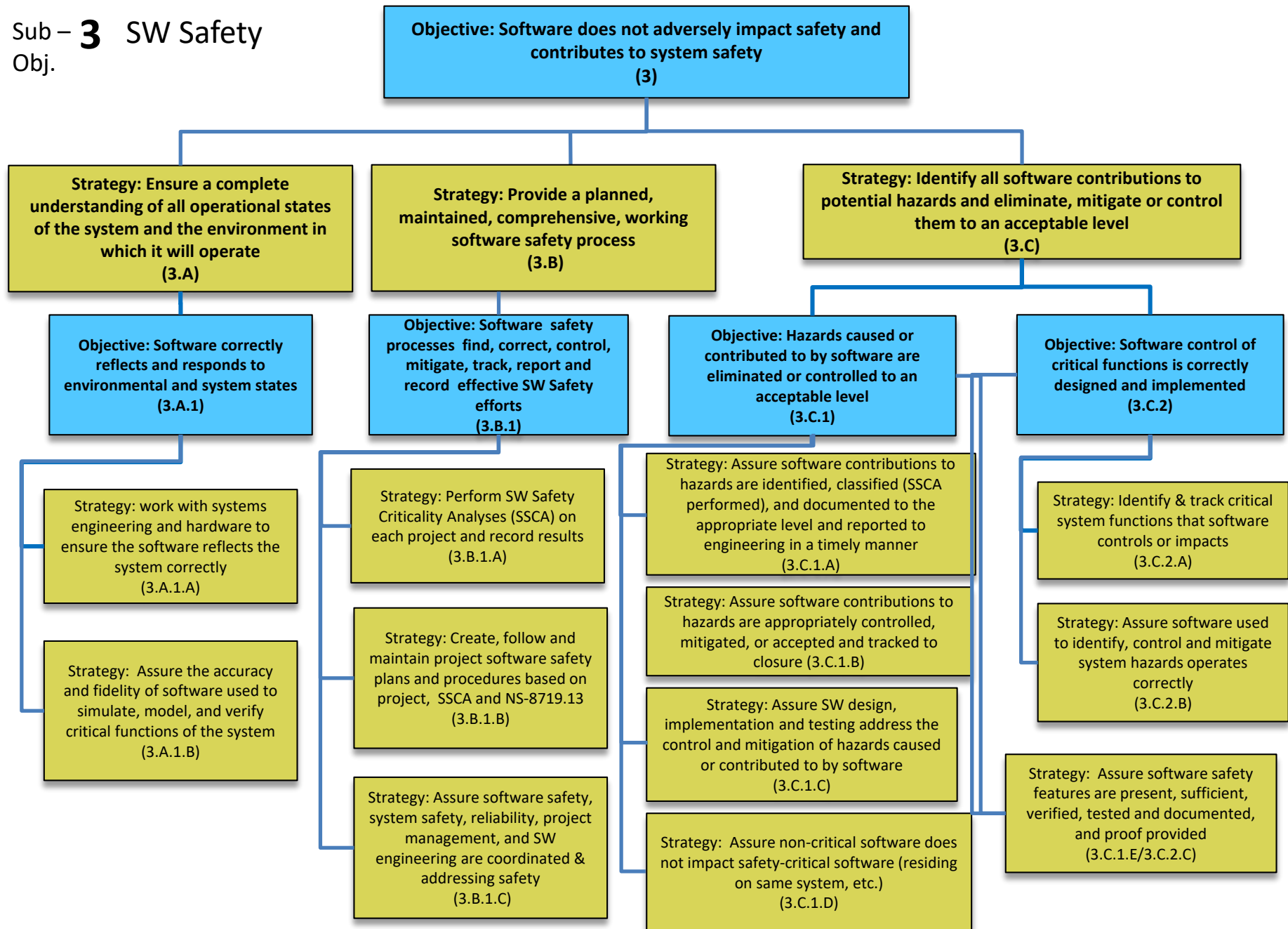
Strategy: Provide assurance personnel needed tools and resources (1.C.2.C)

Strategy: Provide SA access to project data & activities (1.C.2.D)

Sub – 2 SW Quality
Obj.



Sub – 3 SW Safety
Obj.



Sub – **4** SW Reliability & Maintainability
Obj.

Objective: Software system is robust and tolerant to failure & off nominal conditions (4)

Strategy: Assure that software is developed in a robust manner, which decreases/eliminates errors and determines residual risk (4.A)

Strategy: Assure that software is developed with necessary design features to prevent failures and off-nominal conditions from compromising ability to accomplish mission objectives (4.B)

Strategy: Assure continued safe and reliable operations with managed and assured updates (4.C)

Objective: Areas of software development weakness are known and addressed to the proper level (4.A.1)

Objective: Software residual risks and faults are known or predicted and addressed (4.A.2)

Objective: Software is designed to enhance system reliability (4.B.1)

Objective: SW is designed for robust operation (4.B.2)

Objective: SW is maintained for robust operation (4.C.1)

Strategy: Assure robust development environment (4.A.1.A)

Strategy: Identify, classify, collect, trend and report defect metrics (4.A.1.B)

Strategy: Analyze metrics, requirements, and design to predict and address problem areas (4.A.1.C)

Strategy: Predict software potential faults from past performance or similar systems (4.A.2.A)

Strategy: Predict remaining software faults based on testing and analyses, and address and report (4.A.2.B)

Strategy: Participate in system analysis and design to determine areas of weakness, suggesting where software design can improve system reliability (4.B.1.A)

Strategy: Determine appropriate level for functional redundancy between system and software (4.B.1.B)

Strategy: Design in appropriate software architecture to meet and maintain critical functions (4.B.1.C)

Strategy: Assure appropriate software/system functional redundancy (4.B.1.D)

Strategy: Analyze software functionality, requirements and design for vulnerabilities and recommend design strategies (4.B.2.A)

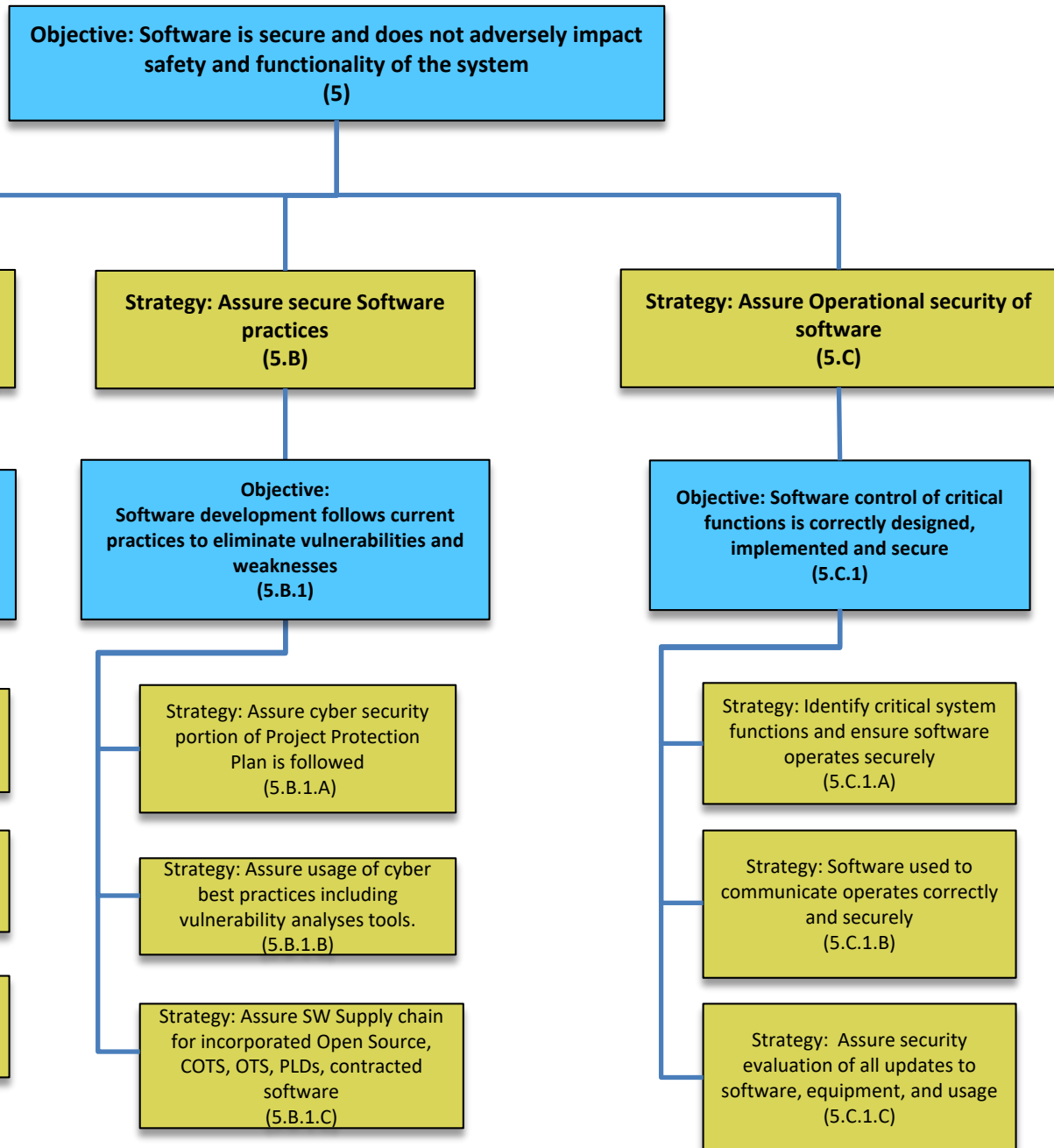
Strategy: Design in protective barriers to prevent software from propagating system faults (system includes software, hardware, and human interface) (4.B.2.B)

Strategy: Provide fault management (detection, isolation, recovery) capabilities (4.B.2.C)

Strategy: Review and assess SW engineering's maintainability plan assuring provisions for contracted, in-house, COTS, GOTS, Open Source and PLD version changes are addressed and followed (4.C.1.A)

Strategy: Assess and report on SW maintainability requirements, design and testing, checking for required system stated maintainability goals (4.C.1.B)

Sub – 5 SW Security
Obj.



Sub – 6 Software
Obj. V&V/IV&V

Objective: Software Verification and Validation Processes provide confidence in the interim and end products (6)

Context: V&V includes reviews, inspections, testing, analyses, demos

Strategy: Verify and validate functionality of software products (6.A)

Objective: Software V&V assures confidence in the interim and end software products (6.A.1)

Strategy: Perform reviews on software products, reporting results and findings to Project (6.A.1.A)

Strategy: Track all review, inspection, testing findings to closure and monitor closure rate (6.A.1.B)

Strategy: Assure test procedures, test sets, test drivers & stubs are complete properly run, and configuration managed; the test results are properly signed off (6.A.1.C)

Strategy: Assure simulations and models are correct and provide necessary inputs & outputs (6.A.1.D)

Strategy: Assure software product deliveries and final as-built configuration (6.A.1.E)

Strategy: Assure the software V&V processes are in place and working to find and remove defects (6.B)

Objective: Assure Software V&V processes are providing robust and safe software products (6.B.1)

Strategy: Assure a maintained, comprehensive working software V&V process exists (6.B.1.A)

Strategy: Assure software V&V procedures are complete, and nominal and off nominal conditions are addressed (6.B.1.B)

Strategy: Assure execution of V&V planning and procedures (6.B.1.C)

Strategy: Independently verify and validate functionality and process for NASA's most mission and safety critical software products (6.C)

Objective: Provide assurance that the safety and mission-critical software will operate reliably and safely (6.C.1)

Strategy: Assure systems' software will perform expected functionality (6.C.1.A)

Strategy: Assure systems' software will not perform unwanted/undesired actions (6.C.1.B)

Strategy: Assure systems' software responds as expected under adverse conditions (6.C.1.C)