

# Chapter 11

## Assessing Project Health

---

### CONTENTS

<b><u>11.1</u></b>	<b><u>INTRODUCTION</u></b> .....	<b>3</b>
<b><u>11.2</u></b>	<b><u>PROCESS DESCRIPTION</u></b> .....	<b>3</b>
11.2.1	<u>INDICATORS OF PROJECT HEALTH</u> .....	4
11.2.2	<u>METRICS</u> .....	4
11.2.3	<u>REVIEWS</u> .....	4
11.2.3.1	<u>Milestone Decision Review (MDR)</u> .....	5
11.2.3.2	<u>Technical Review</u> .....	5
11.2.3.3	<u>Independent Expert Program Review</u> .....	5
11.2.3.4	<u>Walkthroughs</u> .....	6
11.2.4	<u>INSPECTIONS</u> .....	6
11.2.5	<u>TESTING</u> .....	7
11.2.6	<u>SUMMARY</u> .....	8
<b><u>11.3</u></b>	<b><u>PROJECT HEALTH ASSESSMENT CHECKLIST</u></b> .....	<b>8</b>
11.3.1	<u>BEFORE STARTING</u> .....	8
11.3.2	<u>METRICS</u> .....	8
11.3.3	<u>REVIEWS [1]</u> .....	9
11.3.4	<u>INSPECTIONS</u> .....	9
11.3.5	<u>TESTING</u> .....	9
<b><u>11.4</u></b>	<b><u>REFERENCES</u></b> .....	<b>9</b>
<b><u>11.5</u></b>	<b><u>RESOURCES</u></b> .....	<b>10</b>

This page intentionally left blank.

# Chapter 11

---

## Assessing Project Health

*“I see dead people... they’re everywhere. They walk around like everyone else. They don’t even know that they’re dead.” – from the movie “Sixth Sense”*

### 11.1 Introduction

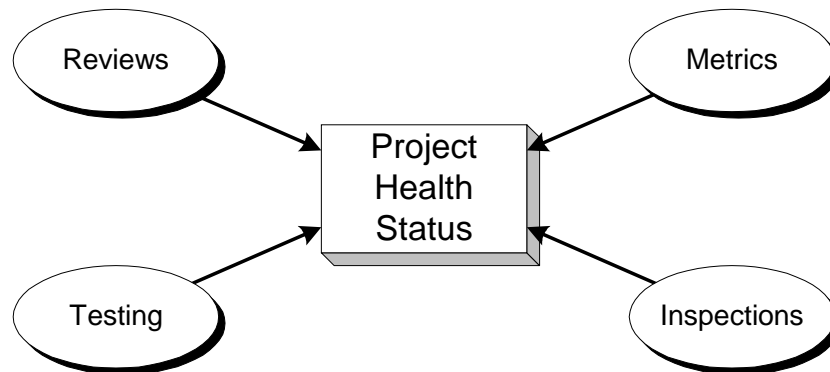
Projects can be viewed as living entities, requiring inputs to keep alive and having a purpose of producing something. In most organizations the project must have a charter or some other document to validate its existence and give it legal standing within the organization. If something can be considered a living entity, it can die. It also has health, or a status of its existence, indicating how well it is fulfilling its purpose or how close it is to death.

If a project is moving steadily towards its goals, according to the project plan, it can be considered healthy. The converse is also true. This, of course, assumes that the plan is realistic and based on historical data. Good health leads to life. Likewise, a project that is over budget, behind schedule, producing a poor quality product, or not meeting requirements can be diagnosed as having poor health. Poor health, if not checked and reversed, leads to early death. Some projects are dead long before anyone knows it.

Knowing whether your project is in good or poor health is absolutely essential to managing it. Hence, once a project has been given life, the first order of business for the Project Manager (PM) is to ascertain and track the project’s health. This chapter discusses indicators and processes of assessing project health.

### 11.2 Process Description

Project health, like a human’s health, can only be determined by observing and measuring specific indicators. There are four primary activities which gather information about project health. These are shown in Figure 11-1.



**Figure 11-1 Activities for Assessing Project Health Status**

These activities vary widely in their approach and are not all useful at the same time in the project life cycle. Reviews are snapshots of project status at specific times. While they can present trends by incorporating information from one or more of the other activities, by themselves, they provide a static look at the project. Inspections can be performed throughout the project whenever there is something, documents or otherwise, to inspect. Metrics can also be gathered throughout the project, depending on what is being measured. Data from both metrics and inspections can be used to spot trends and point to probable outcomes before they actually happen. Testing occurs toward the end of a project or development cycle and indicates quality and error content.

Wise project management will incorporate all these methods in assessing health throughout the project, recognizing their strengths and keeping in mind their weaknesses.

### 11.2.1 Indicators of Project Health

Just as a doctor will check pulse, blood pressure, temperature, and many other vital signs to determine a person's state of health, the activities introduced above examine or evaluate the life signs of a project. While there are many things that can be evaluated or monitored, the following list contains some of the more important project vital signs.

- Cost & Budget
- Risks
- Requirements
- Project Communications
- Schedule
- Funding
- Quality
- Human Resources
- Critical Path
- Planning
- Scope Stability
- Technical Progress

A proper state for each of these is vital to success and has an impact on the project as a whole. Any program of project health assessment should include these as a minimum.

### 11.2.2 Metrics

Metrics are measures of performance, quality, progress, or deviation from the plan. They can be used to evaluate health in any area. Metrics should be selected to indicate when and how progress is to be made, and also illuminate areas where there are problems or deficiencies. A good starting point would be to implement a metrics program that monitors the indicators in the previous section. Chapter 8 of this book provides an overview to establishing a metrics program.

### 11.2.3 Reviews

Reviews are usually structured, scheduled evaluations of a project's progress and vital signs. Project reviews are often associated with project milestones and may be decision points, where the stakeholders or management determine whether to move to the next phase of the project. Other reviews may consider only a portion of the project, such as the software, technical progress, or test readiness. Reviews have many purposes. Some of the more common are listed here.

1. Help reduce cost, shorten schedules, and reduce defects. [1]
2. Provide training and experience by introducing team members to others' work and work styles. [1]
3. Discover and bring management attention to problems and issues.
4. Verify that the deliverables are meeting standards. [1]
5. Ensure everything is ready for the next project phase.
6. Demonstrate progress to stake holders and verify that development is on track.
7. Help identify possible malicious code and prevent its incorporation into the system. [1]

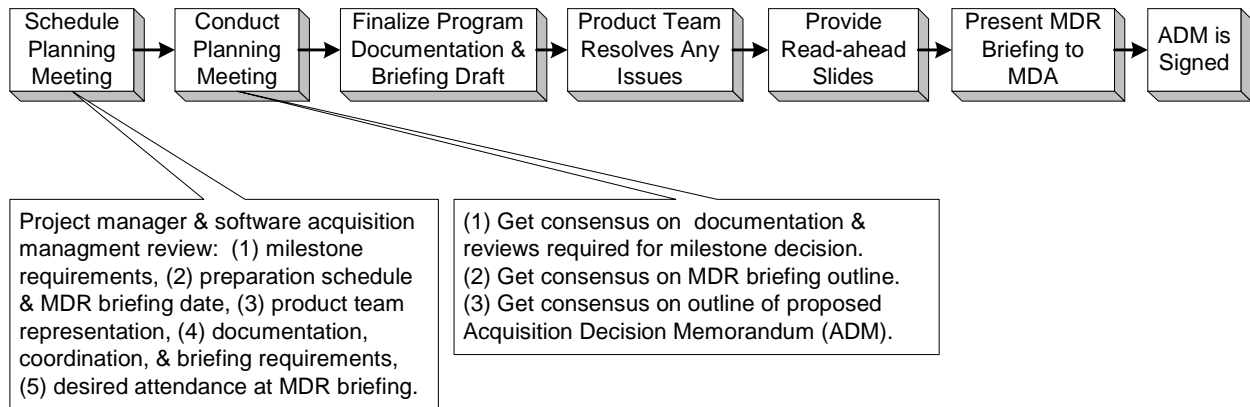
While reviews can evaluate almost anything, and be performed by any particular group, most reviews will be one of four types: [1]

1. Milestone Decision Review
2. Technical Review
3. Independent Expert Program Review
4. Walkthrough

The software development plan should also be reviewed at critical points to confirm, or revise if necessary, project costs, schedule, staffing, scope, etc.

**11.2.3.1 Milestone Decision Review (MDR)**

Milestone Decision Reviews are reviews of technology projects or acquisition programs, performed by the Milestone Decision Authority (MDA) at each milestone and other points in the project desired by the MDA. It usually consists of the presentation of program documentation and progress reports in an MDR briefing, but may also include demonstrations and inspections. When the milestone is achieved and the decision is made to proceed, an Acquisition Decision Memorandum (ADM) or equivalent document is signed. The MDR process is shown in Figure 11-2.



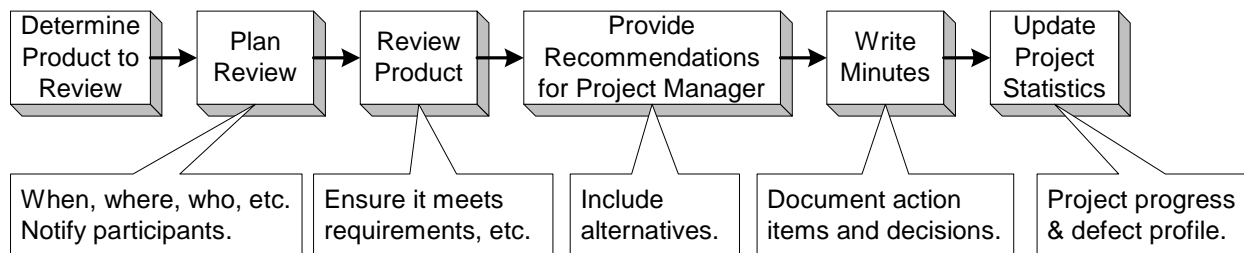
**Figure 11-2 Milestone Decision Review Process**

**11.2.3.2 Technical Review**

Technical Reviews, also called In-Process-Reviews (IPRs), are used to evaluate potential software or other products to determine if they meet the following criteria:

- Are suitable for their intended use.
- Meet project requirements.
- Function efficiently and effectively for the users and the mission.

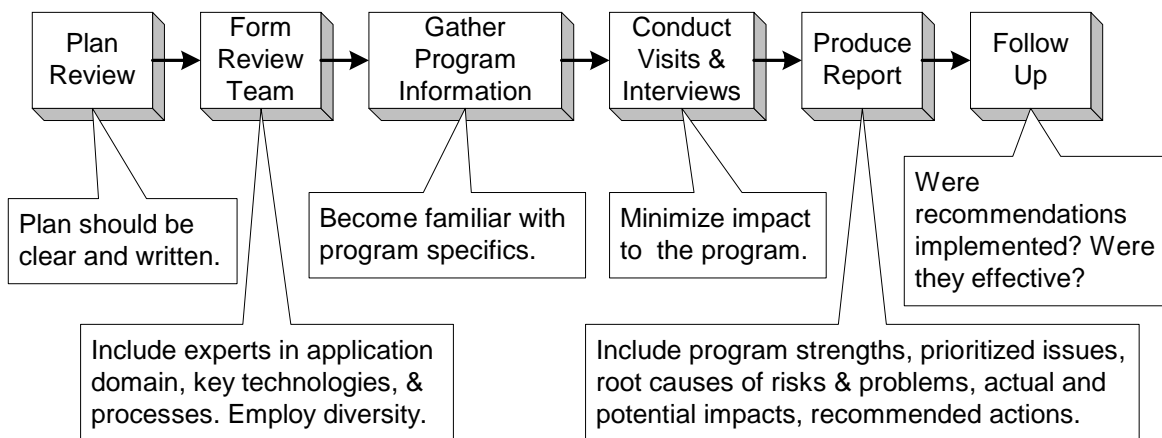
Figure 11-3 depicts the technical review process.



**Figure 11-3 Technical Review Process**

**11.2.3.3 Independent Expert Program Review**

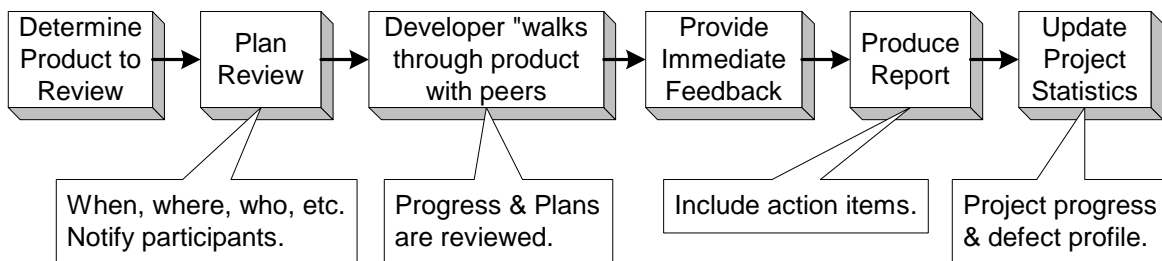
The Independent Expert Program Review is considered a best practice by both defense and commercial industries. In these reviews Subject Matter Experts (SMEs) are brought in from outside the project and stakeholder community to review the project for adequate progress and adherence to standards and best practices. The process is shown in Figure 11-4.



**Figure 11-4 Independent Expert Program Review Process**

#### 11.2.3.4 Walkthroughs

Walkthroughs are used extensively in software development but are not limited to that discipline. As the name implies, a well-prepared individual “walks through” the product, explaining the general concepts to the audience. All parts of the subject software’s requirements, design, and code are subject to walkthroughs. To quote Freedman and Weinberg, “Walking through the product, a lot of detail can be skipped – which is good if you’re just trying to verify an overall approach or bad if your object is to find errors of detail.” [3] The walkthrough process is shown in Figure 11-5.



**Figure 11-5 Walkthrough Process**

#### 11.2.4 Inspections

Another method of assessing project health, as well as improving product quality, costs, and schedule is the use of inspections. Inspection is the process of having individuals visually examine project products and documents, looking for errors and problems. Inspection can begin as soon as there is documentation to look at and can continue until the final product is delivered. Inspections, often implemented through the use of Independent Verification and Validation (IV&V) teams or via peer reviews, have proven their usefulness by discovering defects or design problems before they are implemented, shortening delivery time by reducing the time spent in integration and test phases. [2]

A special benefit of inspections is the fact that they begin to uncover errors long before testing can show the problem. A large German company found that, in comparison with defects found by formal inspection, defects discovered in testing cost 14.5 times as much to rectify, and those discovered by the customer cost 68 times as much. An IBM report placed the cost of fixing an error after release to be 45 times the cost to correct it during design. [2]

A final advantage gained by inspections is the data that can be collected from inspections and used to identify the more common problem areas, determine their causes, and change the development process to reduce or prevent those errors.

What should be inspected? While software source code is usually on the list, inspections should cover far more, beginning with requirements. Some of the better candidates for inspection materials are listed below.

- Requirements Specifications
- Models
- Source Code
- System documentation
- Test Plans and Procedures
- Design Documents
- Software Documentation

These are only basics. Essentially, any human-readable product involved with the development effort should be considered for inspection. [2] Those items produced earlier in the project, such as requirements, have the potential for greater payoff because they uncover defects earlier.

Unfortunately, good things do not come without cost. Inspections can cost from 5% to 15% of the project budget. You will need to determine what type and level of inspections are cost effective. This is done by analyzing the errors and defects found through inspection and realistically estimating what the cost to the project in time and money would have been had they been discovered later through other methods. When the probable cost of the errors has been found, this is compared with the cost of incorporating inspections. Money may not be the top consideration here. It may be that the savings in time far outweighs the cost of inspections.

The inspection process is simple and shown in Figure 11-6. After the initial inspection plan identifying who, what, when, and where, the cycle of inspect - report - statistics continues as each new artifact to be inspected is produced, until the end of the project.

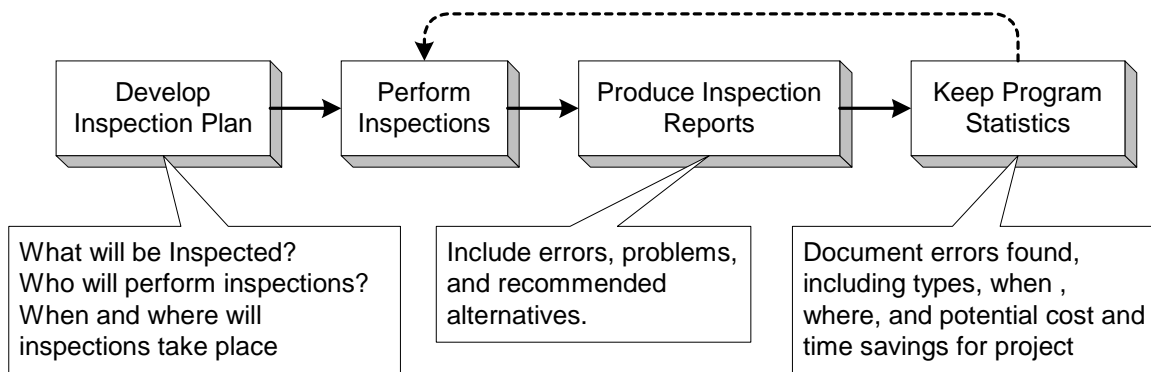
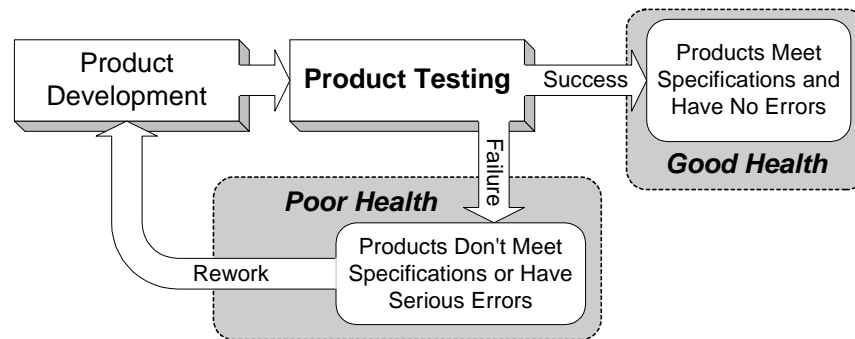


Figure 11-6 Inspection Process

### 11.2.5 Testing

Testing has two major purposes: to make a judgment about quality or acceptability, and discover problems or errors. The primary role of testing in assessing project health is to evaluate the quality of the products, their adherence to standards, functionality, and their compliance with requirements. As such it requires some product to test. This usually places testing in the latter part of the project. Because it works with the actual product, testing is an excellent indicator of health with reference to the product late in the project. On the down side, it is not useful for indicating health in the earlier phases of the project. Hence the importance of thorough test planning as early in each stage of the project as possible. By the time testing uncovers problems, significant rework will be required to correct defects, as shown in Figure 11-7.

*“Inspections find the defect, while testing – which usually occurs one or more development phases after the opportunity to inspect has passed – finds only the symptom.” – Priscilla J. Fowler*



**Figure 11-7 Role of Testing in Assessing Project Health**

It can be seen that while testing is important, it comes too late to rely on as an early warning indicator. An overview of testing can be found in Chapter 12 of this handbook.

### 11.2.6 Summary

By using a combination of reviews, metrics, inspection, and testing, a project's progress or deviation from the planned baseline can be accurately followed and even predicted. This will allow you to proactively manage the project, avoiding or dealing with problems before they slow the project's progress or become showstoppers.

Carefully plan when, where, and how you will use the assessment methods. In addition to constantly monitoring your project's health, regularly evaluate your health assessment program for adequateness and efficiency. Don't let the assessment program adversely affect the project it is monitoring. Make changes as necessary and keep a record of what works well and what doesn't for future reference. Get help from experienced PMs in implementing your program. Use the resources at the end of this chapter to get more information of implementing software-specific metrics and assessments.

## 11.3 Project Health Assessment Checklist

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.

### 11.3.1 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?

### 11.3.2 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?



### 11.3.3 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?

### 11.3.4 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?

### 11.3.5 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?

## 11.4 References

- [1] *Program Manager's Guide for Managing Software*, 0.6, 29 June 2001, Chapter 9:  
[www.geia.org/sstc/G47/SWMgmtGuide%20Rev%200.4.doc](http://www.geia.org/sstc/G47/SWMgmtGuide%20Rev%200.4.doc)
- [2] Wiegers, Karl E., "Improving Quality through Software Inspections,": [www.processimpact.com/pubs.shtml](http://www.processimpact.com/pubs.shtml)
- [3] Friedman, Daniel P. and Gerald M. Weinberg, *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, New York NY, 1990

## 11.5 Resources

Crosstalk Magazine: [www.stsc.hill.af.mil/crosstalk/](http://www.stsc.hill.af.mil/crosstalk/)

- “The Determining Factor”: [www.stsc.hill.af.mil/crosstalk/2000/may/dynes.asp](http://www.stsc.hill.af.mil/crosstalk/2000/may/dynes.asp)
- “Help Identify and Manage Software and Program Risk”:  
[www.stsc.hill.af.mil/crosstalk/2000/nov/baldwin.asp](http://www.stsc.hill.af.mil/crosstalk/2000/nov/baldwin.asp)
- “Software Mini-Assessments: Process and Practice”: [www.stsc.hill.af.mil/crosstalk/1999/oct/natwick.asp](http://www.stsc.hill.af.mil/crosstalk/1999/oct/natwick.asp)
- “How Much Code Inspection is Enough”: [www.stsc.hill.af.mil/crosstalk/2001/07/index.html](http://www.stsc.hill.af.mil/crosstalk/2001/07/index.html)
- “Using Inspection Data to Forecast Test Defects”:  
[www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1998/05/inspection.asp](http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1998/05/inspection.asp)
- “Document Diseases and Software Malpractice”: [www.stsc.hill.af.mil/crosstalk/2002/nov/daich.asp](http://www.stsc.hill.af.mil/crosstalk/2002/nov/daich.asp)

Department of Energy (DOE) *Software Engineering Methodology*: [http://cio.doe.gov/sqse/sem\\_toc.htm](http://cio.doe.gov/sqse/sem_toc.htm)

Ebenau, Robert G. and Susan H. Strauss, *Software Inspection Process*, McGraw-Hill, Inc., New York NY, 1994

Friedman, Daniel P. and Gerald M. Weinberg, *Walkthroughs, Inspections, and Technical Reviews*, Dorset House Publishing, New York NY, 1990

Gilb, Tom and Dorothy Graham, *Software Inspection*, Addison-Wesley, New York NY, 1993

*Guidelines for the Successful Acquisition and Management of Software-Intensive Systems (GSAM)*, Version 3.0, Chapter 13 and Appendix N, OO-ALC/TISE, May 2000. Available for download at:  
[www.stsc.hill.af.mil/gsam/guid.asp](http://www.stsc.hill.af.mil/gsam/guid.asp)

*Program Manager's Guide for Managing Software*, 0.6, 29 June 2001, Chapter 9:  
[www.geia.org/ssstc/G47/SWMgmtGuide%20Rev%200.4.doc](http://www.geia.org/ssstc/G47/SWMgmtGuide%20Rev%200.4.doc)

Radice, Ronald A., *High Quality Low Cost Software Inspections*, Paradoxicon Publishing, Andover MA, 2002

U.S. Treasury Dept., *Systems Development Life Cycle Handbook*, Chapters 3 and 4:  
[www.customs.ustreas.gov/contract/modern/sdlcpdfs/](http://www.customs.ustreas.gov/contract/modern/sdlcpdfs/)

Wieggers, Karl E.,

- *Peer Reviews in Software: A Practical Guide*, Addison-Wesley, 2002
- “Seven Truths About Peer Reviews”: [www.processimpact.com/pubs.shtml](http://www.processimpact.com/pubs.shtml)
- “Do your Inspections Work?”:  
[www.stickyminds.com/sitewide.asp?Function=WEEKLYCOLUMN&ObjectId=3526&ObjectType=ARTCOL](http://www.stickyminds.com/sitewide.asp?Function=WEEKLYCOLUMN&ObjectId=3526&ObjectType=ARTCOL)
- “A Software Metrics Primer”: [www.processimpact.com/pubs.shtml](http://www.processimpact.com/pubs.shtml)
- “Metrics: Ten Traps to Avoid”: [www.processimpact.com/pubs.shtml#metrics](http://www.processimpact.com/pubs.shtml#metrics)
- “Lessons from Software Work Effort Metrics”: [www.processimpact.com/pubs.shtml#metrics](http://www.processimpact.com/pubs.shtml#metrics)
- “Improving Quality through Software Inspections,”: [www.processimpact.com/pubs.shtml](http://www.processimpact.com/pubs.shtml)
- “Seven Deadly Sins of Software Reviews,”: [www.processimpact.com/pubs.shtml](http://www.processimpact.com/pubs.shtml)
- “When Two Eyes Aren't Enough,”: [www.processimpact.com/pubs.shtml](http://www.processimpact.com/pubs.shtml)