

Condensed GSAM Handbook Checklists

I. Project Management Checklist (Chapter 1)

This checklist is provided to guide you in essential actions to ensure your project is on track in meeting cost, schedule, and performance requirements. If you cannot check an item off as affirmative, you need to either rectify the situation or develop a contingency plan to solve problems that may arise. For example, if the staff does not have sufficient technical skill to do the work, you will need to remedy the situation by providing training, or by obtaining sufficiently skilled people.

Beginning a Project

- The project has specific goals to accomplish and you understand the reasoning behind them.
- All stakeholders (interested parties) understand and agree on the expected project outcomes.
- Upper management is solidly behind the project.
- You understand the level of authority you have been granted in relation to the project and the rest of the organization and the level of authority is appropriate.
- You understand how the organization operates, including how to get things done within the organization.
- You understand what you are responsible for delivering at both a macro and a micro level.
- You know the high-priority risks your project faces.

During Project Planning

- You know which external interfaces are not under your control.
- You know the estimated size of the software to be developed, and how the estimate was made.
- Funding has been allocated for the project.
- A credible budget has been prepared, based on project scope and work estimates.
- Adequate time has been allocated to complete the project.
- Adequate staff is or will be available to complete project tasks.
- The project staff has sufficient expertise to perform the work.
- Facilities and tools are or will be available for the project team.
- You know of potential funding cuts and when they might come.
- You know what major problems have plagued projects of this type in the past.
- An appropriate life cycle has been selected for the project and you understand that life cycle.
- You have a credible Work Breakdown Structure (WBS).
- All requirements have work tasks assigned to fulfill them.
- All work tasks are associated with project requirements or support activities.
- Special requirements or constraints are documented.
- You have a budget, schedule, and performance baseline established and documented.

- You have identified the critical path for the project.
- You have a process established to monitor the project and detect problems and departures from the baseline.

During Project Execution

- You know what your project's expenditures are to-date and any difference between that and your budget.
- You know the status of project activity completion along the critical path and any difference between that and the schedule.
- You are aware of any issues or problems with quality or performance that may impact the critical path.
- You are aware of any contract performance issues.

II. Software Life Cycle Checklist (Chapter 2)

This checklist is provided to assist you in choosing an appropriate life cycle for your project if you are beginning a development effort, or to ensure you understand your development life cycle if your project is already under way. If you cannot check an item off as affirmative, you need to rectify the situation yourself or get help in that area.

Beginning a Development Project

- Do you have an understanding of common life cycle models, along with their strengths, weaknesses, and constraints?
- Has the operational concept been analyzed to determine what life cycle method would best support the acquisition?
- Can the requirements be fully defined prior to the beginning of the project? Are they stable?
- Do you know the timeline for deployment of the new system?
- Are funds secure for development of the entire system?
- Are the risks identified?
- Will risks impact the ability of the project to move forward at crucial points in the system development?
- Is new or developing technology to be used in the system?
- Will there be a parallel hardware development effort?
- Do you understand the level of complexity of the system to be developed?
- Do you know what the interfaces to existing and future systems are?
- Do you know the size and magnitude of the development effort?
- Do you understand the users' needs?
- Are users able or expected to participate in the development?
- Do you know what types of acquisition contracts are available for this effort?
- When choosing a life cycle model, do you know why you are choosing it over other models?

Development Project is Under Way

- Do you know what life cycle model was selected for your project?
- Do you understand the project aspects pertaining to the life cycle:
 - Phases – What are they and what are they supposed to accomplish?
 - Milestones – What are they? What is their significance?
 - Criteria for transitioning from one phase to another?
 - Deliverables – What is expected, during phases, and at the end of the project?

- Reviews – What is reviewed when? Who are the reviewers? What actions follow a successful review, an unsuccessful review? What are the entry and exit criteria for each review?
- Feedback mechanisms – How is feedback obtained? Who provides it? Who receives it? How is it used?
- Documentation – What is to be produced and what it is used for?
- Do you know where your project is in the life cycle?
- Is your project following the life cycle?

III. Planning Checklist (Chapter 3)

This checklist is provided as to assist you in developing a project plan. If you cannot check an item off as affirmative, you need to either rectify the situation or develop a contingency plan to solve problems that may arise.

Before Planning

- 1. Have you published a list of contacts for all stakeholders and team members?
- 2. Do you have adequate, unambiguous project objectives or requirements to work with?
- 3. Do you have an action plan for developing the project plan, including a schedule and a responsibility matrix for participants in the planning activities.
- 4. Do you know which activities depend on outputs from other activities?
- 5. Have you documented the planning process?
- 6. Have you documented all constraints and assumptions?
- 7. Do you have a written outline listing the contents of your project plan?
- 8. Have you chosen computer based tools for planning and managing the project?

During Planning

- 9. Are project developers included in the planning process wherever possible?
- 10. Are you avoiding *paralysis by analysis*, where things are studied to such a level of detail that action never takes place?
- 11. Are you following your planning process?
- 12. Are the defined tasks unambiguous?
- 13. Does each task have a single point of responsibility?
- 14. Can each task be performed by an individual or a single team?
- 15. Is each task associated with a single, continuous time frame?
- 16. Have you considered holidays, vacations, and training in your schedule and staffing plans?
- 17. Have you considered project management activities such as planning, meetings, and managing people?
- 18. Have you considered overhead time such as system down time, outages, or system repairs?

After Planning

- 19. Is your plan realistic, that is, achievable?
- 20. Are all stakeholders and participants committed to supporting the project objectives?
- 21. Have all involved parties formally agreed with the project plan?
- 22. Does your project scope or any of the objectives need to be modified?
- 23. Have you documented lessons learned from the planning process?

IV. Requirements Engineering Checklist (Chapter 4)

This checklist is provided to assist you in requirements engineering. If you cannot check an item off as affirmative, you need to either rectify the situation or develop a contingency plan to solve problems that may arise.

Requirements Development

- 1. Have you had extensive user involvement in developing the requirements?
- 2. Do all stakeholders understand and agree on how the system will be used?
- 3. Are all stakeholders satisfied with the requirements?
- 4. Do the developers understand the requirements?
- 5. Are all requirements clear and unambiguous?
- 6. Have you distinguished between needs and wants? Are requirements relevant?
- 7. Are requirements consistent with each other (i.e., they don't conflict.)
- 8. Are requirements complete? Do the requirements cover everything that is supposed to be accomplished?
- 9. Has design detail been left out of the requirements?
- 10. Are all requirements testable?
- 11. Can you see the requirement as an output?
- 12. Are all requirements easily recognized as requirements by using the word shall?
- 13. Have the requirements been prioritized?
- 14. Are requirements feasible with respect to cost, schedule, and technical capability?
- 15. Are requirements verifiable?
- 16. Is the system boundary clearly defined; what is in scope, what is not?
- 17. Are all external interfaces to the system clearly defined?
- 18. Is the specification written so that it can be modified when necessary, with minimal impact to the rest of the document?
- 19. Are you conducting formal and informal reviews of requirements documents?

Requirements Management

- 20. Have all requirements been entered into the requirements database?
- 21. Are the requirements traces sorted to allow requirements lookup by paragraph number, requirement number, or other useful index?
- 22. Can all requirements be traced to original system-level requirements?
- 23. Are all system-level requirements allocated to lower level, subsystem requirements?
- 24. Do you have a requirements change process documented and in place?
- 25. Have you identified members of the requirements change board?
- 26. Is adequate impact analysis performed for proposed requirements changes?
- 27. Do you know who is responsible for making the changes?
- 28. Have requirement changes been traced upward and downward through the higher and lower-level specifications?
- 29. Do you have a process in place to maintain and control the different versions of the requirements specification?

V. Risk Management Checklist (Chapter 5)

This checklist is provided as to assist you in risk management. If you answer “no” to any of these questions you should examine the situation carefully for the possibility of greater risks to the project. This is only a cursory checklist for such an important subject. Please see the reference documents for more detailed checklists.

- 1. Do you have a comprehensive, planned, and documented approach to risk management?
- 2. Are all major areas/disciplines represented on your risk management team?
- 3. Is the project manager experienced with similar projects?
- 4. Do the stakeholders support disciplined development methods that incorporate adequate planning, requirements analysis, design, and testing?
- 5. Is the project manager dedicated to this project, and not dividing his or her time among other efforts?
- 6. Are you implementing a proven development methodology?
- 7. Are requirements well defined, understandable, and stable?
- 8. Do you have an effective requirements change process in place and do you use it?
- 9. Does your project plan call for tracking/tracing requirements through all phases of the project?
- 10. Are you implementing proven technology?
- 11. Are suppliers stable, and do you have multiple sources for hardware and equipment?
- 12. Are all procurement items needed for your development effort short-lead time items (no long-lead items?)
- 13. Are all external and internal interfaces for the system well defined?
- 14. Are all project positions appropriately staffed with qualified, motivated personnel?
- 15. Are the developers trained and experienced in their respective development disciplines (i.e. systems engineering, software engineering, language, platform, tools, etc.?)
- 16. Are developers experienced or familiar with the technology and the development environment?
- 17. Are key personnel stable and likely to remain in their positions throughout the project?
- 18. Is project funding stable and secure?
- 19. Are all costs associated with the project known?
- 20. Are development tools and equipment used for the project state-of-the-art, dependable, and available in sufficient quantity, and are the developers familiar with the development tools?
- 21. Are the schedule estimates free of unknowns?
- 22. Is the schedule realistic to support an acceptable level of risk?
- 23. Is the project free of special environmental constraints or requirements?
- 24. Is your testing approach feasible and appropriate for the components and system?
- 25. Have acceptance criteria been established for all requirements and agreed to by all stakeholders?
- 26. Will there be sufficient equipment to do adequate integration and testing?
- 27. Has sufficient time been scheduled for system integration and testing?
- 28. Can software be tested without complex testing or special test equipment?
- 29. Is the system being developed by a single group in one location?
- 30. Are subcontractors reliable and proven?
- 31. Is all project work being done by groups over which you have control?

- 32. Are development and support teams all collocated at one site?
- 33. Is the project team accustomed to working on an effort of this size (neither bigger nor smaller?)

VI. Cost Management Checklist (Chapter 6)

This checklist is provided as to assist you in cost management. Consider your answers carefully to determine whether you need to examine the situation and take action.

- 1. Is cost management planning part of your project planning process?
- 2. Have you established a formal, documented cost management process?
- 3. Do you have a complete and detailed WBS, including management areas (See Mil-HDBK-881, Appendix H)?
- 4. Do you have historical information, including costs, from previous similar projects?
- 5. Have you identified all sources of costs to your project (i.e. different types of labor, materials, supplies, equipment, etc.)?
- 6. Have you identified proven and applicable estimating methods, models, and/or guides?
- 7. Have you selected computer software to assist you in estimating, budgeting, tracking, and controlling costs?
- 8. Do you have justifiable reasons for selecting your methods, models, guides, and software?
- 9. Are cost issues adequately addressed in your risk management plan?
- 10. Do you have a working change control process in place?
- 11. Does the change control process adequately address cost impact?
- 12. Do your estimates cover all tasks in the WBS?
- 13. Do you understand your project's funding profile, i.e. how much funding will be provided? At what intervals? How sure is the funding?
- 14. Have you developed a viable cost baseline that is synchronized with the project schedule and funding profile?
- 15. Do you have adequate flexibility in the cost baseline?
- 16. Do you have a plan/process for dealing with variances between cost performance and the baseline?
- 17. Have you considered incorporating earned value management into your cost management efforts?
- 18. Are you keeping records of your cost management activity for future efforts?

VII. Time and Schedule Checklist (Chapter 7)

This checklist is provided as to assist you in Time and Schedule Management. Consider your answers carefully to determine whether you need to carefully examine the situation and take action.

Preparing for Schedule Development

- 1. Have you identified an experienced, knowledgeable team to develop the schedule?
- 2. Has a process to develop the project schedule been defined and documented?
- 3. Are all time and date requirements for the project known and documented? (What is needed when?)
- 4. Are there unreasonable time constraints for the project?
- 5. Are resource capabilities and availability known?
- 6. Do you have the project constraints, assumptions, and risk plan documented?

- 7. Do all deliverables listed in the WBS have adequate and appropriate activities identified to produce them?
- 8. Have you chosen an appropriate project management software package, and are you experienced or have you been trained in using it?
- 9. Is historical duration data available for project activities?

Schedule Development

- 10. Have you identified appropriate methods and models for estimating activity duration?
- 11. Have all activities been sequenced by putting them into an activity network and indicating the dependencies between them?
- 12. Have durations been estimated for all activities?
- 13. Have the activity durations been reviewed by people experienced in those activities?
- 14. Has the critical path been identified?
- 15. Has float time been documented for all activities not on the critical path?
- 16. When developing the schedule, are you using resource leveling and remembering holidays, vacations, and sick time?
- 17. Have you developed and documented a reality-based schedule?
- 18. Have you built a time reserve into your schedule for contingencies and unforeseen events?
- 19. Has your schedule been entered into a program management software package?

Schedule Control

- 20. Have you developed and documented a schedule management plan?
- 21. Do you know how, what, when, why, and how much to monitor for schedule control?
- 22. Are you being proactive vs. reactive in your approach to schedule control by looking ahead and asking what could go wrong?
- 23. Do you have a schedule change process documented and implemented?
- 24. Are you monitoring all preparatory actions, acquisitions, deliveries, and resources for each activity to make sure they are all complete and ready when it is time to begin the activity?
- 25. Are you using experienced people to make and review schedule progress reports?
- 26. Have the various groups and individuals been given sufficient levels of responsibility, accountability, and authority to perform their tasks? Have they agreed to their assigned roles?
- 27. Are you employing regular formal and informal schedule monitoring and progress reports?
- 28. Are you constantly aware of project milestones and your schedule progress?
- 29. Are you solving schedule problems by being creative and using common sense vs. extending the schedule?
- 30. Are you documenting your progress, problems, issues, solutions, and lessons learned?

VIII. Measurement and Metrics Checklist (Chapter 8)

This checklist is provided to assist you in developing a metrics program, and defining and using metrics. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action. The checklist items are divided into three areas: developing, implementing, and reviewing a metrics program.

Developing a Metrics Program

- 1. Is your use of metrics based on a documented metrics program plan?
- 2. Are you using the GQM paradigm in developing your metrics?

- 3. Are your metrics based on measurable or verifiable project goals?
- 4. Do your goals support the overall system-level goals?
- 5. Are your goals well defined and unambiguous?
- 6. Does each question elicit only information that indicates progress toward or completion of a specific goal?
- 7. Can questions be answered by providing specific information? (Is it unambiguous?)
- 8. Do the questions ask for all the information needed to determine progress or completion of the goal?
- 9. Is each metric required for specific decision-making activities?
- 10. Is each metric derived from two or more measurements (e.g. Remaining budget vs. schedule)?
- 11. Have you documented the analysis methods used to calculate the metrics?
- 12. Have you defined those measures needed to provide the metrics?
- 13. Have you defined the collection process (i.e. what, how, who, when, how often, etc.)?

Metrics Program Implementation

- 14. Does your implementation follow the metrics program plan?
- 15. Is data collected the same way each time it is collected?
- 16. Are documented analysis methods followed when calculating metrics?
- 17. Are metrics delivered in a timely manner to those who need them?
- 18. Are metrics being used in the decision making process?

Metrics Program Evaluation

- 19. Are the metrics sufficient?
- 20. Are all metrics or measures required, that is, non-superfluous?
- 21. Are measurements allowing project work to continue without interference?
- 22. Does the analysis produce accurate results?
- 23. Is the data collection interval appropriate?
- 24. Is the metrics program as simple as it can be while remaining adequate?
- 25. Has the metrics program been modified to adequately accommodate any project or organizational goal changes?

IX. Configuration Management Checklist (Chapter 9)

This checklist is provided to assist you in establishing an effective CM program. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

CM Planning

- 1. Have you planned and documented a configuration management process?
- 2. Have you identified Configuration Control Board members for each needed control board?
- 3. Has CM software been chosen to facilitate your CM process?

Establishing Baselines

- 4. Have all configuration items been identified?
- 5. Have baselines been established for all configuration items?

- 6. Has a descriptive schema been developed to accurately identify configuration items and changes to their configuration?

Controlling, Documenting, Auditing

- 7. Is there a formal process for documenting and submitting proposed changes?
- 8. Is the Configuration Control Board active and responsible in evaluating and approving changes?
- 9. Is there a “higher authority” to appeal to when the CCB gets “hung,” and can’t come to a consensus?
- 10. Are all changes tracked until they are fully implemented?
- 11. Are all changes fully documented in the baseline documents and change histories?
- 12. Are regular reports and configuration updates published and distributed to interested organizations?
- 13. Are regular audits and reviews performed to evaluate configuration integrity?
- 14. Are configuration errors dealt with in an efficient and timely manner?

Updating the Process

- 15. Is the CM program itself – its efficiency, responsiveness, and accuracy – evaluated regularly?
- 16. Is the CM program modified to include recommended improvements when needed?

X. Software Engineering Processes Checklist (Chapter 10)

This checklist is provided to assist you in understanding the software engineering issues of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Before Starting

- 1. Do you know what software development life cycle your project will be employing and how it coordinates with the software and project life cycles?
- 2. Does the development team have experience in the software development life cycle to be used?
- 3. Do the developers, the stakeholders, and you understand what the steps of the development process are, and what the inputs and products of each step are?
- 4. Has your project been planned in the various software development areas listed in Section 10.2.1 and in Chapter 3?
- 5. Do you know what design method has been chosen for the development effort and why it was chosen over other methods?
- 6. Does your development team have experience with the chosen design method? If not, has the schedule been adjusted to allow for learning the new design method?
- 7. Have proven CASE tools been chosen to assist in the software design?
- 8. Does your development team have experience with the chosen CASE tools? If not, has the schedule been adjusted to allow for learning to use the CASE tools?
- 9. Has an appropriate programming language been chosen and do you know the reasons it was chosen?
- 10. Does your development team have experience with the chosen programming language? If not, has the schedule been adjusted to allow for learning the chosen programming language?
- 11. Is the development team sufficiently skilled and experienced in programming to properly and efficiently design, code, and test the software?

During Project Execution

- 12. Are your requirements complete, unambiguous, and agreed to by both developers and stakeholders?

- 13. Have you completed both system and functional specifications, and have they been reviewed and approved by stakeholders?
- 14. Is the development team familiar with or provided with the appropriate opportunity to become familiar with the operating system and system hardware?
- 15. Is a detailed software design being completed, reviewed, and approved before coding starts?
- 16. Is testing being properly implemented and satisfactorily completed at unit, integration, and system levels before acceptance testing?
- 17. Are human factors being considered sufficiently in the software design?

At Completion

- 18. Does the completed software correctly implement the design?
- 19. Does the software meet the requirements?
- 20. Does the software meet the users' needs?

XI. Project Health Assessment Checklist (Chapter 11)

This checklist is provided to assist you in monitoring the health of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Before Starting

- 1. Have you prepared an overall plan for assessing project health?
- 2. Does your plan include a documented assessment process answering who, what, when, where, and how?
- 3. Does your assessment plan incorporate reviews, metrics, inspections and testing?
- 4. Do you have planned baseline budgets, schedules, etc. to compare actual project status to?
- 5. Have you scheduled regular reviews of your assessment process?
- 6. Do you have a database prepared to record project status for historical purposes?
- 7. Have you minimized the interference of your assessment activities with the project as much as possible?

Metrics

- 8. Have you developed a metrics plan as outlined in Chapter 8?
- 9. Have you selected appropriate measures and metrics for the different areas of your project (e.g. Have you implemented software development metrics applicable to your particular software efforts?)
- 10. Have you used historical data from other projects in establishing your metrics program?

Reviews [1]

- 11. Are reviews included in the project plan and schedule?
- 12. Is there a written plan for each review?
- 13. Does the plan define the scope of the review?
- 14. Does the plan specify how the review results will be documented and reported?
- 15. Does the plan identify the data to be collected?
- 16. Has the review been planned to minimize project impact?
- 17. Is a review follow up scheduled?
- 18. Does the report include recommended actions?

- 19. Are defects documented and tracked to closure?
- 20. Are all milestone stakeholders represented?
- 21. Are the Integrated Product Teams (IPTs) appropriately represented?
- 22. Does at least part of the technical review focus on software development and management?
- 23. Is the software product developed by the project suitable for its intended use?
- 24. Has risk management been addressed?
- 25. Have security issues been addressed?
- 26. Does the software comply with required standards and regulations?

Inspections

- 27. Do you have a plan for what is to be inspected, when, and by whom?
- 28. Have you used historical data to determine what is cost-effective to inspect and what is not?
- 29. Do your inspections minimize interference to the development work?
- 30. Have you emphasized inspections in the earlier stages of the project where testing cannot be performed?
- 31. Do you use the data collected from inspections to correct errors to keep them from propagating into later development stages?
- 32. Do you keep a history or database of all inspection activity, findings, and effectiveness?

Testing

- 33. Do you have a comprehensive test plan for your project, designating who, what, when, where, and how?
- 34. Have you made testing considerations a major input in the early stages of development?
- 35. Do you understand the limits of testing?
- 36. Have you implemented a testing program as outlined in Chapter 12?
- 37. Are you keeping a history of testing plans, results, and effectiveness?

XII. Testing Checklist (Chapter 12)

This checklist is provided to assist you in understanding the testing issues of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Before Starting

- 1. Is testing planned for and considered throughout the entire development life cycle?
- 2. Is the overall testing strategy defined and documented, and is it an integral part of and consistent with the development program?
- 3. Is the testing process well defined, documented, understood, and supported by the development team and management?
- 4. Are test requirements clearly defined?
- 5. Are test methods, techniques, controls, and standards clearly defined and consistent with the testing strategy?
- 6. Is each test activity traceable to specific requirements?
- 7. Are configuration management and quality assurance in place and are they adequate to support the testing strategy?
- 8. Are testers trained, skilled, and motivated people?
- 9. Have adequate time and resources been reserved for testing?

- 10. Are time and resources allocated for test preparation early in the project life cycle?

During Execution

- 13. Is testing used as a primary tool to ensure good project health?
- 14. Is testing implemented as a tool for improving product quality and the development process as a whole?
- 15. Is early life cycle testing used to prevent propagation of defects to later stages of development?
- 16. Is a tracking system being used to record what has been tested and what has not?
- 17. Is a database of test results being maintained for current and future reference?
- 18. Are tests used as milestone and progress indicators?
- 19. Is the right amount of testing being done to balance risk with available time and resources?
- 20. Are you using inspections and other evaluation methods (see Chapter 11) to reduce the errors found through testing?
- 21. Do you know when your testing is complete?

XIII. Systems Engineering Checklist (Chapter 13)

This checklist is provided to assist you in understanding the systems engineering issues of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Systems Engineering

- 1. Do you understand the systems engineering process?
- 2. Are you implementing an optimal systems engineering process?
- 3. Have you implemented proper and sufficient systems engineering controls and techniques?
- 4. Are you implementing systems engineering across the whole development life cycle?
- 5. Is there an experienced and skilled systems engineer directing the systems engineering effort?
- 6. Is a systems engineering representative providing input to or comments on all product change proposals?
- 7. Is the systems engineer seeing that all the various development efforts are coordinated and integrated?
- 8. Do you know what software development life cycle your project will be employing and how it coordinates with the software and project life cycles?
- 9. Are you considering all phases of the entire life cycle in your requirements, architectures, and designs?
- 10. Are you implementing an integrated product environment?
- 11. Have you established integrated (interdisciplinary) product teams?
- 12. Have you included all the necessary disciplines on the integrated product teams?
- 13. Are you documenting all studies, decisions, and configurations?
- 14. Have all internal and external interfaces been defined?
- 15. Are all your requirements verifiable?
- 16. Do all your requirements trace to products and vice versa?

COTS

- 17. Do you conduct make vs. buy vs. rent trade studies instead of just assuming that buy is the right choice?
- 18. Do you use your requirements as the criteria for your trade studies?

- 19. Do you consider the full life cycle when deciding whether to make, buy, or rent?
- 20. Do you know all your options?
- 21. Are you knowledgeable of the marketplace and the vendors?
- 22. Does the vendor understand your needs?
- 23. Are you not lowering your requirements indiscriminately to use COTS?
- 24. Do you understand the total life cycle costs?
- 25. Are the product and vendor likely to be around for the lifetime of the system?
- 26. Have you satisfactorily resolved all security issues?
- 27. Have you reduced all known risks to an acceptable level?

XIV. System Integration Checklist (Chapter 14)

This checklist is provided to assist you in understanding the system integration issues of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Before Starting

- 1. Have you implemented systems engineering as an integrated life cycle effort (see Chapter 13)?
- 2. Do your test plans include and support integration efforts?
- 3. Does your development plan allocate adequate time and resources for system integration efforts, including rework time?
- 4. Are the interfaces between components, assemblies, subsystems, and systems defined in adequate detail?
- 5. Will hardware be available for testing software during integration?
- 6. Is there a contingency plan if the schedule slips if and the integration schedule is compressed?
- 7. Are all elements of the system included in the integration plan?
- 8. Is all documentation current and available for reference?

During Integration

- 9. Is there an efficient rework cycle in place to fix problems found during integration testing?
- 10. Are “fixed” modules or components integrated and retested at all levels of integration up to the level where the problem was found?
- 11. Is the people element (operators, maintainers, logisticians, trainers, etc.) being prepared to work with the system when it is deployed?
- 12. Is the support systems element (logistics, maintenance, training, etc.) being prepared to support the new system when it is deployed?
- 13. Are you following an iterative, progressive integration process?
- 14. Are experienced integrators involved with the integration?
- 15. Are area/subject matter experts involved with the integration?
- 16. Is adequate time being allowed for integration, testing, rework, reintegration, and retesting?
- 17. Are all necessary resources being made available for integration?
- 18. Is adequate testing being performed on integrated units (assemblies, subsystems, elements, system) to ensure that there are no surprises during acceptance testing?
- 19. Are you updating documentation during rework?

- 20. Are integration and system test errors being traced back to requirements and design? And if so, are the requirements and design being updated?

XV. Software Design Checklist (Chapter 15)

This checklist is provided to assist you in understanding the software design issues of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Before Starting

- 1. Do you have a well-documented software development process?
- 2. Do you understand what is to be performed and produced in each phase of the design process?
- 3. Do you have a Software Standards and Conventions Document (SSCD)?
- 4. Does the SSCD contain direction in those areas listed in Section 15.2.3.1?
- 5. Are you familiar with the methods, tools, standards, and guidelines in the SSCD?
- 6. Are applicable and efficient design methods (OOD, etc.) being implemented on your project?
- 7. Are the developers experienced in the chosen development process and methods?
- 8. Is software reuse being considered throughout the development effort?
- 9. Has an analysis of alternatives been completed?
- 10. Is the selection of architecture and design methods based on system operational characteristics?

During Design

- 11. Are CASE tools being used to assist and document the design effort?
- 12. Does your design process include a robust configuration control process?
- 13. Is the design effort being properly documented? Adequate but not burdensome?
- 14. Is your team committed to following the design process?
- 15. Are all design elements traceable to specific requirements?
- 16. Are all requirements traceable to design elements?
- 17. Have all software units been identified?
- 18. Are the characteristics of all data elements identified (type, format, size, units, etc.)?

XVI. Sustainment and Product Improvement Checklist (Chapter 16)

This checklist is provided to assist you in understanding the sustainment and product improvement issues of your project. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

Sustainment

- 1. Is all your software developed with a goal to facilitate its future sustainment?
- 2. Do you understand the four types of sustainment and their purposes?
- 3. Do you understand the place and purpose of the sustainment phase in the software life cycle?
- 4. Do you understand your sustainment process?
- 5. Is there a sustainment plan?
- 6. Is there a process in place to gather problem reports and upgrade requests for the software?

- 7. Does the plan provide for reviewing, evaluating, and prioritizing upgrade requests?
- 8. Are all sustainment activity steps included in the plan?
- 9. Is there a transition plan to move to the upgraded system?
- 10. Have all activities been planned and organized to keep interference and downtime to the operating system to a minimum?
- 11. Does the plan call for running critical systems redundantly during testing and installation?
- 12. Do the deliveries include source code, documentation, and all else that is needed in addition to the software itself to continue maintaining the software?
- 13. Are all products under configuration control?

Product Improvement

- 14. Is your organization following a product improvement strategy?
- 15. Do you know where your organization is at, capability wise, relative to ISO 9001, CMMI or your chosen improvement standard?
- 16. Do you have a plan for achieving higher levels of capability?
- 17. Do your product improvement efforts emphasize improving processes to achieve product improvement?
- 18. Are your development processes documented?
- 19. Are your development processes consistent and repeated?
- 20. Does the leadership of your organization support continuous process improvement?
- 21. Is your organization committed to achieving a state of continuous improvement vs. a certificate of compliance with standards?

XVII. Acquisition Environment and Regulations Checklist (Chapter 17)

This checklist is provided to assist you in better understanding your project in relation to the DoD acquisition environment. If you cannot answer a question affirmatively, you should carefully examine the situation and take appropriate action.

- 1. Do you know what mission your project or acquisition supports?
- 2. Do you understand which FAR regulations apply to your project?
- 3. Do you understand which aspects of DoD 5000.1 apply to your project?
- 4. Do you understand which aspects of DoD 5000.2-R apply to your project?
- 5. Does your project implement an architecture that supports open systems, COTS, etc? (See 17.2.3.2)
- 6. Does your project try to identify and exploit software for reuse?
- 7. Are your language choices based on sound system and software engineering principles?
- 8. Are you employing contractors who have domain experience with similar projects, a successful past performance record, demonstrable development capability, and a mature development process?
- 9. Are you striving to implement best practices in all aspects of your development/acquisition project?