**Software Verification & Validation (V&V)**

**Test Plan (SWTP) Template**

**for use with**

**ESM Chapter 21, Software, Section SOFT-V&V**

**Usage Notes (delete from final SWTP):**

1. Use of this draft (of LANL Template 3055r1) is highly encouraged.
	1. Chapter 21 requires an SWTP but doesn’t not mandate the format, so use of best available template, including this draft, is acceptable.
2. User is responsible for meeting the SWTP requirements of Ch. 21 SOFT-V&V (this template draft has not been verified to contain all necessary steps/information to ensure same).
3. There are template instructions in the second half of the file (starting ~ page 12). It refers to P1040 requirements; however, Ch. 21 users are not subject to P1040. P1040 references can be viewed as additional guidance to Ch. 21 (Ch. 21 has similar but somewhat different requirements).
4. A separate, draft test case (SWTC) template is also posted with SOFT-V&V.
5. Suggestions on the template are welcome; please consolidate them and send to P1040@lanl.gov and stdsweb@lanl.gov

|  |
| --- |
| **1.0 SOFTWARE information** |
| **1.1 Software Name and Version** |  |
| 1.2 Software ID (SWID) |  |
| 1.3 Software Description & Use  |  |
| 1.4 Software Designation (from Form 2033) (check one) | 1.5a SW Risk Level |
|[ ]  Safety Software: SSS (Safety System SW) |[ ]  1 |[ ]  2 |[ ]  3 |
|[ ]  Safety Software: SHADS (Safety Hazard Analysis SW and Design SW) | 1.5b Management Level |
|[ ]  Safety Software: SMACS (Safety Mgmt. and Administrative Controls SW) |[ ]  1 |[ ]  2 |[ ]  3 |
|[ ]  Non-safety: Risk Significant Software |  |  |
|[ ]  Non-safety: Commercially Controlled Software |  |

|  |
| --- |
| **2.0 SOFTWARE TEST PLAN (SWTP) REVIEW AND APPROVAL** |
| The following individuals have reviewed and approved this SWTP and accept their responsibilities as described in Section 5.0.  |
| 2.1 Name (last, first) | 2.2 Z No. | 2.3 Role | 2.4 Review and ApprovalSignature and Date |
|  |  | SW Tester | SW Designer | SU RLM | SW Owner | SRLM | FDAR |  |
|  |  |[ ] [ ] [ ] [ ] [ ] [ ]   |
|  |  |[ ] [ ] [ ] [ ] [ ] [ ]   |
|  |  |[ ] [ ] [ ] [ ] [ ] [ ]   |
|  |  |[ ] [ ] [ ] [ ] [ ] [ ]   |
|  |  |[ ] [ ] [ ] [ ] [ ] [ ]   |
|  |  |[ ] [ ] [ ] [ ] [ ] [ ]   |

|  |
| --- |
| **3.0 REVISION HISTORY** |
| 3.1 Rev. | 3.2 Date | 3.3 SWTP Revision Description and Reason(s) for Revision |
| 0 |  | Initial issue. |
|  |  |  |

|  |
| --- |
| **4.0 purpose & scope** |
| The purpose of this test plan is to demonstrate adherence of the software to documented and approved software requirements through software reviews and testing. SQA requirements for reviews and testing are applied using a graded approach based upon the software designation and software risk level (SRL) in Section 10. Test cases in this Software Verification and Validation (V&V) Test Plan (SWTP) plan are based upon and traceable to documented and approved software requirements. Verification testing (for developed software) is based upon technical (design) requirements and ensures that the software conforms to those requirements. Validation, also known as acceptance testing (required for all software), is based upon business (system) requirements and includes user acceptance testing. This Software V&V Test Plan (SWTP) and related documentation are designed to ensure that the software:1. correctly performs all its intended functions; and
2. does not perform any adverse, unintended function.
 |

|  |
| --- |
| **5.0 SOFTWARE TEST PLAN PREREQUISITE requirements CHECKLIST** |
| Development of this Software Test Plan (SWTP) incorporates the following SQA prerequisite requirements and ensures that they have been met for this software prior to software V&V testing. |
|  5.1 Software Organization |
| [ ]  | Software reviews and testing plans are based upon functional roles and responsibilities. |
|  5.2 Software (Project) Risk Management |
| [ ]  | [ ]  n/a | Software project risks have been identified, as applicable. |
|  5.3 Software Engineering Method (developed software only) |
| [ ]  | [ ]  n/a | Control points and reviews are described in the software engineering method. |
|  5.4 Standards & Conventions |
| [ ]  |  Standards, conventions, and work practices have been used in software documentation. |
|  5.5 Support Software |
| [ ]  | Support software has been graded and reviewed, tested, accepted for use, and managed under configuration control, commensurate with its grading. |
| [ ]  | Changes to support software have been evaluated for impact on the software program to determine the level of reviews and testing required. |
|  5.6 Training & Qualification |
| [ ]  | Training of software testing personnel is commensurate with the scope, complexity, and importance of the testing. |
| [ ]  | Evidence of training and/or qualification is documented. |
|  5.7 Baseline Documents & QA Records |
| [ ]  | Software baseline documents have been maintained under document control. |
| [ ]  | Software quality records have been produced and stored as required. |
|  5.8 Software Configuration Management (SCM) |
| [ ]  | All software configuration items have been maintained under SCM. |
|  5.9 Baseline Identification |
| [ ]  | A baseline labeling system is established to uniquely identify each configuration item. |
| [ ] [ ] [ ] [ ]  | The following software configuration items are identified in the software baseline:* computer programs (e.g., source code, object files, executables, backup files)
* documentation (e.g., software design requirements, test plans and results, user guides)
* data pertaining to the operation of the computer system (e.g., settings, program variables, configuration data, etc.)
* support software (i.e., software tools and system software)
 |
| [ ]  | The software baseline reflects the most recently approved software configuration. |
|  5.10 Change Control |
| [ ]  | Software changes have been documented, evaluated, and approved. |
| [ ]  | Change control includes appropriate verification activities and acceptance testing. |
| [ ]  | Change control includes updates to corresponding software baseline documentation. |
| [ ]  | Changes to software configuration items are controlled until they are incorporated into the approved baseline. |
|  5.11 Status Control |
| [ ]  | Configuration status controls provide for notification of the software change information to affected organizations. |
|  5.12 Software Requirements Identification |
| [ ]  | Software requirements are documented, managed, and traceable to test cases. |
|  5.13 Safety Analysis and Safety Design Methods (developed safety software only) |
| [ ]  | [ ]  n/a | Safety software design includes measures to mitigate the consequences of potential problems as identified in the software safety hazard analysis. |
|  5.14 Design Documentation (developed software only) |
| [ ]  | [ ]  n/a | Software design documentation is structured in a manner that facilitates software verification (i.e., traceable to verification reviews and testing). |
|  5.15 Design Reviews (developed software only) |
| [ ]  | [ ]  n/a | Software design reviews provide adequate confidence that the software performs all intended functions, provides correct solutions, and does not perform or cause any adverse unintended functions; ensures the software design meets the requirements. |

|  |
| --- |
| **6.0 V&V Test plan** |
| This V&V test plan and associated test cases demonstrate adherence to documented and approved software requirements contained in the following Software Design Requirements Document(s). |
| 6.1 Requirements Document Name and Rev. | 6.2 Attached, or in file location |
| *< requirements document name and version >* | [ ]  | *< file location, if not attached to test plan >* |
| *< requirements document name and version >* | [ ]  | *< file location, if not attached to test plan >* |
| 6.3 Software **used in design activities**, if applicable.  |
| [ ]  n/a | [ ]  | Test plan assures that the software produces correct results. Expected results and pass/fail criteria are specified in test cases. |
| 6.4 Software **used for operational control**, if applicable. |
| [ ]  n/a | [ ]  | Test plan demonstrates required performance over the range of operation of the controlled function or process. Expected results and pass/fail criteria are specified in test cases. |
| 6.5 Test plan provides for evaluating technical adequacy through comparison of test results from alternative methods, as applicable. Alternative methods used in this test plan include: |
| [ ]   | Hand calculations used for comparison of test results are specified in test cases, as applicable. |
| [ ]  | Calculations using comparable proven programs or empirical data | *< description of program(s) and/or empirical data used for comparison with test results not specified in test cases >* |
| [ ]  | Information from technical literature or other sources | *< list of literature or other source(s) used >* |
| [ ]  | Other method | *< description of other method(s) used for test result comparison >* |
| 6.6 Required tests (e.g., test cases) and test sequence. |
| *< list of required test plans and test cases including description of test sequence >* |
| 6.7 Anticipated output values. |
| *< description of where anticipated output values are included in test cases >* |
| 6.8 Acceptance criteria. |
| *< description of the overall test acceptance criteria (e.g., which test cases must pass, user test cases required to accept the software for use, etc. >* |
| 6.9 Test reports and test records. |
| *< list of associated test reports/records (formatted using standard formatting and conventions >* |
| 6.10 Required ranges of input parameters, as applicable. |
| [ ]  n/a | Ranges of input parameters and pass/fail criteria are specified in test cases. |
| 6.11 Identification of the stages at which testing is required, as applicable. |
| [ ]  n/a |  *< description of the testing stages >* |
| 6.12 Criteria used for establishing test cases, as applicable. |
| [ ]  n/a | *< list or description of criteria used for creating the test cases >* |
| 6.13 Requirements for testing logic branches, as applicable. |
| [ ]  n/a | *< list or description of requirements for testing logic branches >* |
| 6.14 Requirements for hardware integration, as applicable. |
| [ ]  n/a | *< list or description of requirements for* *hardware integration >* |
| 6.15 Characteristics to be tested and test methods employed. |
| *< list and description of characteristics to be tested and test methods used >* |
| 6.16 Test requirements and acceptance criteria as provided and approved by the responsible design organization. |
| *< description of test requirements and acceptance criteria provided and approved by the design organization >* |
| 6.17 Required tests are controlled using the tools and equipment necessary to conduct the test in a manner to fulfill test requirements and acceptance criteria. |
| *< description of how required tests are controlled and conducted to fulfill test requirements >* |
| 6.18 Tests performed will obtain the necessary data with sufficient accuracy for evaluation and acceptance. |
| *< description of how tests performed will obtain the necessary data for evaluation and software acceptance >* |
| 6.19 Test requirements and acceptance criteria are based upon specified requirements contained in applicable design documents that provide approved requirements. |
| *< description of test requirements and acceptance criteria based upon approved requirements in design documents* *>* |
| 6.20 Software verification and validation test plans are maintained under document control and as a part of the software baseline under SCM. |
| *< description of how and where software verification and validation test plans are maintained under document control and as a part of the software baseline >* |

|  |
| --- |
| **7.0 Software Verification Testing (DEveloped software only)** |
| 7.1 Software design is traceable to the software design requirements. |
| *< description of traceability conventions used for correlating test cases to software design requirements >* |
| 7.2 Verification activity requirements are documented in the software engineering method. |
| *< brief description of verification activity requirements based upon the software engineering method >* |
| 7.3 Software design verification includes a review of verification test results prior to approval of the software for use. |
| *< description of who and how verification test results were reviewed >* |
| 7.4 Verification methods include one or more of the following: |
| design reviews | *< list / description of design reviews, when they were performed, and who performed them >* |
| alternate calculations (specified in test cases) | *< general description of how alternate calculations are used in test cases >* |
| tests performed during software program development | *< description of tests performed during software development >* |
| 7.5 The extent of V&V methods chosen are a function of the: |
| complexity of the software | *< description of complexity as related to V&V methods >* |
| degree of standardization | *< degree of standardization as related to V&V methods >* |
| similarity with previously proved software | *< description of similarity with other approved software and its relevance to selected V&V methods >* |
| importance to safety | *< description of importance to safety as related to V&V methods >* |
| 7.6 The following qualified individuals, or groups, (being other than those who developed and documented the software design) performed the software design verification: |
| *< list of individuals and their qualifications for software verification activities >* |
| 7.7 Software design verification results will include identification of the software design verifier. |
| *< identification of design verifier(s) and location of software verification results (test records) >* |
| 7.8 Software design verification reviews and test plans are maintained under document control and maintained as a part of the software baseline under SCM. |
| *< location of test records and software baseline documents >* |
| 7.9 Software design verification test records and acceptance documentation will be submitted as quality records in accordance with records management requirements. |
| *< location of verification test records and acceptance documentation >* |

|  |
| --- |
| **8.0 Acceptance (Validation) Testing** |
| Acceptance testing, as specified in this test plan, will demonstrate that the software:* properly handles abnormal conditions and events as well as credible failures
* does not perform adverse unintended functions
* does not degrade the system either by itself, or in combination with other functions or configuration items
 |
|  8.1 Configuration items (e.g., modified software code) is maintained under configuration change control prior to acceptance testing. |
| *< description and/or location of software program files >* |
|  8.2 Acceptance testing will be planned and performed for all software design requirements. |
| *< reference to requirements document(s) and/or traceability matrix >* |
|  8.3 Performance of the series of tests specified in this test plan will provides assurance of correct interaction between activities and proper function of individual modules. |
| *< reference to test cases and how they provide assurance of this >* |
|  8.4 Observations of unexpected or unintended results will be documented and dispositioned prior to test result approval. |
| *< reference to test case format and process for disposition of unintended test results >* |
|  8.5 Acceptance testing of changes to the software will be subjected to selective retesting (i.e., regression testing) to detect unintended adverse effects introduced during the change. |
| *< description of regression testing that will provide assurance that changes have not caused unintended adverse effects and that modified systems or components still meet specified software design requirements >* |
|  8.6 Acceptance testing will include a comprehensive acceptance test performed in the operating environment prior to use. |
| *< description of how acceptance testing was performed in the operating environment >* |
|  |
| **9.0 Test Records (REPORT)** |
| 9.1 Software test records are established and maintained as a part of the software baseline. |
| *< list and location of software test records >* |
| 9.2 Name and version of software tested | *< name and version of software being tested >* |
| 9.3 Description of system software used | *< list and/or description of system software used >* |
| 9.4 Description of computer hardware used  | *< description of computer hardware used >* |
| [ ]  n/a | 9.5 Test equipment and calibrations | *< list of test equipment and calibrations >* |
| 9.6 Date of test  | *< date when test was performed >* |
| 9.7 Name of tester or data recorder  | *< name of tester or data recorder >* |
| [ ]  n/a | 9.8 Simulation models used | *< description of simulation models used >* |
| 9.9 Testing problems (e.g., issues) | *< description of any testing problems >* |
| 9.10 Results of testing and applicability  | *< results of testing >* |
| 9.11 Action taken in connection with any deviations noted | *< actions taken for identified deviations >* |
| 9.12 Name of individual evaluating test results  | *< name of person(s) who evaluated test results >* |
| 9.13 Acceptability of test results | *< test results pass/fail >* |
| 9.14 Test results are documented and their conformance with test requirements and acceptance criteria has been evaluated. |
| *< name of person(s) who evaluated acceptance criteria >* | *< date evaluation was performed >* |
| 9.15 The results of software testing have been submitted as quality records. |
| *< location of submitted test records >* |

|  |
| --- |
| **10.0 SWTP ATTACHMENT lIST** |
| 10.1 Att. # or Letter | 10.2 Attachment Title |
| A | Software Requirements Test Case (SWTC) Results |

|  |
| --- |
| **11.0 SWTP references** |
| 11.1 Reference ID and revision | 11.2 Reference Title |
| SQMP-XXX-XX | *Software Quality Management Plan (SQMP) for < software name > Note to Ch 21 users: Ch 21 itself is the SQMP, so list SWID and SWBL instead.* |
| SWTM-XXX-XX, revision (per SWBL) | *Software Requirements Traceability Matrix (SWTM) for* *< software name >.* |
| SCMP, revision (per SWBL) | *Software Configuration Management Procedure (SCMP) for change process. Note to Ch 21 users: Ch 21 itself is the SQMP; instead, indicate whether SWCP or SWNCP form is used.* |

 **GENERAL**

This template may be used to implement the software testing requirements of [P1040](https://int.lanl.gov/policy/information-management.shtml), *Software Quality Management* or [STD-342-100](http://engstandards.lanl.gov/ESM_Chapters.shtml#esm21), *LANL Engineering Standards Manual, Chapter 21-Software,* as applicable.Use of this template is not mandatory; other templates or formats may be used.

The governing program (i.e., P1040 or Chapter 21) provides additional definitions and details to support the implementation of all applicable SQA requirements.

For additional information on software testing requirements, see [P1040](https://int.lanl.gov/policy/information-management.shtml), R11, Section 3.9 (for software governed under P1040), [STD-342-100](http://engstandards.lanl.gov/ESM_Chapters.shtml#esm21), *LANL Engineering Standards Manual, Chapter 21-Software* (for software governed under Chapter 21), or contact an SQA representative at p1040@lanl.gov.

Throughout this template, “n/a” should be entered for fields that are not applicable.

**HEADER**

* Enter the Document ID (e.g., number or letter) and Revision Number for the SWTP, per the governing document control process.
* Enter the SWTP document Effective Date.

**1.0 SOFTWARE INFORMATION**

| **Field** | **Entry Information** |
| --- | --- |
| 1.1 | Enter the name and version number of the software being tested. |
| 1.2 | Enter the software ID (SWID), as applicable. |
| 1.3 | Provide an overview of what the software does and how it is used. For consistency, a summary of the description in the SQMP may be used. |
| 1.4 | Check the box indicating the software category from Form 2033. |
| 1.5 | Check the box indicating the (a) Software Risk Level (SRL) and (b), based on its function (determination), Management Level (ML). |

**2.0 SOFTWARE TEST PLAN (SWTP) REVIEW AND APPROVAL**

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 2.1 | Enter the name of the SWTP reviewer and approver. |
| 2.2 | Enter the Z number for the reviewer. Enter “n/a” if not applicable. |
| 2.3 | Check the box for each role the reviewer is filling. One reviewer may perform multiple roles. |
|  | Roles & Responsibilities specific to software testing (User: verify the below aligns with Ch 21 SOFT‑V&V) |
| **ST** (SW Tester) | * Reviews software design requirements.
* Designs software tests in accordance with applicable SQA requirements.
* Qualified as SME on software being tested.
* Qualified to perform software testing.
* Tests software in accordance with approved test plans.
* Completes and documents in-use tests in the operating environment.
* Documents software test results in accordance with approved test plans.
 |
| **SD** (SW Designer) | * Supports software tester to design software tests in accordance with applicable SQA requirements.
* Performs an integrated hazard analysis, as applicable.
 |
| **SU RLM** (SW User RLM) | * Supports the SRLM in approving software for use.
* Supports completion of in-use tests in the operating environment.
* Participates in software acceptance testing, as applicable.
 |
| **SO** (SW Owner) | * Owns the software and supports the SRLM in complying with the requirements of Ch 21, and implementation of a compliant SQM program for the software.
* Performs Verification and Validation (V&V) on the software.
* Identifies and ensures reviews and tests are performed by competent individuals or groups other than those who developed and documented in the original software design.
* Prepares the approval for use documentation that describes the intended use and any associated limitations, access controls, etc., for using the software.
* Ensures completion of in-use tests in the operating environment.
* Supports the SRLM in approving software for use.
 |
| **SRLM** (SO Responsible Line Manager) | * Manages and maintains software project activities in accordance with Ch 21 to ensure it operates as intended.
* Ensures personnel in the software organization who are managing or working with the software are adequately trained and, as required, qualified.
* Approves software for use.
 |
| **FDAR** (Facility Design Authority Representative) | * Reviews and approves software design requirements.
* Approves software for use.
* Approves changes to the design bases prior to implementation of a change.
 |
| 2.4 | Enter the review date and approval signature of the SWTP reviewer. |

**3.0 REVISION HISTORY**

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 3.1 | As required, revise the SWTP per the governing document control procedure. Enter the SWTP revision number (or letter). Retain prior revision numbers/letters. |
| 3.2 | Enter the effective date of the SWTP revision (same as header). |
| 3.3 | Enter a summary description of the SWTP revision and the reason(s) for the revision. |

**4.0 PURPOSE & SCOPE**

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 4.0 | This section generally describes the purpose and scope of the test plan to meet the applicable SQA software review and testing requirements for the software. |

**5.0 SOFTWARE TEST PLAN PREREQUISITES CHECKLIST**

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
|  | Development of this test plan must necessarily be based upon and/or incorporate the following prerequisite requirements. Section 5.0 is a checklist of these requirements to ensure that the software is ready for testing. |
| 5.1 | **Software Organization** (ref. Ch 21, also P1040, R11 §3.4.1). Software reviews and testing must be developed and executed based upon functional responsibilities and defined levels of authority, as defined in the Software Organization. |
| 5.2 | **Software (Project) Risk Management** (ref. Ch 21 SOFT-GEN, R1 §6.0, also P1040, R11 §3.4.2). Software project risks identified in the software risk management plan must addressed in the development of software requirements, software design, software reviews, software verification test plans, and software acceptance test plans. If this requirement does not apply to the software being tested, check the n/a box. |
| 5.3 | **Software Engineering Method (developed software only)** (ref. Ch 21 SOFT-V&V R1, §4.0, 6.0; also P1040, R11 §3.4.3). For developed software, control points and associated reviews must be performed as defined in the software engineering method. If this requirement does not apply to the software being tested, check the n/a box. |
| 5.4 | **Standards & Conventions** **(developed software only)** (ref. Ch 21 SOFT-V&V R1, §4.0, 6.0; also P1040, R11 §3.4.4). For developed software, standards, conventions, and other required work practices must be documented in software design and testing documentation, as applicable. |
| 5.5 | **Support Software** (ref. Ch 21 SOFT-V&V; also P1040, R11 §3.4.5, §3.4.5.a, §3.4.5.b). Changes to support software must be evaluated for impact on the software product to determine the level of reviews and testing required. |
| 5.6 | **Training & Qualification** (ref. Ch. 21 SOFT-GEN and SWDS; also P1040, R11 §3.4.6). Indoctrination and training of software testing personnel must be commensurate with the scope, complexity, and importance of the testing. Evidence of training and/or qualification must be maintained (i.e., documented). |
| 5.7 | **Baseline Documents & QA Records** (ref. Ch. 21 SOFT-GEN and SWBL; also P1040, R11 §3.4.7, §3.4.7.a, §3.4.7.b). Software baseline documents must be maintained under document control as identified and described in the SQMP and/or SCM procedure.Software quality records (e.g., software testing documentation) must produced and stored as identified and required by the SQMP and/or SCM procedure. |
| 5.8 | **Software Configuration Management (SCM)** (ref. Ch. 21 SOFT-GEN and SWBL; also P1040, R11 §3.5). Software change control must include appropriate verification activities to ensure that changes are implemented correctly and function as intended.  |
| 5.9 | ***Baseline Identification*** (ref. Ch. 21 SOFT-GEN and SWBL; also P1040, R11 §3.5.1). The software baseline must identify and describe all the software configuration items to be managed. The software baseline must be established upon completion of each activity (e.g., testing) in the software design process. A baseline labeling system (e.g., file naming, version numbering) must be established to uniquely identify each configuration item and identify changes to these items. The baseline labeling system must be maintained and remain consistent throughout the software life cycle. The following software configuration items must be identified in the software baseline: * computer programs (e.g., source code, object files, executables, backup files)
* documentation (e.g., SWRS, SWDD, test plans and results, user guides)
* data pertaining to the operation of the software (e.g., settings, program variables, configuration data, etc.)
* support software (i.e., software tools and system software)
 |
| 5.10 | ***Change Control*** (ref. Ch. 21 SOFT-GEN and SWBL; also P1040, R11 §3.5.2). Software changes must be documented, evaluated, and approved by the organization responsible for the original software design. Software change control documentation must include: * a description of the change
* rationale for the change
* identification of affected software baselines

Software change control must include appropriate acceptance testing to ensure that the changes correctly implement business (system) requirements. In-use testing (i.e., regression testing) must be performed after the software when there are significant changes in the operating system. The software change control process must include: * initiation, evaluation, and disposition of a change request
* control and approval of changes prior to implementation (i.e., coding or upgrade)
* traceability of changes to the software design requirements
* requirements for retesting (i.e., regression testing) and acceptance of test results
* update of the software baseline upon acceptance for use of the modified software
 |
| 5.11 | ***Status Control*** (ref. Ch. 21 SOFT-GEN and SWBL; also P1040, R11 §3.5.3). The status of configuration items (i.e., software baseline) must be maintained current. Changes to software configuration items must be controlled until they are incorporated into the approved baseline. Configuration status controls must provide for notification of the software change information to affected organizations. |
| 5.12 | **Software Requirements Identification** (ref. ref. Ch 21 SOFT-V&V; also P1040, R11 §3.6). Requirements traceability means that every software characteristic and/or function, design element, verification test, and acceptance test can be traced back to a documented and approved software requirement.Software requirements must be identified, documented, and traceable throughout the software life cycle. Software requirements must be complete, correct, consistent, clear, verifiable, and feasible.For developed software, the selection of software requirements must be reviewed and approved by the SRLM and, for safety software and risk significant software, the appropriate design authority (e.g., FDAR).Once approved, software requirements must be managed to minimize conflicting requirements and maintain accuracy for validation activities. |
| 5.13 | **Safety Analysis and Safety Design Methods** **(safety software only)** (ref. Ch 21 SOFT-V&V; also P1040, R11 §3.8.1). For safety software, safety analysis and safety design require a safety (hazard) analysis to anticipate and mitigate consequences of potential problems.The safety (hazard) analysis should include mitigation strategies for* standard hazards (i.e., basic hazards associated with the software or process),
* system failures (i.e., the functionality written into the software itself to protect from failure), and
* system overrides (i.e., the functionality written into the software to keep a user or other system from bypassing the safety features within the software item).

Safety software design must include measures to anticipate and mitigate the consequences of potential problems as identified in the software safety hazard analysis.If this requirement does not apply to the software being tested, check the n/a box. |
| 5.14 | **Design Documentation (developed software only)** (ref. Ch. 21 SOFT-DESIGN; also P1040, R11 §3.8.2). Software design documentation must be structured in a manner that facilitates software design verification (i.e., design requirements traceable to verification reviews and testing).Design documentation must describe how the software will interface with other system components and how the software will function internally.Software design documentation must define the computational sequence (i.e., program logic, flow, and supporting functions) necessary to meet the software design requirements.If this requirement does not apply to the software being tested, check the n/a box. |
| 5.15 | **Design Reviews (developed software only)** (ref. Ch. 21 SOFT-DESIGN; also P1040, R11 §3.8.4). Design reviews may be performed and documented separately or together, as appropriate, according to the defined software engineering method.The software development methodology must define the control points and associated reviews. Design reviews must ensure approved software (design) requirements are implemented.Two types of design reviews are required:1. One review must consider the requirements related to preparing the software for acceptance testing. This review can be combined with or be a part of the software design verification.
2. The other review must provide assurance of satisfactory completion of software development cycle activities including acceptance (validation) testing. This review can be combined with or be a part of the software design verification.

If this requirement does not apply to the software being tested, check the n/a box. |

**6.0 V&V TEST PLAN** (ref. Ch 21 SOFT-V&V; also P1040, R11 §3.9.1)

Software test plans are used for many kinds of software testing. Standardized test plans can be used repeatedly, particularly for regression and other in-use testing. Test plans include the testing environment, assumptions, approaches, test cases, expected results, pass/fail criteria, etc.

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 6.1 | Enter applicable software design requirements document by name including the document revision number. For more than one applicable requirements document, add lines as needed.**NOTE:** These are the business, technical, functional requirements for the software itself required for traceability to test cases, not software quality requirements for the test plan. |
| 6.2 | Use the checkbox to indicate if the requirements document is attached to the test plan. Otherwise, indicate the file location for each. |
| 6.3 | For software used in design activities, check the box to indicate that test plans will assure that the software produces correct results for design activities and that expected results and pass/fail criteria will be specified in the test cases.For software that is not used in design activities, check n/a. |
| 6.4 | For software used for operational control, check the box to indicate that test plans will demonstrate required performance over the range of operation of the controlled function or process and that expected results and pass/fail criteria will be specified in the test cases.For software that is not used for operational control, check n/a. |
| 6.5 | If alternative methods are used in this test plan and/or test cases for verification and validation, use the appropriate checkbox(es) and describe. |
| 6.6 | The required tests (i.e., test cases) and test sequence must be described. |
| 6.7 | The anticipated output values and/or point to test cases must be described, as appropriate. |
| 6.8 | Acceptance criteria for the software must be defined. These criteria must be met to approve the software for use. |
| 6.9 | List associated test reports and test records, which will be formatted using standard formatting and conventions, per the SQMP. |
| 6.10 | Describe the required ranges of input parameters and/or point to test cases. If not applicable, check n/a. |
| 6.11 | Identify the stages at which testing is required. If not applicable, check n/a. |
| 6.12 | Describe the criteria used for establishing test cases. If not applicable, check n/a. |
| 6.13 | Describe the requirements for testing logic branches. If not applicable, check n/a. |
| 6.14 | Describe the requirements for hardware integration. If not applicable, check n/a. |
| 6.15 | Describe the characteristics to be tested and test methods to be employed in this test plan and/or test cases. |
| 6.16 | Describe how the test requirements and acceptance criteria have been provided and approved by the responsible design organization. These should be indicated in the software requirements document(s) listed in Field 6.1 (above). |
| 6.17 | Describe how the required tests are controlled using the tools and equipment necessary to conduct the test in a manner to fulfill test requirements and acceptance criteria. This should point to the software configuration management process. |
| 6.18 | Describe how the tests performed will obtain the necessary data with sufficient accuracy for evaluation and acceptance. |
| 6.19 | Describe how test requirements and acceptance criteria are based upon specified requirements contained in applicable design documents or other pertinent technical documents that provide approved requirements. These are best supported by a traceability matrix. |
| 6.20 | Describe how/where the software verification and validation test plans are maintained under document control and maintained as a part of the software baseline under SCM. |

**7.0 SOFTWARE VERIFICATION REVIEWS & TESTING** (ref. Ch 21 SOFT-V&V; also P1040, R11 §3.9.2)

Software verification is accomplished through reviews and verification testing. Verification testing (also known as software design verification) evaluates the technical adequacy of the design approach to assure internal completeness, consistency, clarity, and correctness of the software design. Verification is based primarily upon technical (design) requirements.

Software design verification may be performed by the Software Designer’s supervisor so long as the supervisor:

* did not specify the design approach or establish design inputs used in the design, or
* is the only individual in the organization competent to perform the verification.

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 7.1 | Design verification must verify that the software design is traceable to the software technical (design) requirements. |
| 7.2 | Requirements for the software design verification activity must be documented in the software engineering method. |
| 7.3 | Software design verification includes a review of verification test results that must be completed prior to approval of the software for use. |
| 7.4 | Software design verification methods must include one or more of the following: * design reviews
* alternate calculations
* tests performed during computer program (code) development
 |
| 7.5 | Software design verification must ensure that: * results of the safety software V&V are documented and controlled,
* V&V methods include any one or a combination of design reviews, alternate calculations, and tests performed during program (software) development (see above), and
* the extent of V&V methods chosen are a function of the
	+ complexity of the software,
	+ degree of standardization,
	+ similarity with previously proved software, and
	+ importance to safety.
 |
| 7.6 | Software design verification must be performed by qualified individuals, or groups, other than those who developed and documented the software design. |
| 7.7 | The results of software design verification must be documented and must include identification of the software design verifier. |
| 7.8 | Software design verification reviews and test plans must be maintained under document control and maintained as a part of the software baseline under SCM. |
| 7.9 | Software design verification test records and acceptance documentation must be submitted as quality records. |

**8.0 ACCEPTANCE (VALIDATION) TESTING** (ref. Ch 21 SOFT-V&V; also P1040, R11 §3.9.3)

Acceptance testing activities must demonstrate that the computer program adequately and correctly performs intended functions (i.e., specified in software design requirements).

Acceptance testing must demonstrate that the software:

* properly handles abnormal conditions and events as well as credible failures,
* does not perform adverse unintended functions, and
* does not degrade the system either by itself, or in combination with other functions or configuration items.

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 8.1 | Configuration items must be maintained under configuration change control prior to acceptance testing.  |
| 8.2 | Acceptance testing must be planned and performed for all software design requirements.  |
| 8.3 | Performance of a series of tests must provide assurance of correct translation between activities and proper function of individual modules.  |
| 8.4 | Observations of unexpected or unintended results must be documented and dispositioned prior to test result approval.  |
| 8.5 | Acceptance testing of changes to the software must be subjected to selective retesting (i.e., regression testing) to detect unintended adverse effects introduced during the change. Such testing must provide assurance that changes have not caused unintended adverse effects in the software, and to verify that modified systems or components still meet specified software design requirements.  |
| 8.6 | Acceptance testing must include a comprehensive acceptance test performed in the operating environment prior to use.  |

**9.0 TEST RECORDS (REPORTS)** (ref. Ch 21 SOFT-V&V; also P1040, R11 §3.9.1.b)

Test records are the results of software testing. Test plans are typically used as templates for test reports. Software test records must be established and maintained as a part of the software baseline to indicate the ability of the computer program to perform its intended function and meet its documented requirements. Test reports must include the following elements, as applicable.

|  |  |
| --- | --- |
| **Field** | **Entry Information** |
| 9.1 | List and location of software test records. |
| Software test records may vary depending on the test type, purpose, and application, but will contain the following information, as a minimum:  |
| 9.2 | Name and version of software tested |
| 9.3 | Software tested including system software used |
| 9.4 | Computer hardware used |
| 9.5 | Test equipment and calibrations. If not applicable, check n/a. |
| 9.6 | Date of test |
| 9.7 | Name of tester or data recorder |
| 9.8 | Simulation models used. If not applicable, check n/a. |
| 9.9 | Test problems |
| 9.10 | Results and applicability |
| 9.11 | Action taken in connection with any deviations noted |
| 9.12 | Name of person evaluating test results |
| 9.13 | Acceptability of results pass/fail |
| 9.14 | Test results must be documented and their conformance with test requirements and acceptance criteria must be evaluated.  |
| 9.15 | The results of software testing must be documented and submitted as quality records. |

**10.0 SWTP ATTACHMENT LIST**

| **Field** | **Entry Information** |
| --- | --- |
|  10.1 | List of attachments as appropriate. Enter the attachment number or letter. |
| 10.2 | Provide attachment title. |

**11.0 SWTP REFERENCES**

| **Field** | **Entry Information** |
| --- | --- |
| 11.1 | List of references as appropriate. Include the SQMP (if using ESM Ch. 21, the Software Data Sheet [SWDS]). Enter the reference number and revision. |
| 11.2 | Provide title of reference. |