

Software Risk Checklist - Taxonomy

The following is a software risk checklist. It is organized by development phases of a project, with emphasis on the software portion of the overall project lifecycle. Listed here are some, not an exhaustive list, of the generic risks that should be considered when any project contains software. This checklist contains practical questions that were gathered by experienced NASA engineers and is not a part of the SEI course or guidebook. The SEI has their own taxonomy-based questionnaire that should be considered during any risk assessment (SEI *Continuous Risk Management Guidebook* chapters A-32 to A-34, pg. 471-509 and Taxonomy - Based Risk Identification).

The project manager, software manager, system engineer/manager, any software technical leads, and the software engineers, as a minimum, should review, fill out, and discuss the results of this checklist. Taking into account all the different perspectives and adding risks specific to a project, the review team should then meet to create an agreed-upon set of risks and start planning how they will be addressed. This checklist is only an aid to start the managers and engineers thinking and planning how to realize, avoid, mitigate and accept the risks inherent in any software project. The first step to controlling a project is understanding where it may go out of control and plan to avoid it as much as is possible. As this risk checklist covers many lifecycle stages, it is suggested that this checklist initially be used during systems requirements to establish a baseline risk assessment. At that time, the entire risk checklist should be gone through and an initial risk assessment should be generated. These risks can then be documented in a risk database and/or a risk mitigation plan. Once this initial baseline risk assessment has been created, the project should revisit the risk checklist during each subsequent lifecycle stage in order to see if new risks have been discovered or if issues not previously understood to be a risk now need to be elevated to a risk. If the project is using rapid prototyping, the spiral lifecycle, or some other iterative lifecycle, then period at which the list will be revisited should be established at the beginning of the project and followed throughout. The software management plan or software risk management plan would be the appropriate place to document the entire risk approach, schedule and process.

The checklist is laid out with the generic risks listed followed by a column to indicate if this is a risk for a particular project. **Yes**, this is a risk; **No**, not a risk for this project at this time; **Partially** a problem as stated, further clarification should be added. The last column is to indicate if this risk should be accepted or needs to be worked, i.e. the risk needs to be researched, mitigated, or watched. (See the SEI *Continuous Risk Management Guidebook* page 63.)

Remember, this checklist is not an exhaustive list of all possible generic risks. It is meant to generate ideas and is not meant to be a complete list of all potential risks that could be considered. The user should consider the checklist, along with the Taxonomy Based Questionnaire provided in the SEI *Continuous Risk Management Guidebook* (Chapters A-32 to A-34, pages 471-509), as a basis for starting to examine possible risks on a project. The risk checklist should be added to, and tailored, to fit a project/program's needs. Sometimes the wording on the questions contained in the

checklist are open-ended in order to get the project team to think beyond what is written.

Also remember, not all risks are technical. Development environment, schedule, resources, etc. all have risks that need to be considered.

System Requirements Phase	<u>RISK</u> Yes/No /Partial	<u>ACTION</u> Accept/ Work
Are system-level requirements documented? To what level? Are they clear, unambiguous, verifiable ?		
Is there a project-wide method for dealing with future requirements changes?		
Have software requirements been clearly delineated/allocated?		
Have these system-level software requirements been reviewed, inspected with system engineers, hardware engineers, and the users to insure clarity and completeness?		
Have firmware and software been differentiated; who is in charge of what and is there good coordination if H/W is doing "F/W"?		
Are the effects on command latency and its ramifications on controllability known?		
Is an impact analysis conducted for all changes to baseline requirements?		

Software Planning Phase	<u>RISK</u>	<u>ACTION</u>
Is there clarity of desired end product? Do the customer & builders (system and software) agree on what is to be built and what software's role is?		
Are system-level requirements on software documented? Are they complete/sufficient and clearly understood?		
Are all interface requirements known & understood?		
Are roles and responsibilities for system & software clearly defined and followed and sufficient?		
Have the end user/operator requirements been represented in the concept phase such that their requirements are flowed into the software requirements?		
Has all needed equipment, including spares, been laid out? and ordered? Is there sufficient lead time to get needed equipment? Is there a contingency plan for not getting all equipment? Is there a contingency plan for not getting all equipment when needed?		
Is the needed level of technical expertise known?		
Is the level of expertise for software language, lifecycle, development methodology (Formal Methods, Object Oriented, etc.), equipment (new technology), etc. available: within NASA? from contractors? Will expertise be available as the schedule demands? Is there more than one person with a particular expertise/knowledge (i.e. is too much expertise held by only one team member? What if they quit, or get sick?)		
Training: Is there enough trained personnel? Is there enough time to train all personnel? on the project itself? on equipment/ software development environment, etc.? Will there be time and resources to train additional personnel as needed?		
Budget: Is the budget sufficient for: equipment? needed personnel? training? travel? etc.		

Software Planning Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Schedule: Is the schedule reasonable considering needed personnel, training, and equipment? Does the system-level schedule accommodate software lifecycle? Can needed equipment be made available in time?		
Has all the slack/contingency time on the critical path been used up?		
Are software metrics kept and reported regularly? Weekly? Monthly?		
Are deviations to the development plan being tracked? Trended? Are the trends reported in a manner to allow timely and appropriate software and project management decisions?		
Will new development techniques be used?		
Will a new or different development environment be used?		
Is this a new technology?		
Will simulators need to be designed and built? Is there time and resources allocated for this?		
Is there a schedule that covers development of both ground and flight software? Is it reasonable, does it match reality? Is it being followed? Are changes tracked and the reasons for the changes well understood?		
Do the schedules for ground and flight software match with what is needed for test and delivery?		
Are there separate schedules for flight and ground? Are different people in charge of them? Are they coordinated by some method?		
Will test software need to be designed and developed? Are there time and resources allocated for this?		
Distributed development environment: Will this be a distributed development (different groups or individuals working on parts of the project in different locations e.g. out of state)? Are there proper facilities and management structure to support distributed development?		
Inter/Intra group management: Are interfaces with other developers, suppliers, users, management, and the customer understood and documented? Is there a known way to resolve differences between these groups (i.e., conflict resolution/ who has ultimate authority, who is willing to make a decision)?		

Software Planning Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Management Planning: Is management experienced at managing this size and/or type of team? (Is there an experienced project manager?) Is management familiar with the technology being used (e.g., Formal Methods, OOA/OOD and C++)? Is there a well-constructed software management plan that outlines procedures, deliverables, risk, lifecycle, budget, etc. Is it reasonable, does it match reality? Is it being followed?		
Does software lifecycle approach & timeframe meet needs of overall project; does it have a chance of being close to what is needed?		
Has time been allotted for safety analysis and input?		
Has time been allocated for reliability analysis (e.g., Failure Modes and Effects Analysis (FMEA), Critical Items List (CIL), Fault Tolerance Analysis) input?		
Has time been allocated for software (s/w) quality analysis input and auditing?		
Have software development standards & processes been chosen?		
Have software documentation standards been chosen?		
Has Software Product Assurance given input on all standards, procedures, guidelines, and processes?		
Is funding likely to change from originally projected? Is there a plan in place to handle possible funding changes? Prioritization of requirements? Phasing of requirements delivery?		
Is there a procedure/process for handling changes in requirements? Is it sufficient?		
Examine detailed technical considerations such as: Can the bus bandwidth support projected data packet transfers? Are system requirements defined for loss of power? Is the system reaction to loss of power to the computers known or planned for? Have UPS (Uninterruptable Power Supplies) been planned for critical components?		

Software Requirements Phase	<u>RISK</u>	<u>ACTION</u>
<p>Software schedule:</p> <p>Is there an adequate software schedule in place?</p> <p>Is it being followed?</p> <p>Are changes to schedule being tracked?</p> <p>Are changes to the schedule made according to a planned process?</p> <p>As events change the schedule, is the decision process for updating the schedule also examined? That is, question if there is something wrong in the process or program that needs to change in order to either make schedule or affect the schedule-updating process?</p> <p>Has the overall schedule been chosen to meet the needs of true software development for this project or has the software schedule merely been worked backwards from a systems NEED date with no consideration for implementation of recommended software development process needs?</p>		
Has all the slack/contingency time on the critical path been used up?		
Are software metrics kept and reported regularly? Weekly? Monthly?		
Are deviations to the development plan being tracked? Trended?		
Are the trends reported in a manner to allow timely and appropriate software and project management decisions?		
<p>Are parent documents baselined before child documents are reviewed?</p> <p>Is there a process in place for assessing the impact of changes to parent documents on child documents?</p> <p>Is there a process in place for assessing the impact of changes to parent documents from changes within child documents?</p>		
Are review/inspection activities and schedules well defined and coordinated with sufficient lead time for reviewers to review material prior to reviews/inspections?		
Is there a process for closing out all TBDs (to be determined) before their uncertainty can adversely affect the progress of the project?		
<p>Have all the software-related requirements from the systems-level requirements been flowed down?</p> <p>Have the system level and software level standards been chosen?</p> <p>Have the requirements from these standards been flowed down from the system level?</p> <p>Have guidelines, etc., been established?</p>		

Software Requirements Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Has the project planned how to handle changing requirements? Compartmentalized design? Are procedures/change boards in place for accepting/rejecting proposed changes Are procedures in place for dealing with schedule impacts due to changes? Is the project following these procedures? Is there good communication with the principle investigators/customer? Have requirements been prioritized? Is this prioritization tracked, reviewed, and periodically updated? Is there a clear understanding of what is really necessary for this project?		
Have there been changes/reductions in personnel since first estimates?		
Are there sufficient trained software personnel? Does all the knowledge for any aspect of project reside in just one individual?		
Is there a software testing/verification plan?		
Is the software management plan being followed? Does it need to be adjusted?		
Is the software development environment chosen and in place?		
Does work contracted out have sufficient controls and detail to assure quality, schedule, and meeting of requirements?		
Is a Software Configuration Management (SCM) Plan in place and working?		
Are backups of SCM system/database planned and carried out on a regular basis?		
Are inspections or peer reviews scheduled and taking place?		
Software Quality/Product Assurance (SQA or SPA): Is SPA working with development to incorporate safety, reliability and QA requirements? Is s/w development working with SPA to help establish software processes? Does SPA have a software-auditing process and plan in place?		
Are there good lines of communication established and working between software project groups?		

Software Requirements Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
<p>Are good lines of communication established and working with groups outside software development?</p> <p>Are there written agreements on how to communicate?</p> <p>Are they followed?</p> <p>Are they supported by management and systems group?</p> <p>Are there good interface documents detailing what is expected?</p> <p>Did all the concerned parties have a chance to review and agree to them?</p>		
<p>Have resources been re-evaluated (equipment, personnel, training, etc.)?</p> <p>Are they still sufficient?</p> <p>If not, are steps being taken to adjust project schedule, budget, deliverables, etc. (more personnel, re-prioritization and reduction of requirements, order new equipment, follow previously established mitigation plan, etc.)?</p>		
<p>Are COTS being used?</p> <p>How are COTS maintained? Who owns and who updates them?</p> <p>Is the product affected by changes to COTS?</p> <p>Will new releases of one or more COTS be maintained/supported?</p> <p>Are COTS releases coordinated with the developed software maintenance and releases?</p> <p>Do COTS meet the necessary delivery schedule?</p> <p>Do personnel have a good understanding of how to use/integrate COTS into final product?</p> <p>If the COTS incorporated into the system meet only a subset of requirements of the overall requirements (that is, the COTS software does not completely fulfill the system requirements) , have the integration task and time been correctly estimated for merging the COTS with any in-house or contracted software that is needed to complete the requirements? Can this integration task be estimated?</p> <p>Will custom software need to be written to either get different COTS to interact correctly or to interact with the rest of the system as built or planned?</p>		
<p>Is a new technology/methodology being incorporated into software development? Analysis? Design? Implementation? (e.g., Formal Methods, Object Oriented Requirements Analysis, etc.)</p> <p>Has the impact on schedule, budget, training, personnel, current processes been assessed and weighed?</p> <p>Is there process change management in place?</p>		

Software Requirements Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Is a new technology being considered for the system? Has the impact on schedule, budget, training, personnel, current processes been assessed and weighed? Is there process change management in place?		
Is the project planning to do prototyping of unknown/uncertain areas to find out if there are additional requirements, equipment, and/or design criteria that may not be able to be met.		

Software Design Phase	<u>RISK</u>	<u>ACTION</u>
Is the software management plan being followed? Does it need to be updated?		
Is the requirements flow-down well understood?		
Are standards and guidelines sufficient to produce clear, consistent design and code?		
Will there be, has there been, a major loss of personnel (or loss of critical personnel)?		
Is communication between systems and other groups (avionics, fluids, operations, ground software, testing, QA, etc.) and software working well in both directions?		
Requirements: Have they been baselined & are they configuration managed? Is it known who is in charge of them? Is there a clear, traced, managed way to implement changes to the requirements? (i.e., is there a mechanism for inputting new requirements, or for altering old requirements, in place and working)? Is there sufficient communication between those creating & maintaining requirements and those designing to them? Is there a traceability matrix between requirements and design? Does that traceability matrix show the link from requirements to design and then to the appropriate test procedures?		
Has System Safety assessed software? Does any software involved hazard reports? Does software have the s/w subsystem hazard analysis? Do software personnel know how to address safety-critical functions, how to design to mitigate safety risk? Are there fault detection, isolation, and recovery (FDIR) techniques designed for critical software functions?		
Has software reliability been designed for? What level of fault tolerance has been built in to various portions /functions of software?		
Is there a need to create simulators to test software? Were these simulators planned for in the schedule? Are there sufficient resources to create, verify and run them? How heavily does software completion rely on simulators? How valid/accurate (close to the flight unit) are the simulators?		

Software Design Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Are simulators kept up-to-date with changing flight H/W?		
How heavily does hardware completion rely on simulators?		
Is firmware and/or any other software developed outside the software flight group ? Is it being integrated? Is it being kept current based on changes to requirements & design? Is it configuration managed?		
Does work contracted out have sufficient controls and detail to assure quality, schedule, and meeting of requirements?		
Will design interfaces match in-house or other contracted work?		
Is a software configuration management plan in place and working?		
Are backups of SCM system/database planned and carried out on a regular basis?		
Are Inspections and/or peer reviews scheduled and taking place?		
Software Quality/Product Assurance (SQA or SPA): Is SPA working with development to incorporate safety, reliability, and QA requirements into design? Does SPA have a software-auditing process and plan in place? Have they been using it?		
Are parent documents baselined before child documents are reviewed? Is there a process in place for assessing the impact of changes to parent documents on child documents? Is there a process in place for assessing the impact of changes to parent documents from changes within child documents?		
Are review/inspection activities and schedules well defined and coordinated with sufficient lead time for reviewers to review material prior to reviews/inspections?		
Has all the slack/contingency time on the critical path been used up?		
Are software metrics kept and reported regularly? Weekly? Monthly?		
Are deviations to the development plan being tracked? Trended? Are the trends reported in a manner to allow timely and appropriate software and project management decisions?		

Software Implementation Phase	<u>RISK</u>	<u>ACTION</u>
Coding and unit test		
Is the software management plan still being used?		
Is it up-to-date?		
Are there coding standards?		
Are they being used?		
Are software development folders (SDFs) being used to capture design and implementation ideas as well as unit test procedures & results?		
Are code walk-throughs and/or inspections being used?		
Are they effective as implemented?		
Is SQA/SPA auditing development processes and SDFs?		
Is the design well understood and documented?		
Are requirements being flowed down through design properly?		
Is the schedule being maintained?		
Have impacts been accounted for (technical, resources, etc.)?		
Is it still reasonable?		
Has all the slack/contingency time on the critical path been used up?		
Are software metrics kept and reported regularly? Weekly? Monthly?		
Are deviations to the development plan being tracked? Trended?		
Are the trends reported in a manner to allow timely and appropriate software and project management decisions?		
Have any coding requirements for safety-critical code been established?		
If so, are they being used?		
Does the chosen development environment meet flight standards/needs?		
Has System Safety assessed software (subsystem safety analysis)?		
Has software reviewed this safety assessment?		
Has software had input to this safety assessment?		
Do software personnel know how to address safety critical functions?		
Is software working with systems to find the best solution to any hazards?		
Has FDIR (fault detection, isolation, and recovery) and/or fault tolerance been left up to implementers (i.e., no hard requirements and/or no design for these)?		
Is there a known recourse/procedure for design changes?		
Is it understood?		
Is it used?		
Does it take into account changes to parent documents?		
Does it take into account subsequent changes to child documents?		

Software Implementation Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Coding and unit test (cont.)		
Is there a known recourse/procedure for requirements changes? Is it understood ? Is it used? Is it adequate; does it need to be altered? Does it take into account changes to parent documents? Does it take into account subsequent changes to child documents?		
Is there development level Software Configuration Management (SCM) (for tracking unbaselined changes and progress)? Is it being used by all developers, regularly? Are backups performed automatically on a regular basis?		
Is there formal SCM and baselining of requirements and design changes?		
Are the design documents baselined?		
Are the requirements baselined?		
Have test procedures been written and approved? Are they of sufficient detail? Will these tests be used for acceptance testing of the system? Are these procedures under SCM? Are they baselined?		
Do some software requirements need to be tested at the systems level for complete verification? Are these documented? Do the systems-level test procedures adequately cover these? Does the requirements/verification matrix indicate which requirements are tested at the systems level?		
For subsystem-level testing: Has software been officially accepted by the subsystems (sign-off, baselined)? Are software testing facilities maintained for any regression testing?		
Are unit testing procedures and results maintained via SCM?		
Is there auto-generated code?		
Is unit testing planned for auto-generated code? Are there procedures for testing unit level auto-generated code?		
Are implementation personnel familiar with the development environment, language, and tools? Sufficiently trained coders (e.g., understand OOA, OOD, C++, Formal Methods, etc., whatever is needed)? Sufficient level of expertise (not first or second time ever done, not just trained)?		

Software Implementation Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Coding and unit test (cont.)		
Are coders sufficiently familiar with project function/design?		
Do coders have ready access to someone with sufficient expertise and whose time is available for participation in code walk-throughs or inspections and for technical questions?		
Is there sufficient equipment?		
Are there build procedures? Are they documented? Are they under SCM? Are they being followed?		
Are there burn procedures for any PROMS? ROMS? EPROMS? Are they documented? Are they under SCM? Are they being followed? Do they include a method for clearing PROMs (if applicable) and checking them for defects prior to burning? Does the procedure include a method to determine and recording the checksum(s)?		
Are test plans complete? Is further testing needed? Unit level testing? CSCI level testing? Integration testing CSCIs? System-level testing?		
Is the test/requirements matrix up to date?		

Software Implementation Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Integration and Systems Testing		
Are review activities and schedules well defined and coordinated?		
Is there a sufficient number of experienced test personnel? Who are experienced on similar projects? Who are experienced with this project? Who are experienced with test equipment, set-up, simulators, hardware? Who are experienced with development environment?		
Is the software test plan being followed? Does it need to be modified? Does it include COTS? Does it include auto-generated code?		
Are there well-written, comprehensive test procedures? Are they up to date? Do they indicate the pass/fail criteria? Do they indicate level of regression testing?		
Are test reports written at the time of the tests?		
Are test reports witnessed and signed off by SPA?		
Is the test/requirements matrix up to date?		
Is there a known recourse/procedure for testing procedure changes? (i.e., is there an Software Configuration Management Process that covers the test procedures?) Is it understood? Is it used? Does it take into account possible changes to parent documents of the test plan or other parent documents? Does it take into account subsequent changes to child documents? Does it take into account regression testing?		
Is there a known recourse/procedure for requirements changes? Is it understood? Is it used? Is it adequate, does it need to be altered? Does it take into account changes to parent documents (e.g., systems requirements)? Does it take into account subsequent changes to child documents (e.g., design and testing documents)?		
Is there Software Configuration Management (SCM) (for tracking baselined changes and progress)? Is it being used? Are backups performed automatically on a regular basis?		

Software Implementation Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Integration and Systems Testing (cont.)		
Is there formal SCM and baselining of requirements and design changes?		
Are the design documents formally baselined and in SCM?		
Are the software requirements formally baselined?		
Have test procedures been written and approved? Are they of sufficient detail? Do they exist for unit test? Do they exist for CSCI level testing Do they exist for CSCI integration-level testing? Do they exist for software system-level testing? Will these tests be used for acceptance testing to the system? Are these procedures in SCM? Are they baselined?		
Do some software requirements need to be tested at the systems level for complete verification? Are these requirements verification procedures documented? Where are they documented? In software test procedures? In systems test procedures? Do the systems-level test procedures adequately cover these? Does the requirements/verification matrix indicate which requirements are tested at the systems level?		
For system-level testing: Has software been officially accepted by systems (sign-off, baselined)? Are software testing facilities maintained for any regression testing?		
Is firmware ready and tested? Is it baselined and in SCM?		
Are there separate test personnel that have not been designers or coders scheduled to perform the tests? Do they need training? Is time allowed for their unfamiliarity with the system?		
On the flip side, are testers too familiar with software? Will they have a tendency to brush over problems or fix problems without going through proper channels/procedures?		
Have requirements/design/code personnel been moved to other tasks and are no longer available to support testing or error correction?		
Are test pass/fail criteria known and understood?		

Software Implementation Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Integration and Systems Testing (cont.)		
<p>Is regression testing planned for? Is there time in the schedule for it? Have estimates been made at each test point of the amount of regression testing necessary to cover fixes if test fails? (e.g., certain failures require complete (end-to-end) re-testing, others may require only re-testing of that test point.)</p>		
<p>Is ground software (or other related software) available for testing or for use in testing flight s/w?</p>		
<p>Has testing of COTS at the software system level been adequately covered and documented? Are there test procedures specifically for proving integration of COTS? Does the requirements to test matrix indicate where COTS is involved?</p>		
<p>Has testing of COTS at the system level been adequately covered and documented?</p>		
<p>Is there good configuration management in place? Is it used? Is there version control? Is error/failure tracking in place? Are PRACA (Problem Report and Corrective Action) and/or s/w change records created? Are problem/change records tracked to closure? Is error correction written into each new release of a module (in code comments, in file header, in SCM version description)? Are incorporated PRACAs listed in the build release version descriptions?</p>		
<p>Will a tight schedule cause: Dropping some tests? Incomplete regression testing? Dropping some fixes? Insufficient time to address major (or minor) design and/or requirements changes? No end-to-end testing?</p> <p>Are these issues being addressed? Who makes these decisions? The change control board? How are they recorded? Does the version description document (VDD) indicate true state of delivered software?</p>		

Software Implementation Phase (cont.)	<u>RISK</u>	<u>ACTION</u>
Integration and Systems Testing (cont.)		
Has all the slack/contingency time on the critical path been used up?		
Are software metrics kept and reported regularly? Weekly? Monthly?		
Are deviations to the development plan being tracked? Trended? Are the trends reported in a manner to allow timely and appropriate software and project management decisions?		

Acceptance Testing and Release	<u>RISK</u>	<u>ACTION</u>
Has pre-ship review already taken place?		
Is actual flight equipment available for software testing? Do the logbook and test procedures record actual flight hardware used for testing?		
Are pass/fail criteria established and followed?		
Is a regression testing procedure documented and known? Is it used?		
Is the procedure to handle PRACAs (Problem Report and Corrective Action) at the acceptance level documented? Is there a change review board in place? Has there been configuration management of changes? Is the PRACA/SPCR (S/W Problem and Change Request) log maintained with status?		
Is systems-level testing adequate to insure software requirements or some software-level testing done separately and documented?		
Is appropriate personnel witness and sign-off testing? SPA or QA involved?		
Are all parts of the architecture verified on the ground prior to flight?		
Does a complete VDD (Version Description Document) exist? In the VDD, are: All delivered software release versions listed? All COTS and their versions listed? All hardware versions appropriate for this release noted? SCM release description(s) provided? Build procedures given? Burn procedures given? Installation procedures provided? List of all incorporated (closed) problem reports and change requests included? List of all outstanding problem reports and change requests included? List of any known bugs and the work-arounds provided? Changes since last formal release indicated? List of all documentation that applies to this release, and its correct version, provided? If there are known discrepancies to hardware, documentation, etc. are these listed and discussed in the VDD?		
Is there clean customer hand-off: Up to date documentation? User/Operations Manual? Code Configuration Managed? All PRACAs & SPCR closed?		

Acceptance Testing and Release (cont.)	<u>RISK</u>	<u>ACTION</u>
Is there good configuration management wrap-up: Is there a method for future updates/changes in place? Proper off-site storage of data, software and documentation? What happens to SCM and data when project is over?		