

This is the final edit Jan 12 2000 for the article scheduled for publication in Software Quality Professional (American Society for Quality) March 2000. Editor Taz.Daughtrey@bw.mcdermott.com
SQP_Editor@asqnet.org

Tom Gilb Gilb@acm.org

Planning to Get the Most Out of Inspection

Tom Gilb, Result Planning Limited

Inspection, a proven technique for achieving quality and identifying process improvements, should be applied to documents throughout software development. The greatest value from inspection can be gained through a proper understanding of its purposes and benefits. Newer practices, such as sampling, should be incorporated into the more traditional application of this technique. The full benefit of inspection can be found as it contributes to measurement, exit control, and defect injection prevention.

Key words: defect detection, defect prevention, documentation, process improvement, project management, return on investment, training

INTRODUCTION

Inspection is a proven technique for achieving quality control of specifications and identifying associated process improvements. In fact, inspection can be applied to any discipline that produces

documents. It has been applied with excellent results to hardware engineering, management planning, and sales contract documentation.

Software Inspections are widely known within the software industry, but most organizations do not make the most of them. This is because many people misunderstand and misinterpret Inspection. Often, they assume there is only one inspection method--the 1976 IBM version.

This article aims to provide some direction on getting the most value from Inspections. It will also update readers on new practices, such as sampling. The authors assume that readers are familiar with the basic Inspection process presented in the book *Software Inspection* (Gilb and Graham 1993) (see Figure 1 for a simplified overview). Readers who want to learn about Inspection in depth should see (Gilb and Graham 1993; and URL www.result-planning.com)

SOME BASIC DEFINITIONS

The author's process of *Inspection* (shown with a capital "I" to differentiate it from the old Inspection method) consists of two main processes: the defect detection process and the defect prevention process. The defect detection process is concerned with document quality, especially identifying and measuring defects in the documentation submitted for Inspection and using this information to decide how best to proceed with the main (product) document under inspection.

The defect prevention process is concerned with learning from the defects found and

suggesting ways of improving processes to prevent them from reoccurring. Note, the process brainstorming component of the defect prevention process is not a costly, in-depth examination of all defects; for each Inspection, it simply involves brainstorming the reasons and preventive cures for several selected defects. The major part of the defect prevention process involves in-depth process analysis, and it is actually carried out off-line from the normal day-to-day inspection of specific documents.

A major defect is a defect that, if not dealt with at the requirements or design stage, will probably have an order-of-magnitude or larger cost to find and fix when it reaches the testing or operational stages. On average, the find-and-fix cost for major defects is one work hour upstream but nine work hours downstream (Gilb and Graham 1993, 315).

A page is a logical page. It is defined as a unit of work on which Inspection is performed. A page must be defined as a quantity of noncommentary words (for example, 300 words).

BENEFITS AND CURRENT BEST PRACTICE

Given adequate management support, Inspection can quickly be turned from the initial chaos phase (20 or more major defects per page) to relative cleanliness (two or fewer major defects per page at exit) within a year (Gilb and Graham 1993). A good example is the experience of the British Aerospace Eurofighter Project. In software documentation, more than 20 defects per page were reduced to 1 to 1.5 defects per page within 18 months.

On one pass, the defect detection process can find up to 88 percent of existing major defects in a document (Gilb and Graham 1993, 23). This is important, but there is actually greater benefit achieved by the teaching effect of Inspection feedback. By attending Inspections, software engineers go through a rapid learning process, which typically reduces the number of defects they make in their subsequent work by two orders of magnitude within about five inspection experiences, and within a few weeks.

The defect detection process can and should be extended to support continuous process improvement by including the associated defect prevention process. The defect prevention process is capable of at least 50 percent (first year of project) to 70 percent (second or third year) defect cause reduction, and more than 90 percent in the longer term (Gilb and Graham 1993). It has also shown at least 13-to-1 return on investment (ROI) for the ratio of the downstream cost savings of engineering time (rework cost saved by using Inspection) compared to the operational cost of carrying out the Inspections (Gilb and Graham 1993; The Raytheon Report 1995; and Kaplan, Clark, and Tang 1994).

The defect prevention process is the model for the Software Engineering Institute's Capability Maturity Model (SEI CMM) level 5. Robert Mays worked with Ron Radice, who developed the CMM model at IBM. This model was the basis for the SEI model (Radice et al. 1999; Radice and Phillips 1988). Radice himself codeveloped Inspection with Michael Fagan (Kohli and Radice 1976).

Raytheon provides a good case study. In six years, from 1988 to 1994, using the defect detection process combined with the defect prevention process, Raytheon reduced rework costs from about 43 percent to between 5 percent and 10 percent, and, for process improvement, achieved ROI of 7.7-to-1. It improved software code generation productivity by a factor of 2.7-to-1, reduced negative deviation from budget and deadlines from 40 percent to near zero, and reduced defect density by about a factor of three (The Raytheon Report 1995).

Smaller software producers (30 to 60 programmers) have also experienced major business improvements as a result of using Inspection (Holland 1999). Further detailed costs and benefits can be found in (Gilb and Graham 1993; and URL www.result-planning.com)

IMPROVING INSPECTIONS

The following sections contain tips for improving the Inspection process and achieving the kinds of results cited previously. The tips are grouped under the part of the Inspection process they chiefly apply to. (Readers should keep in mind that some tips do cover a broader section of the process) (See Figure 1, Overview of the Inspection process, and Figure 2, List of key tips, which shows the mapping of the key tips to the Inspection process).

Inspection Strategy

- *Don't misuse Inspection as a clean-up process. Use it to motivate, teach, measure, control quality, and improve processes.* Many people think Inspection is for cleaning up bad work, embedded faults, and other defects. The greatest payback, however, comes from the improved

quality of future work. Ensure that the Inspection process fully supports the aspects of *teaching* and *continuous process improvement*.

For continuous process improvement, integrate the defect prevention process into conventional inspections. The defect prevention process must be practiced early and should be fully integrated into inspection (see Gilb and Graham 1993, ch. 7 and 17 for details). CMM level 5 is too important to be put off until later--it needs to be done from the start.

- *Use Inspection on any technical documentation.* Most people think Inspection is about source code inspection. Once one realizes that Inspection is not a clean-up process, it makes sense to use it to measure and validate any technical documentation--even technical diagrams. Requirements and design documentation contribute 40 percent to 60 percent of code defects anyway (Pence and Hon 1993).
- *Inspect upstream first.* By the end of the 1970s, IBM and Inspection-method founder, Michael Fagan, recognized that defects, and thus the profitable use of Inspection, actually lie upstream in the requirements and design areas. Bellcore found that 44 percent of all defects were due to defects in requirements and design reaching the programmers (Pence and Hon 1993). Because systems development starts with contracts and management and marketing plans, the Inspection activity must start there, where the problems originate.

One of the most misunderstood dictums from early inspections is “No managers present.” This

is wrong. Managers should only be excluded from Inspection that they would corrupt by their presence. They should not be excluded from Inspection of management-level documents, such as requirements or contracts. Nor should they be excluded if they are trying to experience the method with a view to supporting it. Having managers take part in Inspections is a great way to get their understanding and support. “No managers present” is a rule from the past when IBM was doing source code inspections.

- *Make sure there are excellent standards to identify defective practices.* Inspection requires that good work standards (Gilb and Graham 1993, 424) be in place. Standards provide the rules for the author when writing technical documents and for the Inspection process to subsequently check against. An example of a simple, powerful generic rule is “specifications must be unambiguous to the intended readership and testably clear.” Violation of this rule is a defect.

Standards must be built by hard experience; they must be brief and to the point, monitored for usefulness, and respected by the troops. They must not be built by outside consultants or dictated by management. They must be seen as the tool to enforce the necessary lessons of professional practice on the unwary or unwilling.

- *Give Inspection team leaders proper training, coaching after initial training, certification, statistical follow-up, and, if necessary, remove their “license” to inspect.* Proper training of team leaders takes about a week (half lectures and half practice). Experience shows that less than this is not adequate. Inspection team leader certification (an entry condition to an Inspection) should

be similar in concept to that for pilots, drivers, and doctors--based on demonstrated competence after training. Note, at present there is no industry-recognized license or certification standard for Inspection.

Team leaders who will not professionally carry out the job, even if it is because their supervisor wants them to cut corners, should have their “licenses” revoked. Professional Inspection team leadership must be taken seriously so checkers will take inspection seriously. Ensure that there are enough trained Inspection team leaders to support Inspections within an organization--at least 20 percent of all professionals. Some clients train all their engineers on a one-week team leader course.

Entry Conditions

- *Use serious entry conditions, such as minimum level of numeric quality of source documents.*

Lack of discipline and lack of respect for entry conditions wastes time. One of the most important entry conditions is mandating the use of upstream source documents to help inspect a product document. It is a mistake to try to use the experts’ memory abilities (instead of updated, inspection-exited source documents). It is also a mistake to use source documents with the usual uncontrolled, uninspected, unexited, 20-or-more major defects per page to check a product document (The figure “20 or more” comes from the author’s experience over several years. In fact, from 20 up to 150 major defects per page is not uncommon in environments where Inspection is new).

It is not a good idea for the author to generate a product document using a poor quality source document. It is easy to check the state of a source document by using inexpensive sampling. A half-day or less on a few pages is a small price to pay to ensure the quality of a document. Another serious entry condition is carrying out a cursory check on the product document and returning it to the author if it has too many remaining defects. For example, if while planning the Inspection, the team leader performs a 15-minute cursory check that reveals a few major defects on a single page, it is time for a word with the author in private. If necessary, pretend the document was never seriously submitted. Do not waste the Inspection team's time to try to approve shoddy work.

In short, learn which entry conditions have to be set and take them seriously. Management needs to take a lead on this. It is often managers who are actually responsible for overriding the entry criteria. For example, carrying out an inspection is often mistakenly seen as fulfilling a quality process (regardless of the Inspection results). Managers have been known to demand that Inspections proceed even when a team leader has determined that the entry condition concerning majors per page is violated.

Planning

- *Plan Inspections well using a master plan.* Use a one-page master plan (the latest forms are on www.result-planning.com, or see slightly older ones in Gilb and Graham 1993, 401.) rather than the conventional invitation. Document the many supporting documents needed, assign checkers special defect-searching roles, and carefully manage the rates of checking and the total checking

time needed. Establish the formal purpose(s) of each specific Inspection--they do vary. Ensure a team numeric stretch goal is established and that there is a specific strategy to help attain it. A good master plan avoids senseless bureaucracy and lays the groundwork for intelligent Inspections.

- *Plan Inspection to address the Inspection purposes.* There are more than 20 distinct purposes for using Inspection, including document quality, removing defects, job training, motivation, helping a document author, improving productivity, and reducing maintenance costs (see Figure 4). Each Inspection will address several of these purposes to varying degrees. Be aware which purposes are valid for a specific Inspection and formally plan to address them (that is, by choosing checkers with relevant skills and giving them appropriate checking roles).
- *Inspect early and often while documents are still being written.* Leaving Inspection until after a large technical document is finished is a bad idea. If the process that generates the document is faulty, discover it early and fix it. This saves time and corrects bad processes before they cause too much damage. This is one form of sampling.
- *Use sampling to understand the quality level of a document.* It is neither necessary nor desirable to check all pages of long documents. Representative samples should provide enough information to decide whether a document is clean enough to exit at, for example, “0.2 major defects per page maximum remaining.”

The main purpose of Inspection is *economic*--to reduce lead time and people costs caused by downstream defects. As in Harlan Mills' IBM "cleanroom" method (Mills and Linger 1987), defects should be cleaned up or avoided using disciplines such as Watts Humphrey's Personal Software Process (PSP) (1995), structured programming (Mills 1972; Mills and Linger 1987), defect prevention/continuous improvement (Gilb and Graham 1993), inspection, and verification. If all this works as it should, cleaning is unnecessary and sampling provides the information to decide if it is economically safe to release the document. Perfection is not required; it costs infinite resources and is dangerous as a guiding concept.

- *Check against source and kin documents; check them for defects, too.* Because of potentially poor quality control practices and craftsmanship, and because Inspection is imperfect on first pass (30 percent to 88 percent effective) (Gilb and Graham 1993, 23), one must focus on the major defects that still persist in source and kin documents. Source documents are input documents used as upstream engineering process inputs to produce the product document being evaluated for possible exit. Kin documents are documents derived from the same source documents as the product document. For example, a requirements document can be a source document and used to produce a design specification (a product document) that requires inspection. Associated kin documentation to consider including in the Inspection would be the testing specification.

Most people over focus on the product document. In fact, the aim should be to find roughly 25 percent of the total defects external to the product document, mainly in source documents, even

when they have exited with no more than one major defect per page.

- *Check the significant portions of the material--avoid checking commentary.* Most organizations waste time checking nonsignificant document areas. It is a waste of checker energy to check at optimum rates to uncover minor defects with no downstream savings. It is necessary to go at optimum rates to find the major defects but ensure that time is not wasted at those rates (one logical page of 300 noncommentary words checked per hour plus or minus 0.9 logical pages is the expected optimum checking rate range). The result of indiscriminate checking of trivia at an optimum rate could be 90 percent minor defects and 90 percent waste of time. It is equivalent to checking comments for 90 percent of the time instead of real code.

From practical experience, it pays to have a general specification rule that technical authors must distinguish between noncommentary and commentary text (or diagrams). Noncommentary (or “meat”) text is text where any defects might translate into serious downstream costs (that is, major defects could be found). Commentary (or “fat”) text can only contain minor defects and so is less important.

The distinction between “meat” and “fat” can be achieved, for example, by using italics for the *fat*. Some clients have even created Microsoft Word macros to count the volume of noncommentary text (nonitalics) and print the logical page count on the first page (Holland 1994; Holland 1999). Of course, the checker is allowed to scan and reference the commentary words but is not obliged to check them against all sources, rules, and checklists; it is not worth

it.

- *Define a major defect as “possible larger costs downstream.”* It does not matter if a defect is not precisely a nonconformance or if it is visible to a customer. If it could lead to significant costs if it escapes downstream, classify it as a major defect, and treat it with due respect.

Note major and minor after the checklist questions or rule statement. It is often useful to indicate super major or showstopper (a defect where the downstream effect could be an order of magnitude bigger than an average major, which is about nine hours downstream loss average) (Gilb and Graham 1993, 315). Super majors can be highlighted for management attention.

Concentrate on the major defects. This helps avoid the “90 percent minor syndrome” that often hampers Inspection. As mentioned previously, employees will waste time identifying 90 percent minor defects, unless strongly redirected. There are at least 18 different tactics that shift the focus from minor to major defects (see Figure 3). For example, using checklists can help people identify majors rather than minors. The checklist contents should aim to detect majors and not minors. (Note: checklists are only allowed to help interpret the rules, which are an organization’s official standards for writing a given document, and which define what constitute defects).

Another useful tactic is to log only majors at a meeting and calculate the ROI for Inspections only on the basis of the majors. This sends a clear message not to waste time on minor defects.

- *Check at an organization's optimum (coverage) checking rates to find major defects.* This is the big one. Most people, including many teachers of Inspection, manage to miss this point. Or worse, they recommend checking rates that are 10 times optimum speed (Kelly 1990; The Raytheon Report 1995). Optimum checking rate is not a reading rate. Checking in real Inspections involves checking a page against all related documents. This can involve up to 10 or 20 individual documents; these are source documents of large size, checklists, and standards. The requirement is to check a single product document line against many sources, and it takes time.

Adequate Inspection statistics can prove that an organization's employees have a clear, dramatic, and consistent optimum checking rate on specific document types. The expected optimum checking rate range is between 0.2 and 1.8 pages of 300 noncommentary words per checking hour. For example, at Raytheon it was about 20 plus or minus 10 lines per hour (0.3 pages) (Haley et al. 1995). Unfortunately, in spite of its own data, Raytheon suggested rates of about 100 to 250 lines per hour. This was probably because it had finite deadlines and did not understand sampling (see Figure 5).

As the checking speed moves toward an optimum speed for effectiveness of finding major defects, the curve for optimum checking rate moves dramatically upward in terms of major defects identified per logical page. The optimum may seem slow, but considering the amount of checking that has to be done, it is fast. The point is that there is a best speed at which to check,

and an organization could easily be operating at only 1 percent of defect identification effectiveness if it fails to heed it.

Note that the optimum checking rate applies both to the checking carried out during the individual checking phase (also known as preparation) and to the optional checking carried out during the logging meeting. This second logging-meeting check will produce roughly an additional 15 percent defects (Gilb and Graham 1993, 86). In fact, there is no need to carry out this extra checking if the document is found to be clean enough to exit as a result of initial checking sampling, or if it is so polluted that a major rewrite is required anyway. It is only useful in clean-up mode.

- *Use an optimum number of people on a team to serve the current purpose of Inspection, for example, effectiveness, efficiency, and training.* For 13 years, one large U. S. telecommunications company had 12 to 15 people on each inspection team because each person “had” to be sent there to protect territorial interests. There seemed to be no motivation to cut these costs.

The number of people needed on an inspection team is a function of the Inspection purposes. By measuring Inspection experiences, it has been established that best effectiveness at finding major defects uses four to six people; best efficiency (effect over cost) needs two to four people; and only ‘teaching as a purpose’ justifies larger numbers (Kelly 1990; Weller 1993). The results of varying team sizes should be monitored within an organization to discover the optimum for a given document type.

- *Allocate special defect-searching roles to people on the team.* Each person on an Inspection team should be finding different defects. Much like a coach on a ball team, the Inspection team leader should assign specialist roles to team members (for example, identify time and money risks, check against corporate standards for engineering documentation, and check security loopholes) when planning the inspection. Special role checklists help people know exactly what to look for.

Individual Checking

- *Use checking phase data that are collected at the beginning of a logging meeting or beforehand (such as pages checked, majors found, time used, and checking rate) from individual checkers to decide whether it is worth holding a logging meeting.* Older inspections plunge into the logging meeting without forethought and consequently waste time. A process of logging meeting entry evaluation must be carried out before holding a logging meeting. To do this, collect the data from the checkers about their checking rates and major issue density (Note, to avoid personal conflict, issues--not defects--are logged during the logging meeting. An issue may or may not become a defect, as judged by the responsible product document editor later). Based on this checking-phase data, make a series of decisions about the logging meeting and the rest of inspection. The most critical decision is whether a meeting is necessary. Other decisions include whether to log minors and whether to continue checking. Shutting down the rest of the inspection entirely is also a possibility.

- *Use the individual checkers' personal notes instead of proper meeting defect logs when the major-issue density is (nonexit level) high, or when there are many minor defects.* Checkers should not be required to use any particular method to make notes during the checking process. In practice, most checkers choose to mark the defective words on a paper document, using underlines, circles, or highlights. Some use electronic notes. It is important that they note, against the defective words, exactly which rule was broken (the issue). A note classifying subjective decisions as to severity (major or minor) of the issue is also required.

Whenever there are more issues than would be an allowable exit level, it is suggested that, with author agreement, the notes made during individual checking (sometimes known as 'scratchings') be simply handed over to the author. This is better than pedantically logging all the issues. In such situations, authors must rewrite and resubmit their documents, and they might as well use the rough checking information to correct their work. The usual problem leading to a high defect density is that the author fails to take sources and rules sufficiently seriously.

Logging Meeting

- *At logging meetings, avoid discussions and suggesting fixes.* Inspection is not for talkers and quibblers--it is for professionals committed to making maximum, meaningful progress on the project. It is important to have a good time but not by detailed technical discussion, idle gossip, or insults.

Process Brainstorming

- *Use the defect prevention process on Inspection itself for continuous improvement.* Recognize that systematic continuous improvement of the inspection process is necessary. Initially, this is required not only to improve the Inspection process but also to learn to implement the correct Inspection process and tailor it to the organization.

Exit Conditions

- *Use serious exit conditions, such as “maximum probable remaining major defects per page is 0.2 for exit.”* Exit conditions, if correctly formulated and taken seriously, can be crucial. It is wrong to have the customary vote to accept a document once the logged defects are fixed, because this ignores the known factor of the number of remaining unfound defects (a value that is computable and verifiable from past data and experience).

Remember that Inspection processes, like other testing processes, have a maximum effectiveness for a single pass in the range of 30 percent to 88 percent of existing defects. If the maximum probable remaining defect density is a high-quality low count, such as 0.2 majors per page, then it does not matter much if the detected defects are removed at this stage; the document is clean enough (economically speaking) to exit without fixing them. It is probably better to catch them later.

If defect density is high, for example, 20 or more majors per page (quite common), the undetected defects, at say 50 percent effectiveness, are more than enough to make exit uneconomical. If there are 10 majors remaining per page in a 100-page document, there are an

expected 9 x 10 x 100 hours of additional project work to clean them up by testing and discovery in the field. It costs an order of magnitude less to find them now. Admittedly, this situation is only the lesser of two evils. Ideally they should have been prevented in the first place using the defect prevention process rather than being cleaned up, even if at an earlier stage than test, using the defect detection process. Management must understand the large-scale economics of this and take action to make clear policy about the levels of major defects per page that will be allowed to escape (Gilb and Graham 1993, 430-431).

Inspection Statistics

- *Build or buy an automated software tool to process Inspection basic data.* Use automated software to capture data-summary data and to present trends and reports (Software Development Technologies 1997). Inspection generates a lot of useful data. It is vital that good computer support be given early, so the process owners and management take the data seriously and the early champions are not overwhelmed.

The key distinction between Inspections and other review processes is the use of data to manage inspections. For example, the optimum checking rates must be established early and updated as they change, through continuous improvement. It also is vital to statistically see the consequences of inadequate exit levels (too many major defects floating downstream), which then must be caught with expensive testing processes.

- *Put Inspection artifacts on a company Web site.* If an organization has an intranet, all relevant

Inspection artifacts, standards, experiences, statistics, and problems should be added as soon as possible.

- *Measure the benefit from using Inspections.* Inspection should always be highly profitable, for example, 10-to-1 ROI. If not, then it is time to adjust the Inspection process or to stop it. Benefits to be measured include rework costs, predictability, productivity, document quality, and ROI (Haley et al. 1995). Inspection profitability must be evaluated for each type of specification individually. In general, the upstream Inspections (requirements, contracts, bids) will be the most profitable.

SUMMARY

The art of Inspection has progressed considerably since it was first publicly documented by IBM. It has shifted focus from clean up to sampling, measurement, exit control, and defect injection prevention. By taking the technical points that made inspection strong at IBM and elsewhere and combining them with the recent process improvements, inspection will continue to be a powerful software process tool. Ignoring the process improvements makes it likely that one will end up with a costly failure of a process.

ACKNOWLEDGMENTS

Thanks are due to Lindsey Brodie for helping edit this article.

REFERENCES

- Fagan, M. 1976. Design and code inspections to reduce errors in program development. *IBM Systems Journal* 15, no. 3: 182-211. (Reprinted in *IBM Systems Journal* 38, no. 2: 259-287 or see URL document www.almaden.ibm.com/journal).
- Gilb, T., and D. Graham. 1993. *Software inspection*. London: Addison-Wesley Longman.
- Haley, T., B. Ireland, E. Wojtaszek, D. Nash, and R. Dion. 1995. *Raytheon electronic systems experience in software process improvement* (CMU/SEI-95-TR-017). Pittsburgh: Software Engineering Institute, Carnegie Mellon University. (or see URL document www.sei.cmu.edu/products/publications/documents/95.reports/95.tr.017.html)
- Holland, D. 1999. Document inspection as an agent of change. *Software Quality Professional* (December): 22-33. (See also chapter 5 of Jarvis and Hayes, eds. 1999. *Dare to be excellent*. Englewood Cliffs, N. J.: Prentice Hall PTR).
- Holland, D. 1994. See URL document www.pimsl.com/inforserver/public/spi/index.htm.
- Humphrey, W. 1995. *A discipline for software engineering*. New York: Addison-Wesley.
- Kaplan, C., R. Clark, and V. Tang. 1994. *Secrets of software quality: 40 innovations from IBM*. New

York: McGraw Hill.

Kelly, J. C. 1990. An analysis of Jet Propulsion Laboratory's two year experience with software inspections. In *Proceedings of the Minnowbrook Workshop on Software Engineering*, Blue Lake, N. Y.

Kelly, J. 1990. An analysis of defect density found during software inspection. In *Proceedings of 15th Annual Software Engineering Workshop* (NASA SEL-90-006), Jet Propulsion Labs, Pasadena, Calif.

Kohli, O. R., and R. A. Radice. 1976. Low-level design inspection specification. *IBM Technical Report* (TR 21.629). Armonk, N. Y.: IBM.

Mills, D., and Linger. 1987. Cleanroom software engineering. *IEEE Software* (September): 19-25.

Mills, H. D. 1972. *Mathematical foundations for structured programming*. (FSC 71-6012). Bethesda, Md.: IBM Corporation Federal Systems Division.

Pence, J. L. P., and S. E. Hon III. 1993. Building software quality into telecommunications network systems. *Quality Progress*. (October): 95-97.

Radice, R. A., J. T. Harding, P. E. Munnis, and R. W. Philips. 1999. A programming process study.

IBM System Journals 2 and 3.

Radice, R. A. and Phillips. 1988. *Software engineering, an industrial approach, vol. 1*. Englewood Cliffs, N. J.: Prentice Hall.

Raytheon. 1995. *The Raytheon report*. URL document

www.sei.cmu.edu/products/publications/documents/95.reports/95.tr.017.html. Also see URL document www.Result-Planning.com and www.stsc.hill.af.mil/SWTesting.

Software Development Technologies. 1997. *Software Inspections Automation*, Edward Kit, URL document www.stdcorp.com.

Weller, Ed. F. 1993. Lessons from three years of inspection data. *IEEE Software* (September): 38-45.

**CMM is a trademark of Carnegie Mellon University.

BIOGRAPHY

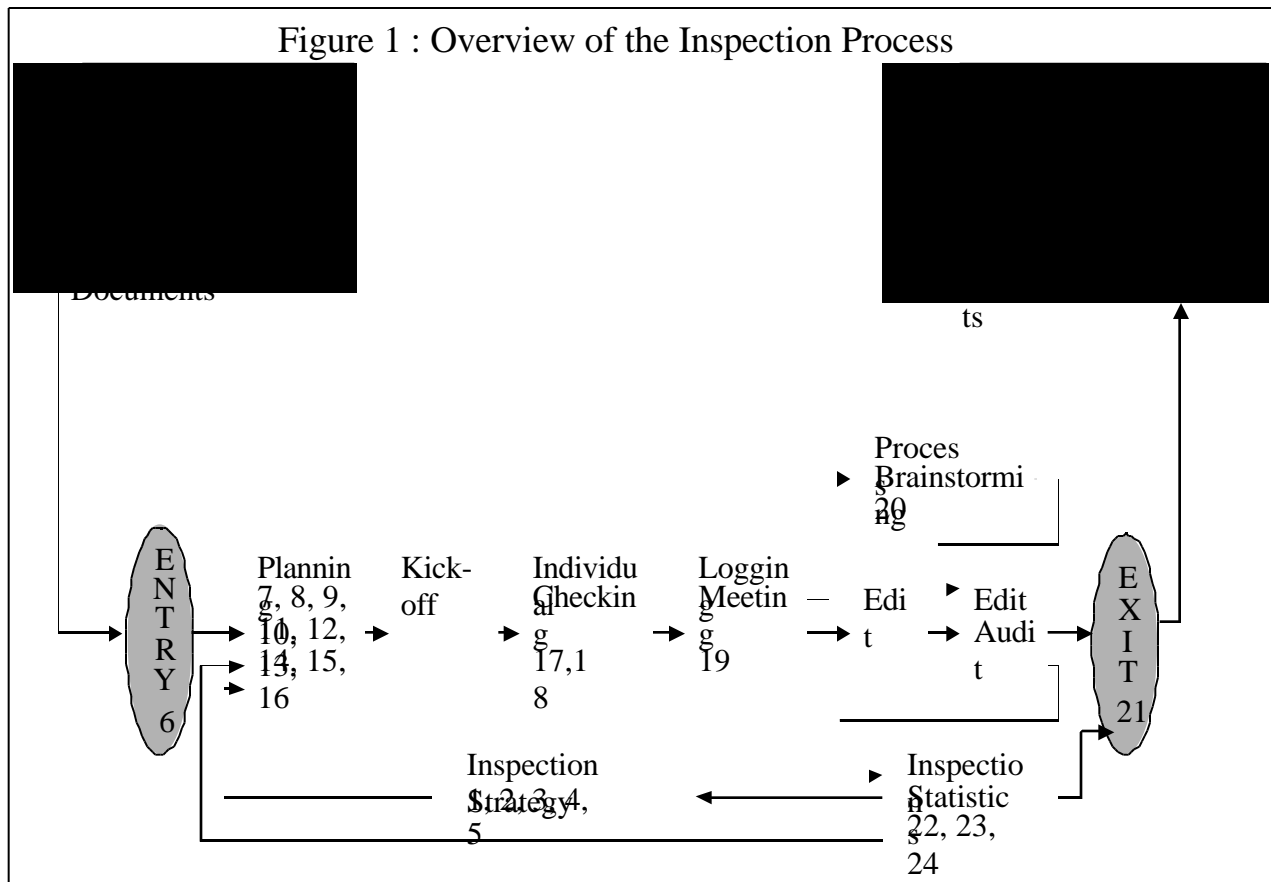
Tom Gilb has been an independent consultant since 1960. He is the author of several books, including *Principles of Software Engineering Management* (1988) and *Software Inspection* (1993 (with Dorothy Graham)). Gilb immigrated to Europe from his birthplace, California, in 1956 and

joined IBM Norway in 1958. He spends approximately half his time working in the United States and half in Europe. Gilb can be contacted at Iver Holtersvei 2, N-1410 Kolbotn, Norway, or by e-mail at Gilb@ACM.org.

NOTE for this Word paper Jan 12 2000 the figure below is a bit corrupted in the word version.

Please live with it! TG

Figure 1 : Overview of the Inspection Process



Notes for Figure 1:

This diagram shows a simplified overview of the Inspection process. The numbers in the process boxes refer to the numbers of the key tips for improving Inspection. Some key tips could have appeared in several process boxes; for simplicity, each was put where it was considered to have its main impact.

The processes within Inspection are:

Inspection strategy: An organization has to understand its aims for Inspection. There has to be strong senior management support. Inspection strategy produces the tactics the organization must adopt to achieve its inspection goals.

Entry: The team leader ensures that the entry conditions for Inspection are met. This involves obtaining the product document, source documents, and kin documents, and checking them against the entry criteria.

Planning: The team leader produces the master plan for an Inspection. Basically, the team leader decides what material within the product document is to be inspected, what documents are to be included within the Inspection, what rules must be used, who is going to be on the Inspection team, and what their roles are. He or she also decides the optimum checking rate.

Kick off: The team leader ensures the Inspection team understands its task.

Individual checking: The team members individually carry out their checking roles. They report to the team leader any serious problems they discover that might impact the Inspection process, for example, number of issues (potential defects) discovered is sufficiently large to consider abandoning

the inspection.

Logging meeting: Inspection team members meet to report their issues and possibly carry out further group checking. There must not be discussion concerning the issues discovered--just logging of the issue (rule violation and position in the documentation).

Process brainstorming: The Inspection team brainstorms the causes and suggests preventive cures for several of the defects logged during the logging meeting.

Edit: The editor (usually the author) examines the logged issues, determines how to resolve them, and fixes the issues that are considered as defects.

Edit audit: The team leader audits the results of the edit process.

Exit: The team leader updates the database of Inspection statistics and decides whether the exit conditions are fulfilled.

FIGURE 2 List of key tips for improving the Inspection process

Inspection strategy

1. Do not misuse Inspection as a clean-up process. Use it to motivate, teach, measure, control quality, and improve your processes.
2. Use Inspection on any technical documentation.
3. Inspect upstream first.
4. Make sure there are excellent standards to identify defective practices.
5. Give inspection team leaders proper training and coaching after.

Entry conditions

6. Use serious entry conditions, for example, numeric quality of source documents.

Planning

7. Plan Inspections well using a master plan.
8. Plan Inspection to address the inspection purposes.
9. Inspect early and often while documents are still being written.
10. Use sampling to understand the quality level of a document.
11. Check against source and kin documents; check them for defects, too.
12. Check the significant portions of the material--avoid checking commentary.
13. Define a major defect as "possible larger costs downstream."
14. Check at an organization's optimum rates to find major defects.
15. Use the optimum number of people on a team to serve the current purpose of Inspection (for example, effectiveness, efficiency, and training).
16. Allocate special defect-searching roles to team members.

Individual checking

17. Use checking data (such as pages checked, majors found, time used, and checking rate) from individual checkers to decide whether it is worth holding a logging meeting.
18. Use the checkers' personal notes instead of proper meeting defect logs when the density of major issues is (nonexit level) high, or when there are many minor defects.

Logging meeting

19. At logging meetings, avoid discussions and avoid suggesting fixes.

Process brainstorming

20. Use the defect prevention process on Inspection itself for continuous improvement.

Exit conditions

21. Use serious exit conditions, for example, "maximum probable remaining major defects per page is 0.2 for exit."

Inspection statistics

22. Build or buy an automated software tool to process inspection basic data.
23. Put the inspection artifacts on a company Web site.
24. Measure the benefit from using Inspections.

FIGURE 3 List of tactics to encourage checkers to focus on major defects rather than minor ones

1. Plan special roles with special role checklists that only ask questions directed at major defects.
2. Teach at kick-off meeting that the search for majors is primary.

3. Hand out checker procedures that define the fact that majors are the primary concern.
4. Use rule sets that for approximately 19 out of 20 rules are identifying major defects.
5. Limit rule sets and checklists to a maximum of one page. This has the effect of squeezing trivial ideas “off the page,” as soon as higher priority major defect identification rules, task activities, and checklist questions are gradually identified.
6. Identify the probable classification of an issue identified in a checklist question next to the question itself as major or minor.
7. Ask checkers to do their own personal classification of major/minor during the checking activity.
8. Ask checkers to report (at the bottom of the inspection-plan form) how many issues they found during checking in the various severity categories. This is done orally with the team. Those who report numerous issues categorized as minors and few as majors will feel motivated to do better next time.
9. Have checkers always state orally “major” or “minor” when they report an issue to be logged at the “public” logging meeting. It’s embarrassing to constantly cite “minor” when others are stating “major.”
10. Calculate Inspection “return on time invested” based on major defects found, never on minors.
11. State numeric Inspection team process improvement objectives in terms of major defects per page and per hour. Never have it a team objective to get better at finding minor defects.
12. Never discuss the root causes of minor defects at the process brainstorming meeting. Only majors are worth discussing.
13. When time for logging is particularly short, report only major defects. If time permits, allow the reporting of minors and formally log them.
14. Report only a symbolic sample of minors, for example, on a single page. The rest, if checkers have notes of them, can be handed informally to the editor. They are simply not worth more formal treatment or priority.
15. If editing time is under pressure, only major defects might actually be fixed.
16. If edit audit has an overwhelming number of things to check, then the leader would take majors

seriously and skim over the minors, perhaps looking at a sample only.

17. Report Inspection statistics to management based only on major defects.

18. Determine optimum rates of checking that ignore minors and use only majors found per hour per page as the basis for calculation of rates.

Note: This list is not necessarily complete.

FIGURE 4 Purposes for Inspection

1. Reducing time to delivery
2. Measuring document quality
3. Measuring the quality of the process producing the document
4. Enabling estimation of the number of remaining defects
5. Identifying defects
6. Removing defects
7. Preventing additional downstream defects being generated by removing existing defects
8. Improving the document production process
9. Improving the inspection process
10. On-the-job training for the checkers
11. Training the team leader
12. Certifying the team leader
13. Peer motivation
14. Motivating the managers
15. Helping the author
16. Reinforcing conformance to standards
17. Capturing and reusing expert knowledge (by use of rules and checklists)
18. Reducing costs
19. Team building
20. Fun – a social occasion

Note: This list is not necessarily complete.

FIGURE 5 This diagram is from the Raytheon Report (1995). It shows that their optimum checking rate is probably about 10 to 20 statements per hour for an individual checker.

