Chapter
8

SPECIFICATION QUALITY CONTROL

How to know how well you specified

GLOSSARY CONCEPTS

Specification Quality Control (SQC)
Defect Detection Process (DDP)
Defect Prevention Process (DPP)
Specification
Source
Kin
Checklist
Issue
Major Defect
Minor Defect
Checking Rate

8.1 Introduction: Specification Quality Control

Specification Quality Control (SQC) is the name I shall use to refer to this method in this text. Within the software community, the term 'Inspection' is used. However, it is a poor choice for engineering communities, which already use 'inspection' in another sense during final production line quality control. SQC is remote from such assembly-line inspections, as it takes place from the earliest stages of idea specification and has different organizational impacts (for example, team building and assisting in 'on the job' training).

The primary purpose of SQC is systems engineering process control through sampling measurement of specification quality. Through SQC, we can improve systems engineering processes, save project time and increase systems engineering productivity.

Improving Process

Control of projects, designs, strategies, marketing, selling and buying, management planning, and programming, all have one thing in common at least – they rely on ideas *specified* by people, and *read* by people. If those ideas are misunderstood by the reader, incomplete, wrongly written or out of date, then we are doomed to lose control and be less competitive, no matter how well we design, plan and implement!

For *software*, studies have long since shown that a considerable percentage (44% at Bellcore (Pence and Hon 1993) and 62% (Thayer, Lipow and Nelson 1978)) of all bugs in computer programs were not due to faulty programming. They were due to faulty requirements and design being handed to the programmers and the testers. In many cases, the testers, unwittingly, checked that an erroneous specification was 'correctly' programmed! Testing, in this situation, does not solve the problem: it confirms it. However, SQC can address such problems.

In *aircraft design* at Douglas Aircraft (now Boeing), 'engineering order' faults cost $2,965 each to correct and 30% of engineering orders needed correction. After SQC was applied in 1987–88, the percentage of faulty engineering orders fell to 0.5% (Personal Experience). We achieved similar results in 1989 at Boeing, Renton on all aspects of aircraft design.

The tendency to commit some kind of error, when communicating complex ideas in writing to other people, is much worse than most people realize. My own experience in industrial measurement of defects suggests that technical documents, initially and routinely, contain at least 20–60, and often far more, 'major' engineering specification defects in each 'logical' page (300 non-commentary words). Through systematic use of feedback from SQC to specification writers, this level can be brought down to well under one remaining major defect/page (British Aerospace, Eurofighter Project, Wharton, achieved this in 18 months (Personal Communication)).

Saving Time

Without SQC, a major defect left in a technical specification can cost an average of 9.3 work-hours to deal with.[1] Use of SQC at an early stage (during writing the specification) would cost only *one* work hour to remove it. "A stitch in time saves nine" (or an SQC hour saves nine-point-three to be exact! (Gilb and Graham 1993)).

Increasing Productivity

The reduction of defects (as a result of using SQC) saves 'rework', which is otherwise about half of all effort in software projects. Raytheon (Haley et al. 1995) found that software engineering productivity for about a thousand programmers increased by a factor of 2.7 over a few years of using SQC (Inspection and Defect Prevention Process).

One major reason for defect reduction is the 'training effect' of SQC on individuals. The number of defects injected by a systems engineer reduces by about 50% each time they go through an SQC process (Personal Experience since 1988). Systems engineers rapidly learn to take the rules seriously. They see that their peers expect them to comply with the rules and that their work cannot exit, and be 'finished', until they reach at least the exit level for the estimated number of remaining major defects. I have found that this is as true in software as it is in hardware engineering.

Industrial Usage

The methods needed for quality control (QC) of specifications originated in the early 1970s within IBM, when they were used under the name of 'Design and Code Inspections'.[2] Since then significant changes have occurred, resulting in the SQC method described in this book. The most notable change was the introduction, again within IBM, of the Defect Prevention Process (DPP) (Mays 1995). The other major change is the shift to 'sampling', rather than 100% checking and trying to clean up defects.

[1] As measured on a 1,000 defect sample by (then) Thorn EMI (electronics industry) in 1990. See Section 8.8 and (Gilb and Graham 1993 Page 315: Reeve).

[2] Fagan, M. E. 1976. Design and code inspections. *IBM Systems Journal*. Volume 15. Number 3. Pages 182–211. Reprinted 1999. *IBM Systems Journal*. Volume 38. Numbers 2 and 3. Pages 259–287. See http://www.research.ibm.com/journals/sj/

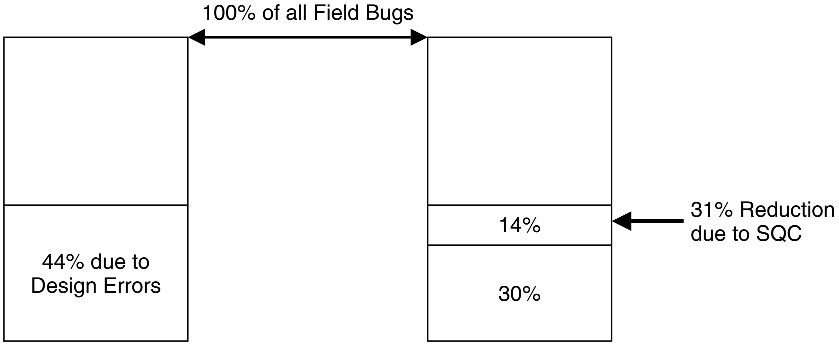


Figure 8.1
Due to use of SQC during development of telecommunications software, a 31% reduction in design errors that caused bugs in the field was measured after 2 releases (Pence and Hon 1993).

The first large-scale *hardware* engineering uses of SQC took place at Douglas Aircraft (1988) and Boeing (1989) under this author's guidance. In recent years, Siemens, Alcatel and Ericsson have also successfully used the method on a large scale (hundreds trained) for total product development purposes. Hewlett Packard has reported estimated savings due to SQC (some use within hardware product planning) of $21.5 million and $34 million in 1993 and 1994 respectively (Grady and Van Slack 1994).

The use of SQC *outside* of the software area is, as yet, little understood or appreciated, except by the few corporations who have tried it out such as Ericsson, Douglas and Boeing. It is time that this industrial experience was more widespread knowledge. There is little difference in the specification of software engineering, management planning or hardware engineering with regard to human specification errors, their causes and their consequences.

8.2 Practical Example: Specification Quality Control

Take the simple performance requirement statement:

'The objective is to get higher adaptability using modular structure.'

Do you see any problems with it? Is it similar to statements you see every day? Well, if you have read this book this far, you would notice that it violates some rules we have suggested. Of course, there is nothing wrong with it, unless we agree that these rules are in force. For some purposes they should be in force, for others not.

SQC works by using the formal rules that are in force: a ‘defect’ is a rule violation. SQC discovers whether people have applied the agreed rules or not. A specification writer must always know the rules that apply (and have agreed to them in advance). The specification writer should welcome any help to follow them. Rules, after all, should be ‘best practice’ rules.

Let us now (for the sake of this example) introduce a few short rules, which apply to the quality requirement statement above.

Rules For Performance Requirements

Tag: Rules.OBJ.

Clear: They must be *unambiguously* clear to the intended readers (not to ‘anyone,’ just the relevant people).

Detail: They must detail complex concepts as a set of elementary measurable concepts.

Scale: They must specify a scale of measure to define the concept (all performance attributes are quantifiable).

Quantify: They must specify at least two points of reference on the defined Scale to define ‘relative’ terms, such as ‘‘higher.’’ These are called the benchmark and target specifications.

Qualify: Targets must specify exactly ‘when’ a performance level is to be available. Other qualifier notions, such as ‘where’ and ‘if’ should also be made explicit, if the target is not elsewhere specified.

Ends: They must not put ‘designs’ in the specification of ‘performance requirements.’ Specify the *Ends*, not the *Means*.

Source: The source statements for each requirement must be precisely referenced (for example, <- the contract and marketing documentation).

Fuzzy: Fuzzy unclear concepts shall be marked with <fuzzy/angle brackets> to indicate there is room for improvement.

A checker (a person assigned to check a specification and its selected source documents against these rules) would be obliged to report, for the performance requirement statement about ‘higher adaptability’, that all the above rules were violated.

There are, therefore, at least eight defects in the requirement statement. If these defects *might* have much higher costs later in a project (if not fixed at specification time), they should be classed as ‘major’ defects. Majors are the defects it pays to fix now, at a tenth of the cost we would otherwise suffer later. (Fixing majors early is useful, but preventing their injection is even more profitable.)

Checkers are friendly, confidential personal advisors to the specification writer. The checker’s first job is to point out potential problems for correction before a specification is released to other engineers, or to customers. Checking is a service the writer will likely perform, in return, when their former peer checkers specify something themselves, and want SQC help. The responsible engineer will take a list of the checker's suggested advice regarding 'potential defects' (issues) and consider correcting them. They should address similar defects, outside the sample checked, as far as necessary, according to the applicable rules, procedures and source documents. However, it may pay off to totally rewrite the specification. The specification document 'Exit Level' is based on a general calculation of what is the best project time-saver. We don't exit, if cleaning up the specification now saves the most time, in the long run. **The following are the expected results of a *single pass* of SQC**:

(Note: Multiple passes should be rare.)

1. Based on defects found and corrected and, on an assumed SQC effectiveness at spotting defects of 50%, a calculation will be made about the (probable) remaining major defects in the specification (which is about as many defects as we found – since we cannot expect to be much better than 50% effective in finding defects). If these are more than permitted by the exit conditions, the specification will not be released. This is because the estimated unfound remaining majors would cause more loss of time than savings to be gained, if we let them exit downstream; that is, if we released the specification immediately.
2. The specification writer will learn about current agreed rules and their peers' interpretations of these rules. As a result they are likely, by my industrial experience, to learn to produce a specification with half the number of defects next time. (Ultimately, after several SQC experiences for the writer, about 100 times cleaner – using major defect reduction as the measure – specification is usually achieved!)
3. The checkers themselves will learn best practice rules and their peers' attitudes towards those practices. This will influence the checkers' specification work quality.
4. The 'users' of the specification will learn to expect (in terms of their entry condition) a minimum specification quality level (such as no more than one remaining major defect/page).
5. The SQC team will continuously suggest process improvements to reduce future major defects. (Poor working processes, training, tools and the working environment 'force' defects on the workforce according to Deming (Deming 1986)).
6. Project productivity will at least *double*, mainly due to fighting fewer defects later (Dion has reported productivity increasing by a factor of 2.7 (Dion 1993; Haley et al. 1995)).

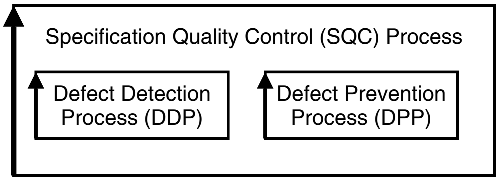
As a result of SQC we will have data to decide if it pays off to release the specification to another engineering process, or fight the defects now.

8.3 Language Core: Specification Quality Control

Basic Definitions (see also Glossary terms)

Specification Quality Control (SQC)

Specification Quality Control (SQC) consists of two main processes: the Defect Detection Process (DDP) and the Defect Prevention Process (DPP).


Specification Quality Control (SQC) Process
Defect Detection Process (DDP)
Defect Prevention Process (DPP)

Defect Detection Process

The Defect Detection Process is concerned with document quality, mainly with identifying defects in the documentation and using this information to make decisions about how best to proceed with the main document under SQC – the main specification.

Ideally, though sometimes not done due to the economics of the situation, a known defect must be removed as soon as possible after the error has been committed. This is to avoid the high cost of late removal (at test or in field) of the defect, or to avoid the high cost of its consequences. "*A stitch in time saves nine.*"

Defect Prevention Process

The Defect Prevention Process is concerned with learning from the defects found and suggesting ways of improving processes to prevent them reoccurring in future. The process improvement suggestions are routed on to the relevant process owner for further consideration. "*An ounce of prevention is worth a pound of cure.*"

Here are some other basic SQC concepts.

Issue

An issue is a *perceived* defect in a document. It is a non-confrontational way for a checker to say, "I think I may have identified a defect."

Defect

A failure to observe a formal, written, required rule. It is not a personal opinion or personal taste. It is failure to observe a group norm, or required best practice.

Major defect

A major defect is a defect (rule violation) which, if not fixed at the requirements or design stage of specification, will possibly grow approximately an order of magnitude or larger in 'cost-to-find-and-fix' and/or damage potential. It is often intentionally written with a capital 'M'. Minor defects tend not to be economic to identify or fix (*but you sometimes have to identify them to determine that they are indeed minor and not, major*).

Page

A logical page, as opposed to a physical page, is defined as a specific number of non-commentary words. If no other definition is given then use '300 non-commentary words' for each logical page (default 'volume' definition). This ensures measurements of checking rates and defect densities are consistent.

Checking Rate

The checking rate is the average speed with which an individual checker searches a specification for defects, allowing time for checking it against rules, sources, kin documents and checklists. This is a critical factor to control for effective checking. You have to go surprisingly slowly to raise your checking effectiveness from 5% to 50%. (For example: one page an hour!)

Optimum Checking Rate

The optimum checking rate is the rate, which gives the highest checking productivity (effectiveness in finding majors). It is the checking speed that in fact works best on a given document type for an individual checker to do their assigned tasks. It is found by establishing the most effective average historical checking rate in terms of finding major defects. The optimum checking rate is usually in the range of 300 non-commentary words/hour (plus 300/minus 270). This is used as a guide for team planning. Individuals need to tune in to their personal optimum rate, which varies from this average.

The major trick to going at this 'slow' rate is to *sample*, not to attempt 100% checking of all pages and consequent 'defect clean up'.

Remaining Major Defects

The remaining major defects are the *estimated* remaining major defects/'volume' measure (which could be a page, a technical drawing or an entire specification) given for a sample or an entire specification. It is estimated based on 'total found' and 'known % effectiveness.'

Checklist

A 'checklist' is a list of questions, which can be asked about a document's contents by a checker, with a view to improving the effectiveness of that checker in finding major defects. Checklist questions are always directly derived from individual official rules. They are not allowed to be the rules, or to change the rules, just to *interpret* them.

Rule

A rule is a standard for the production of a *written* process output. A rule can be used to judge the objective quality ('defect-freeness' according to current rules) of a written process output. Violations of rules are defined as 'defects.'

Rules are often grouped into sets according to the type of standard, which they are setting (for example, 'specification clarity' or 'specification consistency').

Main Specification

The main specification is one of potentially *many* documents involved in a single SQC. However, it distinguishes itself as the one we are trying to get formal 'exit' for. Exit (acceptable exit level) is based primarily on the specification's quality with respect to the official systems engineering standards (rules) for writing it.

Source Documents

The source documents are the 'parents' used to produce a specific main specification. For example, contracts are typical sources for requirements. Requirements are a source for design. Requirements and design are sources for Impact Estimation. Design is source for planning, estimating and construction or programming. Change requests are sources for an updated specification.

Kin Documents

'Offspring of the same 'parent' (source) documents are 'kin.' *For example, test plans, source code and user handbooks could all be derived from the same requirements or the same design.* The use of kin documents is that they can serve as information to perform defect checking in SQC.

8.4 Standards: Specification Quality Control

Rules are standards, and are central to the SQC process; specifications must be checked against their agreed specification rules. However, the rules to be used depend on the specification type, so we won't attempt to list them here. The rules given in other chapters of this book are suitable examples of such rules (*but they are by no means a complete list*).

Here is a list of guidelines for assessing whether your overall SQC process is functioning correctly.

Guidelines for assessing functioning of overall SQC

Economic: SQC must always make economic sense. If SQC is not saving in the order of 10 hours for every hour spent on SQC, then your SQC process should probably be modified or abandoned.

SQC Champion: There must be an SQC champion within the organization. (At the very least, a nominated person responsible for SQC; an SQC process owner.)

Team Leaders: There must be a list of current SQC team leaders. It should show that there is a sufficient number of team leaders within the organization and also that the team leaders are trained, tested and 'certified' to ensure they know what they are doing.

Statistics: The SQC statistics must be up-to-date on the SQC database.

Meetings: All meetings must be of maximum length of two hours (tiredness reasons). If more time is needed, schedule a set of such meetings (*but do consider the possibility of using sampling*).

Checkers: Unless you are training novices, the number of checkers at a meeting should be five or less. Two or three people is typically most cost-effective, Four to five is generally more 'effective.'[3]

Checking Rate: All checking must be carried out near the relevant optimum checking rate. This rate will vary by document type and organization. It is about 1 page/hour.

Condition: Entry and exit conditions must be taken seriously. They are there to save you wasting time. The number of remaining major defects/page for successful exit must be ultimately less than one (major defect/page).

Standards: There must be an up-to-date (intranet) 'library' of official rules, checklists and forms.

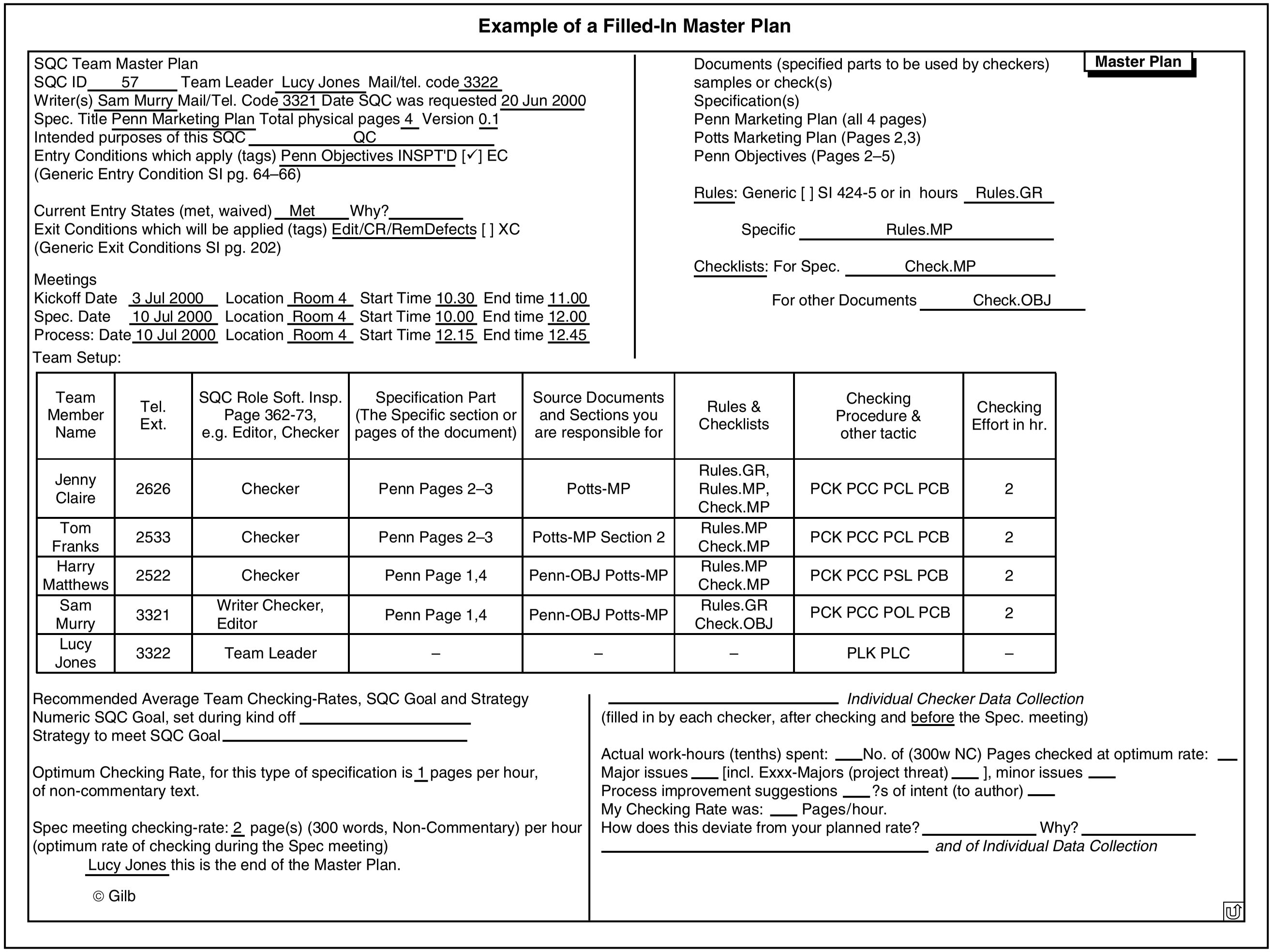
Upstream Pollution: The number of major issues identified by your team in source specifications, which have just previously-exited SQC, should be approximately 15% of the total number of logged issues. Otherwise, this is a sign that your team is not taking the 'second-round' opportunity to find source defects, seriously.

Forms

SQC uses four main forms: the Master Plan, the Editor Advice Log, the Data Summary and the Process Meeting Log. There are examples of these forms filled in, in Figures 8.2, 8.3, 8.4 and 8.5. Blank forms are given in Section 8.9.

Note forms are a 'procedure' (in the format of the form) for gathering data. Most of our clients have their own local variation of the forms and automate them (usually on an intranet web site).

[3] The original evidence for this came from research performed by Søren Nielsen in the Danish electricity industry (Danish Technical Institute, Lyngby, 1987; cited in Gilb and Graham 1993), and was confirmed by further research at Jet Propulsion Labs by John Kelly (Kelly 1990a; 1990b). Optimum effectiveness (number of unique issues per checker) was achieved with teams of 4–6 people, optimum efficiency (cost per unique issue found) with teams of 2–4 people. The recommended team size of 4–5 people achieves the best compromise between these factors. It was Edward Weller, analysing data from more than 6,000 inspection meetings conducted at Bull HN (Weller 1993), who reported that "four-person teams were twice as effective . . . as three person teams." Also included in Wheeler, Brykczynski and Meeson (1996).



Example of a Filled-In Master Plan

Master Plan

SQC Team Master Plan
SQC ID 57 Team Leader Lucy Jones Mail/tel. code 3322
Writer(s) Sam Murry Mail/Tel. Code 3321 Date SQC was requested 20 Jun 2000
Spec. Title Penn Marketing Plan Total physical pages 4 Version 0.1
Intended purposes of this SQC QC
Entry Conditions which apply (tags) Penn Objectives INSPT'D [✓] EC
(Generic Entry Condition SI pg. 64–66)

Current Entry States (met, waived) Met Why? ____
Exit Conditions which will be applied (tags) Edit/CR/RemDefects [ ] XC
(Generic Exit Conditions SI pg. 202)

Meetings
Kickoff Date 3 Jul 2000 Location Room 4 Start Time 10.30 End time 11.00
Spec. Date 10 Jul 2000 Location Room 4 Start Time 10.00 End time 12.00
Process: Date 10 Jul 2000 Location Room 4 Start Time 12.15 End time 12.45

Documents (specified parts to be used by checkers)
samples or check(s)
Specification(s)
Penn Marketing Plan (all 4 pages)
Potts Marketing Plan (Pages 2,3)
Penn Objectives (Pages 2–5)

Rules: Generic [ ] SI 424-5 or in hours Rules.GR

Specific Rules.MP

Checklists: For Spec. Check.MP

For other Documents Check.OBJ

Team Setup:

Team Member Name	Tel. Ext.	SQC Role Soft. Insp. Page 362-73, e.g. Editor, Checker	Specification Part (The Specific section or pages of the document)	Source Documents and Sections you are responsible for	Rules & Checklists	Checking Procedure & other tactic	Checking Effort in hr.
Jenny Claire	2626	Checker	Penn Pages 2–3	Potts-MP	Rules.GR, Rules.MP, Check.MP	PCK PCC PCL PCB	2
Tom Franks	2533	Checker	Penn Pages 2–3	Potts-MP Section 2	Rules.MP Check.MP	PCK PCC PCL PCB	2
Harry Matthews	2522	Checker	Penn Page 1,4	Penn-OBJ Potts-MP	Rules.MP Check.MP	PCK PCC PSL PCB	2
Sam Murry	3321	Writer Checker, Editor	Penn Page 1,4	Penn-OBJ Potts-MP	Rules.GR Check.OBJ	PCK PCC POL PCB	2
Lucy Jones	3322	Team Leader	–	–	–	PLK PLC	–

Recommended Average Team Checking-Rates, SQC Goal and Strategy
Numeric SQC Goal, set during kind off ____
Strategy to meet SQC Goal ____

Optimum Checking Rate, for this type of specification is 1 pages per hour, of non-commentary text.

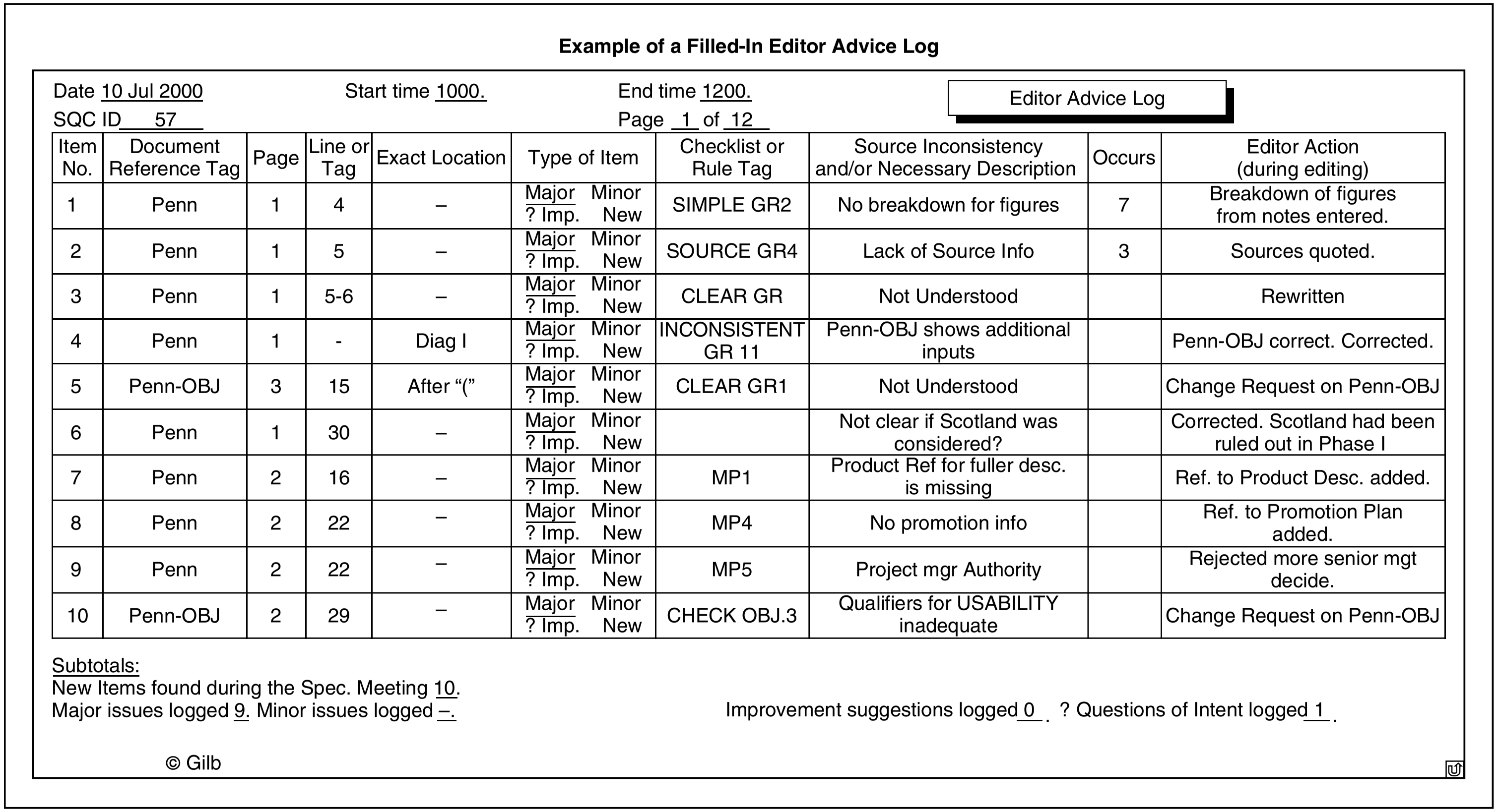
Spec meeting checking-rate: 2 page(s) (300 words, Non-Commentary) per hour
(optimum rate of checking during the Spec meeting)
Lucy Jones this is the end of the Master Plan.

© Gilb

____ *Individual Checker Data Collection*
(filled in by each checker, after checking and before the Spec. meeting)

Actual work-hours (tenths) spent: ___No. of (300w NC) Pages checked at optimum rate: ___
Major issues ___ [incl. Exxx-Majors (project threat) ___], minor issues ___
Process improvement suggestions ___?s of intent (to author) ___
My Checking Rate was: ___ Pages/hour.
How does this deviate from your planned rate? ____ Why? ____
____ *and of Individual Data Collection*

Figure 8.2 Filled-in Master Plan.



Example of a Filled-In Editor Advice Log

Date 10 Jul 2000 Start time 1000. End time 1200.

SQC ID 57 Page 1 of 12

Editor Advice Log

Item No.	Document Reference Tag	Page	Line or Tag	Exact Location	Type of Item	Checklist or Rule Tag	Source Inconsistency and/or Necessary Description	Occurs	Editor Action (during editing)
1	Penn	1	4	–	Major Minor ? Imp. New	SIMPLE GR2	No breakdown for figures	7	Breakdown of figures from notes entered.
2	Penn	1	5	–	Major Minor ? Imp. New	SOURCE GR4	Lack of Source Info	3	Sources quoted.
3	Penn	1	5-6	–	Major Minor ? Imp. New	CLEAR GR	Not Understood		Rewritten
4	Penn	1	-	Diag I	Major Minor ? Imp. New	INCONSISTENT GR 11	Penn-OBJ shows additional inputs		Penn-OBJ correct. Corrected.
5	Penn-OBJ	3	15	After "("	Major Minor ? Imp. New	CLEAR GR1	Not Understood		Change Request on Penn-OBJ
6	Penn	1	30	–	Major Minor ? Imp. New		Not clear if Scotland was considered?		Corrected. Scotland had been ruled out in Phase I
7	Penn	2	16	–	Major Minor ? Imp. New	MP1	Product Ref for fuller desc. is missing		Ref. to Product Desc. added.
8	Penn	2	22	–	Major Minor ? Imp. New	MP4	No promotion info		Ref. to Promotion Plan added.
9	Penn	2	22	–	Major Minor ? Imp. New	MP5	Project mgr Authority		Rejected more senior mgt decide.
10	Penn-OBJ	2	29	–	Major Minor ? Imp. New	CHECK OBJ.3	Qualifiers for USABILITY inadequate		Change Request on Penn-OBJ

Subtotals:

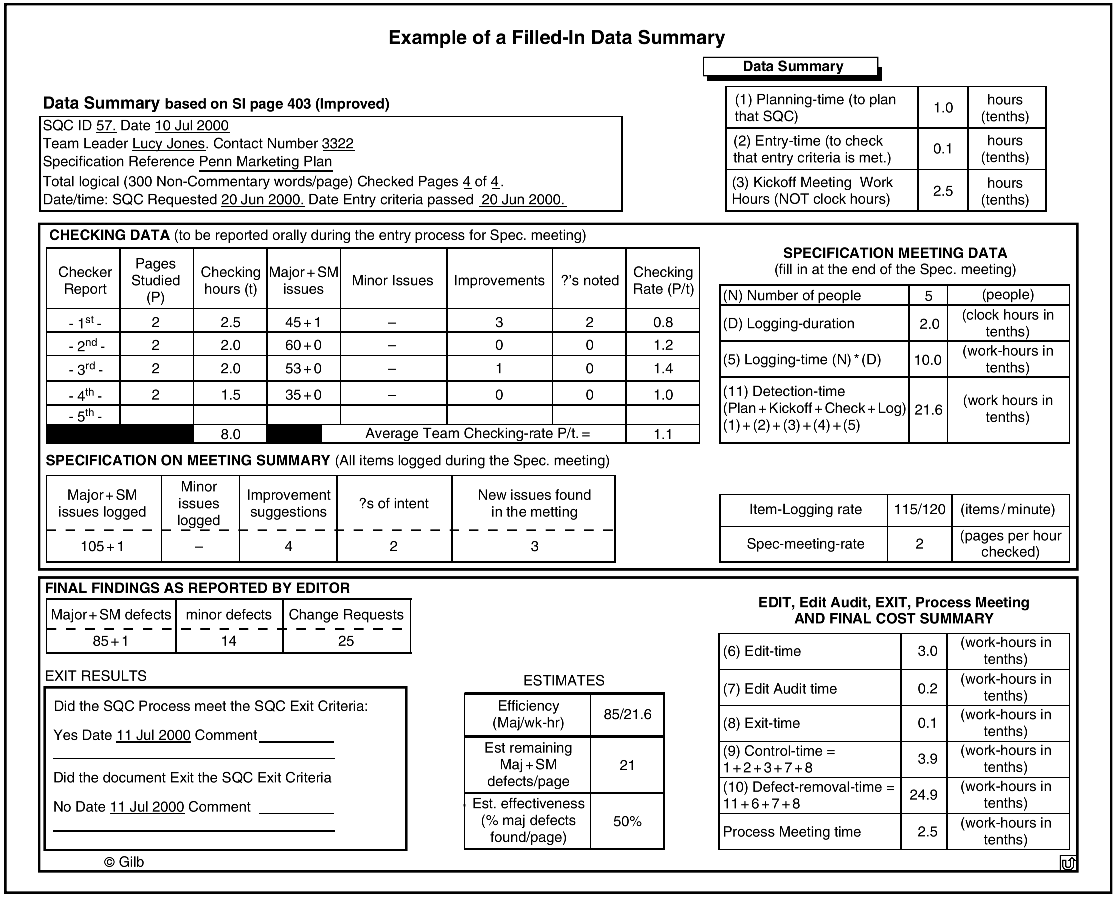
New Items found during the Spec. Meeting 10.

Major issues logged 9. Minor issues logged –.

Improvement suggestions logged 0 . ? Questions of Intent logged 1 .

© Gilb

Figure 8.3 Filled-in Editor Advice Log.



Example of a Filled-In Data Summary

Data Summary

(1) Planning-time (to plan that SQC)	1.0	hours (tenths)
(2) Entry-time (to check that entry criteria is met.)	0.1	hours (tenths)
(3) Kickoff Meeting Work Hours (NOT clock hours)	2.5	hours (tenths)

Data Summary based on SI page 403 (Improved)

SQC ID 57. Date 10 Jul 2000
Team Leader Lucy Jones. Contact Number 3322
Specification Reference Penn Marketing Plan
Total logical (300 Non-Commentary words/page) Checked Pages 4 of 4.
Date/time: SQC Requested 20 Jun 2000. Date Entry criteria passed 20 Jun 2000.

CHECKING DATA (to be reported orally during the entry process for Spec. meeting)

Checker Report	Pages Studied (P)	Checking hours (t)	Major + SM issues	Minor Issues	Improvements	?'s noted	Checking Rate (P/t)
- 1st -	2	2.5	45 + 1	–	3	2	0.8
- 2nd -	2	2.0	60 + 0	–	0	0	1.2
- 3rd -	2	2.0	53 + 0	–	1	0	1.4
- 4th -	2	1.5	35 + 0	–	0	0	1.0
- 5th -							
		8.0		Average Team Checking-rate P/t. =			1.1

SPECIFICATION MEETING DATA
(fill in at the end of the Spec. meeting)

(N) Number of people	5	(people)
(D) Logging-duration	2.0	(clock hours in tenths)
(5) Logging-time (N) * (D)	10.0	(work-hours in tenths)
(11) Detection-time (Plan + Kickoff + Check + Log) (1) + (2) + (3) + (4) + (5)	21.6	(work hours in tenths)

SPECIFICATION ON MEETING SUMMARY (All items logged during the Spec. meeting)

Major + SM issues logged	Minor issues logged	Improvement suggestions	?s of intent	New issues found in the metting
105 + 1	–	4	2	3

Item-Logging rate	115/120	(items / minute)
Spec-meeting-rate	2	(pages per hour checked)

FINAL FINDINGS AS REPORTED BY EDITOR

Major + SM defects	minor defects	Change Requests
85 + 1	14	25

EXIT RESULTS

Did the SQC Process meet the SQC Exit Criteria:

Yes Date 11 Jul 2000 Comment _________

Did the document Exit the SQC Exit Criteria

No Date 11 Jul 2000 Comment _________

ESTIMATES

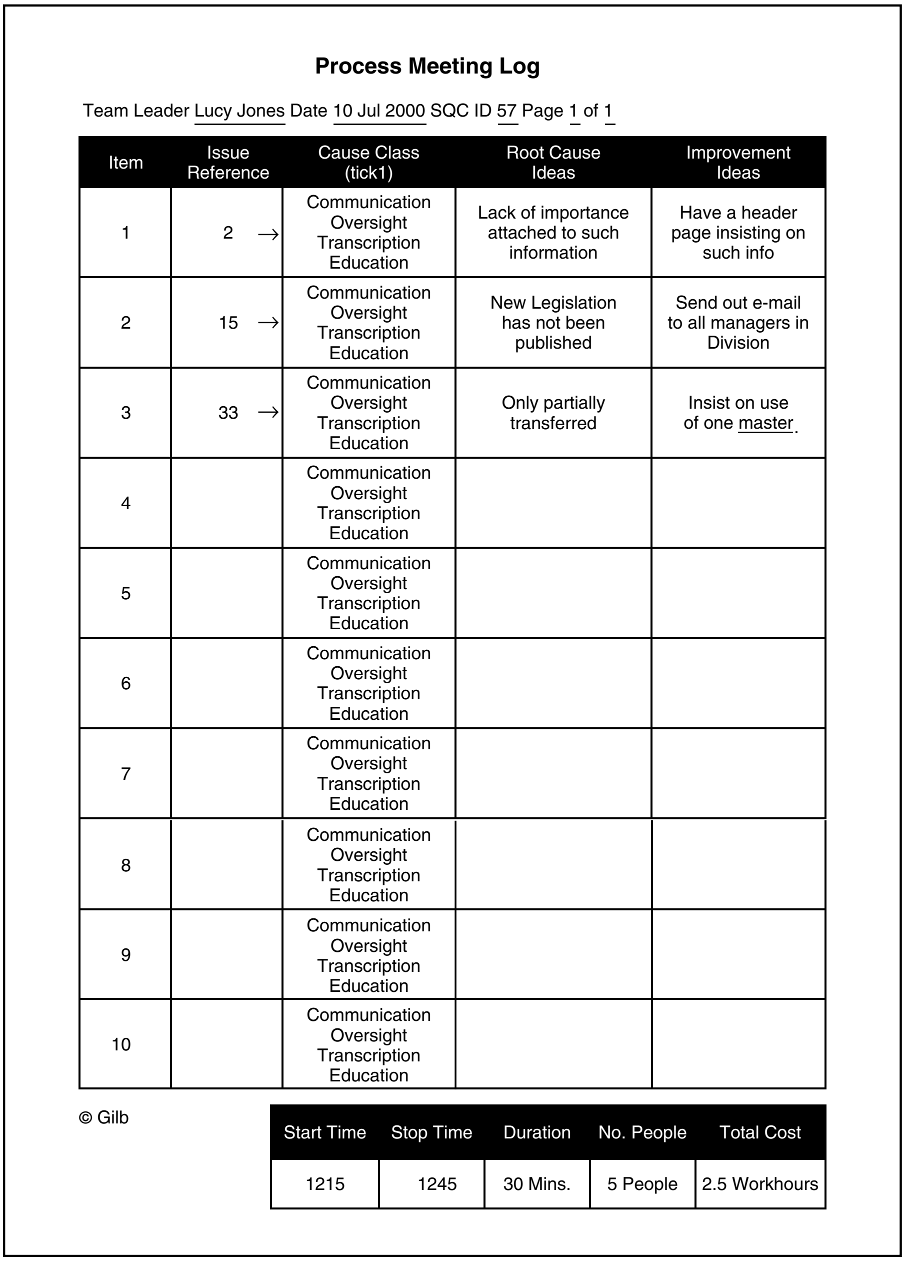
Efficiency (Maj/wk-hr)	85/21.6
Est remaining Maj + SM defects/page	21
Est. effectiveness (% maj defects found/page)	50%

EDIT, Edit Audit, EXIT, Process Meeting AND FINAL COST SUMMARY

(6) Edit-time	3.0	(work-hours in tenths)
(7) Edit Audit time	0.2	(work-hours in tenths)
(8) Exit-time	0.1	(work-hours in tenths)
(9) Control-time = 1 + 2 + 3 + 7 + 8	3.9	(work-hours in tenths)
(10) Defect-removal-time = 11 + 6 + 7 + 8	24.9	(work-hours in tenths)
Process Meeting time	2.5	(work-hours in tenths)

© Gilb

Figure 8.4 Filled-in Data Summary.



Process Meeting Log

Team Leader Lucy Jones Date 10 Jul 2000 SQC ID 57 Page 1 of 1

Item	Issue Reference	Cause Class (tick1)	Root Cause Ideas	Improvement Ideas
1	2 →	Communication Oversight Transcription Education	Lack of importance attached to such information	Have a header page insisting on such info
2	15 →	Communication Oversight Transcription Education	New Legislation has not been published	Send out e-mail to all managers in Division
3	33 →	Communication Oversight Transcription Education	Only partially transferred	Insist on use of one master.
4		Communication Oversight Transcription Education		
5		Communication Oversight Transcription Education		
6		Communication Oversight Transcription Education		
7		Communication Oversight Transcription Education		
8		Communication Oversight Transcription Education		
9		Communication Oversight Transcription Education		
10		Communication Oversight Transcription Education		

© Gilb

Start Time	Stop Time	Duration	No. People	Total Cost
1215	1245	30 Mins.	5 People	2.5 Workhours

Figure 8.5
Filled-in Process Meeting Log.

SQC Process Roles and Responsibilities

An **efficient team** *(most major defects/work-hour) uses 2 or 3 people in total. An* **effective team** *(most major defects/page) uses a maximum of 3 to 5 people in total.*

Team Leader

A team leader is responsible for managing an SQC process. The team leader is responsible for knowing SQC thoroughly and helping the team members to perform. They follow the 'best-practice' SQC processes. An SQC team leader is normally trained for about a week, and is then formally approved to practice by virtue of their practical ability and continued correct practice. Inadequate SQC team leader training leads to failure of the SQC process (Grady and Van Slack 1994).[4]

Checker

Checkers are primarily 'consultants to the writer' and their detailed knowledge of the defectiveness of the writer's work is confidential. Almost all engineering team members work as checkers on occasion, including the writer and probably the team leader. (The team leader might choose to be a 'non-playing captain' of the team. They would not check in order to focus their time on the team leader responsibility or because they were not technically capable in the specification 'language.')

Checkers are SQC team members who actively check a set of documents: the main specification, its source specifications, kin specifications, the rules, checklists and procedures. They focus on using the checklists and rules to find major defects. Exactly which documents a specific checker uses, and what they check for, is determined by the role or roles assigned to them by the team leader.

Checkers are also invited to submit specific comment on possible improvements to the process and the process standards (procedures, rules, entry conditions, exit conditions and forms). They will, hopefully, get some insights during their checking work (for example, about the need for better rules).

[4] Grady reported that HP failed to achieve results from 1983 to 1988 until they properly trained their team leaders on a week-long course (designed and held, as cited there, by this author). This article is reproduced in Wheeler, Brykczynski and Meeson (1996).

Writer: Also Known as Author

The writer is the person currently responsible for writing or updating a specification. The SQC process serves the writer primarily: in confidence. SQC serves the organization secondarily.

Editor

The editor is usually the same person as the writer. The editor is the person, who takes over the issues in the Editor Advice Log, decides (based on standards) what action is required and carries it out. Some issues will be defects and need fixing. Some issues will require clarification. Some issues will be rejected and others will require change requests to other documents to be raised. The Editor Advice Log can be updated with the editor's decisions.[5]

In extreme cases, but unfortunately all too frequently, the defect density found (for example 90 majors in a page) will effectively spell out the fact that 'burning' the work and completely rewriting it will be more cost-effective.

Scribe

The scribe writes up the Editor Advice Log or other team notes at an SQC meeting. This can be any one of the team members. By default, the team leader will scribe. 'Who scribes' is not a critical decision.

Others

In a larger organizational setting, there are 'players' outside the team who support the SQC process. These include quality management, SQC process champions, process owners (for both SQC processes and the work processes, for example, 'Requirement process owner'), senior SQC team leaders, SQC process trainers and engineering data analysts (perhaps specialized in SQC data statistics). When the SQC process is applied to perform specification content reviews, the participants will be senior staff expected to use judgment and to take responsibility for the consequences of their approval.

[5] There are many ways to report what editing action has been undertaken and any suitable method is fine.

8.5 Process Description: Specification Quality Control

Process: Specification Quality Control

Tag: Process.SQC.

Version: October 7, 2004.

Owner: TG.

Status: Draft.

Note: See (Gilb and Graham 1993) for more detail on the sub-processes. All sub-processes are DDP unless marked as DPP.

Entry Conditions

E1: The specification writer must have requested the SQC hoping to get help and exit validation for the specification.

E2: A team leader for the SQC is found from amongst the '*approved* team leaders' group.

E3: All relevant documents (main specification, kin documents, source documents, rules, checklists and forms) are available and ideally have successfully exited SQC – apart from the main specification!

SQC Sub-Processes

Entry

The team leader ensures that the SQC *entry conditions* are met. This includes obtaining the relevant documents and checking their status. Entry conditions are evaluated *during* the Planning phase.

Planning

The team leader produces the master plan for the SQC (about 1 hour's work). This involves deciding what material within the specification is to be sampled, what documents are to be included, what rules must be used, who is going to be on the team and what their roles are. The optimum checking rate is determined based on history.

SQC Strategy

The team leader decides the purpose(s) of this SQC and ensures a suitable overall SQC strategy. Again this is evaluated *during* Planning.

Kickoff

The team meets at a Kickoff meeting, where the team leader makes sure every member knows what they *need to know* about the SQC process and the project documents, and that they are committed to working the plan *as a team*. The team may approve a suggested 'quantified goal' and 'appropriate strategy' to meet it (a DPP component).

Checking

The team members individually carry out their assigned defect-search roles at their self-adjusted optimum checking-rate, looking for major defects. They collect data about the cost and result of their personal checking activity. This process will typically, for a sample of about two logical pages, take two hours for each person. Checkers will ask the team leader for help if necessary. They will also report to the team leader any unusual or serious problems they discover that might impact the future course of the SQC process, for example, that the number of issues (potential defects) discovered is sufficiently large to consider abandoning the SQC.

Specification Meeting

This is a team meeting (real or virtual) of up to two hours duration. The duration and meeting content depend on data collected from Checking.

- *Checkers' Report*: At the beginning of the specification meeting, checkers *report* their data from Checking. The team leader evaluates this data and makes decisions about how the meeting and the rest of this SQC process should proceed. The meeting may be cancelled or modified in content and duration.
- *Issue Logging*: The checkers report their issues, mainly potential majors, which can be in any of the participating documents. A scribe logs issues in the Editor Advice Log. There should be no discussion concerning the issues discovered, just unconditional logging of the issue (the rule violation and its location in the specification). Checkers may also make process improvement suggestions (*Note: This is part of the DPP process*), and log technical 'questions of intent' to the writer. Issue logging within a specification meeting takes up to 30 minutes.
- *Double Checking*: If it is desired that additional defects are found, then double checking at the experienced specification meeting[6] optimum checking rate will be carried out during the meeting. This identifies

[6] This rate is similar but may vary by about 30% from the optimum rate average found for individual checking activity. In addition, it is a group activity rate and is not directly tunable to single individuals. Of course, single individuals will exploit the given time more or less effectively, depending on their personal ability and motivation.

about 15% additional major defects and adds about 1.5 hours to the meeting. This extra checking is only useful in 'cleanup mode,' not when sampling and measuring to determine exit (normal mode).

Process Meeting

After the specification meeting and a short break, the team optionally may spend up to 30 minutes, analyzing up to 10 logged potential major defects. For each chosen potential defect, one minute is spent describing the issue, one minute is spent brainstorming possible root causes and one minute is spent brainstorming preventative cures. This data will later be recorded in a quality assurance (QA) database as inputs ('grass root insights': suggestions, hints, ideas) to the organization's more-systematic and formal process improvement specialists. (*Note: A Process Meeting is part of the DPP process.*)

Edit

The editor (usually the specification writer) takes over the 'Editor Advice Log', which consists of the issues (that could warrant correction or action) logged at the specification meeting. The editor examines the logged issues, determines how to resolve them and then at least fixes the issues considered to be major defects. The editor may discover additional defects and should make corrections to any majors identified outside the sample checked. Other reasonable action is taken, such as sending out change requests to owners of other documents. An extreme edit is a full rewrite according to all rules.

Edit Audit

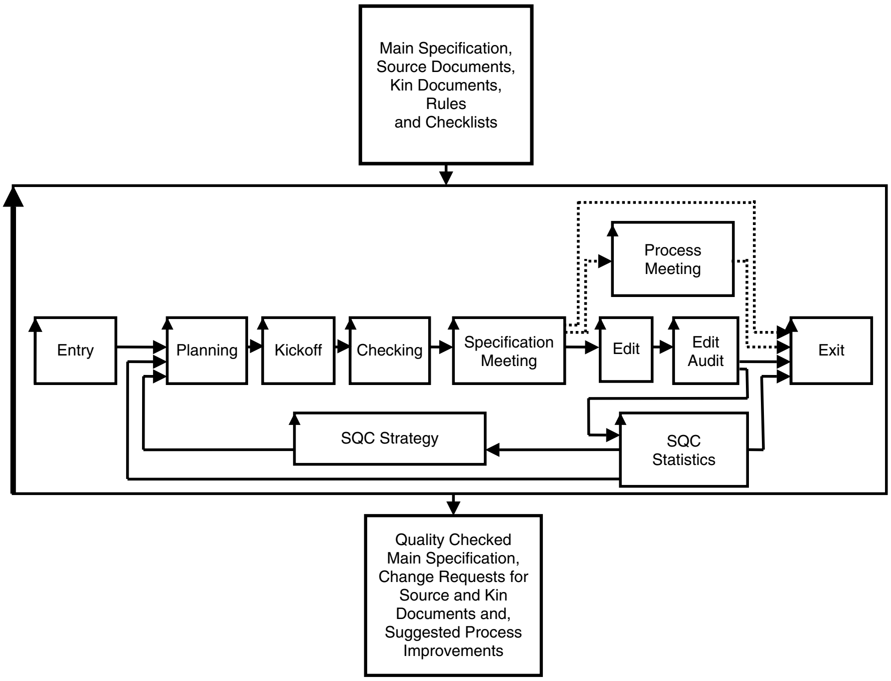
A process carried out by the team leader to verify that a reasonable and complete editing job has been done. Consequently, the editor takes formal responsibility for the editing. This can be done in minutes.

SQC Statistics

The team leader will ensure that all the required statistics from the SQC are captured in the SQC database. This assumes a process control use of SQC data.

Exit

The team leader evaluates the formal SQC exit conditions to see if the specification may be released 'economically' for normal use. The


Main Specification, Source Documents, Kin Documents, Rules and Checklists
Process Meeting
Entry
Planning
Kickoff
Checking
Specification Meeting
Edit
Edit Audit
Exit
SQC Strategy
SQC Statistics
Quality Checked Main Specification, Change Requests for Source and Kin Documents and, Suggested Process Improvements

Note: The 'Process Meeting' sub-process is exclusively a part of Defect Prevention Process (DPP). All the rest is Defect Detection Process (DDP), although there may be a small component of DPP within some of these sub-processes.

Figure 8.6
Diagram of the SQC Process showing the sub-processes.

estimated number of remaining major defects in the specification is especially important. If the main specification is not released, the team leader must work towards acceptable exit-levels of quality, usually in cooperation with the specification writer.

Exit Conditions

X1: The main specification must have fewer remaining major defects/ page than the agreed exit standard (a maximum of 'one remaining' is a reasonable ambition level, initially).

A Simplified SQC Process

SQC as described in the procedure above is the full-scale version. There are situations when a simplified SQC process is more appropriate (for example, to obtain a rapid assessment of the specification quality of a contract or to demonstrate to management some of the power of SQC to get their 'buy-in').

A 'Simplified SQC Process' is presented below.

Note: There are several limitations to this simplified process:

- it is only a small sample so the accuracy is not as good as a full or larger sample
- the team will not have time or experience to get up to speed on the rules and the concept of major defect
- a small team of two people does not have the known effectiveness of three or four people
- you will not have the basis for making corrections to the entire specification
- the checking will not have been carried out against all the possible source documents. (Usually in the simplified SQC process, no source documents are used and memory is relied on. While this means that the checking is not nearly as accurate, it does considerably speed up the process.)

However, if the sample turns up a defects density estimation of 50 to 150 major defects/page (which is quite normal), that is more than sufficient to convince the people participating, and their managers, that they have a serious problem.

The immediate solution to the problem of high defect density is not to remove the defects from the document. The most effective practical solution is to make sure each individual specification writer takes the defect density criteria (and its 'no exit' consequence) seriously. They will then learn to follow the rules and, as a result, will reduce their personal defect injection rate. On average, a personal defect injection rate should fall by about 50% after each experience of using the SQC process. Widespread use of SQC will result in large numbers of engineers learning to follow the rules.

To get to the next level of quality improvement, the next step is to improve the rules themselves.

Simplified SQC Process

Tag: Simplified SQC.
Version: October 7, 2004.
Owner: Tom@Gilb.com.
Status: Draft.

Entry Conditions

- A group of two, or more, suitable people* to carry out Simplified SQC is assembled in a meeting.
- These people have sufficient time to complete a Simplified SQC. Total elapsed time: 30 to 60 minutes.
- There is a trained SQC team leader at the meeting to manage the process.

Procedure

P1: Identify Checkers: Two people, maybe more, should be identified to carry out the checking.

P2: Select Rules: The group identifies about three rules to use for checking the specification. (My favorites are clarity ('clear enough to test'), unambiguous ('to the intended readership') and completeness ('compared to sources'). For requirements, I also use 'no design'.)

P3: Choose Sample(s): The group then selects sample(s) of about one page in length (300 non-commentary words). Choosing a page at random can add credibility – so long as it is representative of the content subject to quality control. The group should decide whether all the checkers should use the same sample or whether different samples are more appropriate.

P4: Instruct Checkers: The SQC team leader briefly instructs the checkers about the rules, the checking rate, and how to document any issues and determine if they are major defects (majors).

P5: Check Sample: The checkers use between 10 and 30 minutes to check their sample against the selected rules. Each checker should 'mark up' their copy of the document as they check (underlining issues and classifying them as 'major' or not). At the end of checking, each checker should count the number of 'possible majors' they have found in their page.

P6: Report Results: The checkers each report to the group their number of 'possible majors.' The SQC team leader leads a discussion to determine how many of the 'possible majors' are actually likely to be majors. Each checker determines their number of majors and reports it.

P7: Analyze Results: The SQC team leader extrapolates from the findings the number of majors in a single page (about 6 times** the most majors found by a single person, or alternatively 3 times the unique majors found by a 2 to 4 person team). This gives the major defect density. If using more than one sample, average the densities found by the group in different pages. The SQC team leader then multiplies this average major defects/page density by the total number of pages to get the total number of major defects in the specification (for dramatic effect!).

P8: Decide Action: If the number of majors/page found is a large one (ten majors or more), then there is little point in the group doing anything, except determining how they are going to get someone to write the specification properly. There is no economic point in looking at the other pages to find 'all the defects', or correcting the majors already found. There are too many majors not found.

P9: Suggest Cause: Choose any major defect and think for a minute why it happened. Then give a short sentence, or better still a few words, to capture your verdict.

Exit Conditions

- Exit if less than 5 majors/page extrapolated total density, or if an action plan to 'rewrite' has been agreed.

Notes:

[*] A suitable person is anyone, who can correctly interpret the rules and the concept of 'major.'
[**] Concerning the factor of multiplying by '6': We have found by experience (Gilb and Graham 1993: Bernard) that the total unique defects found by a team is approximately twice that of the number found by the person who finds the most defects in the team. We also find that inexperienced teams using Simplified SQC seem to have about one third effectiveness in identifying the major defects that are actually there. So 2 × 3 = 6 is the factor we use (or 3 × the number of unique majors found by the team).

Simplified Specification Quality Control Form

SQC Date: **May 29, 200X**. SQC Start Time: ______
SQC Leader: **Tom**.
Author: **Tino**.
Other Checkers: **Artur**.

Specification Reference: **Test Plan**.
Specification Date and/or Version: **V 2** Total Physical Pages: **10**.

Sample Reference within Specification: **Page 3**.
Sample Size (Non commentary words): **approx. 300**.

Rules used for Checking: **Generic Rules, Test Plan Rules.**
Planned Exit Level (Majors/logical page): ______ or less.

Checking Time Planned (Minutes): **30**. Actual: **25**.
Checking Rate Planned (Non commentary pages/hour): **2**.
(*Note this rate should be less than 2 logical pages/hour*)

Actual Checking Rate (Non commentary words/minute): ______
Number of Defects Identified by each Checker:
Majors: **6, 8, 3**. Total Majors Identified in Sample: **17**.
Minors: **10, 15, 30**.

Estimated Unique Majors Found by Team: **16 ± 5**.
(*Note 2 × highest number of Majors found by an individual checker*)

Estimated Average Majors/Logical Page: ~**16** × **3** = **48**.
(*A Logical Page = 300 Non commentary words*)
Majors in Relation to Exit Level: **48/1 (47 too many)**.
Estimated Total Majors in entire Specification: **48** × **10** = **480**.
Recommendation for Specification (*Exit/Rework/Rewrite*): **No exit, redo and resubmit**.

Suggested Causes (of defect level): **Author not familiar with rules**.

Actions suggested to mitigate Causes: **Author studies rules, All authors given training in rules**.

Person responsible for Action: **Project Manager**.
SQC End Time: **18:08.** Total Time taken for SQC: ______

Version: August 15, 2004. Owner: Tom@Gilb.com

8.6 Principles: Specification Quality Control

1. **The Principle of 'Illegality'**
   'Defects' are *objective* violation of accepted written rules.
2. **The Principle of 'Majors are the pay off'**
   Major defects are the only *economically interesting* defects.
3. **The Principle of 'Keen to be seen clean'**
   The main purpose of SQC is to measure that the specification is clean enough: *not* to clean up a specification that *isn't*.
4. **The Principle of 'Cleanup your own mess'**
   Specification cleanup is the writer's responsibility, *before* SQC.
5. **The Principle of 'Prevention is better than cure'**
   There are many effects of SQC, but the most useful are learning to avoid defects caused by bad process, and committed by the writer.
6. **The Principle of '50% effectiveness'**
   History shows that you can only expect to find and fix about *half* the defects that are there.
7. **John Craven's Principle (within Hewlett Packard)**
   The team is there to make the "writer look like a *hero*."
8. **The Principle of 'Magnificent Profitability'**
   The expected return on investment for SQC is at least 'ten to one.'
9. **The Principle of 'Client-Server'**
   The writer is the *client* and the checkers serve as advisors.
10. **The Principle of 'The Pilot in Command'**
    The team leader is responsible for the SQC process.

Good execution of a badly executed specification will tend to execute you!

8.7 Additional Ideas: Specification Quality Control

There are some *central* ideas of SQC, which are worth looking at in more depth:

Economics of using SQC

The cost of finding and fixing defects has to be balanced against the benefit of removing the defects. The cost of fixing a defect escalates the longer it is left unfixed. In general, as we move from requirements/

design-stages to test-stages, the total system-wide costs of *removing* major defects increase by an order-of-magnitude. As we move into design implementation, manufacturing, installation, servicing and distribution, yet *another* additional order-of-magnitude of cost is generally our penalty for dealing with major defects later.

The cost of finding defects also varies. There are several QC nets that specifications pass through as a product is developed. Also, just because there *is* a defect doesn't mean that it *will* cause damage. This is where sampling and understanding your document quality level is essential.

If there is more than approximately one remaining major defect/page, it will tend to pay off to fight the defects immediately, using SQC, rather than downstream. With less than that, it probably pays off to let that major defect (exact location unknown) slip through *this particular* QC net, and hope it is still caught in some other QC net in the systems engineering process.

Unfortunately most real engineering environment 'approved' documents are at least one order-of-magnitude *worse* quality than one major defect/page: 10 to 20 or far more major defects/page is common, according to my frequent measurements. But without SQC, to measure for us, we don't 'know' this.

Effectiveness of SQC

If SQC is consistently carried out according to official guidelines (critically including the 'checking rate' being at the optimum level), then experience in IBM Rochester Labs, MN (Gilb and Graham 1993 Page 23) shows that the defect-finding *effectiveness is relatively stable*. Thirty percent effectiveness is a beginner's level (my experience). For a mature SQC process, effectiveness, for a single-pass attempt, tends to be in the range 60% to 90% (Gilb and Graham 1993: IBM Experience), depending on the type of specification being checked. By systematic SQC process improvement, the effectiveness can slowly be improved to its maximum potential. Cumulative SQC effectiveness, for multiple passes, has been shown to reach a maximum of 95% (IBM UK and Sema UK Case (Gilb and Graham 1993: Leigh, D.)).

Determining Effectiveness and Estimating Remaining Defects

SQC can be used directly to *measure* defects found and, indirectly to *estimate* the defects not found. Providing that 'effectiveness' (% of 100%) at finding defects is known and is relatively stable, it can be used to estimate the number of *unfound* defects. Effectiveness (of your teams on your specifications at finding defects) can be determined in two ways:

1. For a specific specification, we can measure the percentage of the 'available' defects, which a given SQC process was able to find. We can work out this percentage if we know the number of defects found by the SQC and the total number of defects found at later stages by other QC processes, testing and field use. IBM has practiced this for decades in software engineering (Kaplan, Clark and Tang 1994; IBMSJ 1994).
2. Another method, faster and cheaper, and more credible locally, is to repeat SQC on the same sample. If we find 30 defects on first attempt, fix them all, and hypothesize that we are 30% effective at finding them, this means we should have about 70 defects not found in our sample, right? If, after fixing all the 30 we found, the second SQC, done on the same sample, consistently finds about 21 defects (± about 6), it would confirm our prediction of 30% (21 of the 70 remaining from the first SQC). The '% of available defects found' is the effectiveness of the given SQC process. We use this method regularly on our training courses, and it works! It will also work for *any* test or QC process.

Once you have determined your effectiveness, you can estimate the remaining defects in a specification. We use the number of 'estimated probable remaining defects' to decide if a specification can exit (a typical exit condition is 'no more than one probable remaining major defect/ page'). See Figure 8.7 for the calculation of 'estimated probable remaining defects.' We use effectiveness to determine the number of defects unidentified, and then we improve the accuracy by considering the effects of the editing. One sixth of fix attempts during editing fail (M. Fagan 1986:[7] IBM experience), unless an SQC for each fix is done to reduce fix failure (IBMSJ 1994: Kan). In addition, defect injection occurs during editing as a side effect of the fixes; the defect injection rate is sometimes 2% to 5% – but is highly variable and uncertain.

The final consideration is the uncertainty in the estimate. I have found that this remaining defect estimate is reasonably correct, and even in poor circumstances is ±30% uncertain, which is good enough for most purposes.

A specification can have 'too high a density' of major defects (equals serious engineering-cost rule violations) to be acceptable for use (to be allowed 'entry' or 'exit'). *'We will find it in test'* is a dangerous delay. Delaying *action* on your specification's major defects threatens not only cost (thus profit), but time-to-market and competitive quality. It pays off to deal with most major defects *early*.

[7] Fagan. M. E, 'Advances in Software Inspections', *IEEE Transactions on Software Engineering*. Volume SE-12, Number 7, Pages 744–751, July 1986.

Estimating the Remaining Major Defect Density

Assumptions:
A logical page (page) is 300 non-commentary words.

- 30 major defects/page have been found during SQC.
- Your SQC effectiveness is 60% and your SQC is a statistically stable process.
- One sixth of your attempts to fix defects fail (One sixth is average failure to fix).
- New defects are injected during your attempts to fix defects at 5%.
- The uncertainty factor in the estimation of remaining defects is ±30%.

Probable remaining major defects/page = 'Probable unidentified majors' + 'Bad fix majors' + 'Majors injected'

Let E = Effectiveness expressed as a percentage (%) = 60%

Probable unidentified majors = major defects acknowledged-by-editor for each page at Edit × (100 − E) / E = 30 major defects/page found × (100 − 60) / 60 = **20 major defects/page**.

Bad fix majors = One sixth of fixed majors = So, of 30 attempted fixes, **5 major defects in each page** are not fixed.

Majors injected = 5% of majors attempted to be fixed = **1.5 major defects/page**.

Probable remaining major defects/page = 20 + 5 + 1.5 = 26.5 remaining major defects/page.

Taking into account the uncertainty factor of ±30% and rounding down to the nearest whole number gives **26 ± 7 Remaining Major Defects/Page**

(Minimum = 19, Maximum = 33 remaining major defects/page).

Figure 8.7
Estimating Remaining Major Defect Density: the main specification exit condition.

SQC and Rules

SQC is completely dependent on the rules that are applied. Just because you exit from an SQC process does not mean that all quality checking has been completed. It simply means that checking has been completed against the rules actually used, and has demonstrated an acceptable defect level.

By using different rules, different types of quality checking can be achieved. It is not simply a case of using the relevant rule set to match the type of specification. You need to consider what type of defects you are checking for and their potential cost if not detected.

Extending from SQC into Specification Review

There is no reason why the SQC method shouldn't be used to prepare for management reviews. You might have checked the content of a specification for consistency, completeness and clarity (Specification Rules).[8] But maybe you have not yet checked for the relevance to

[8] Note: This chapter mainly discusses and illustrates SQC from the viewpoint of checking for specification clarity, completeness and consistency. This ties in with the rules found in the other chapters, which are *Specification Rules.*

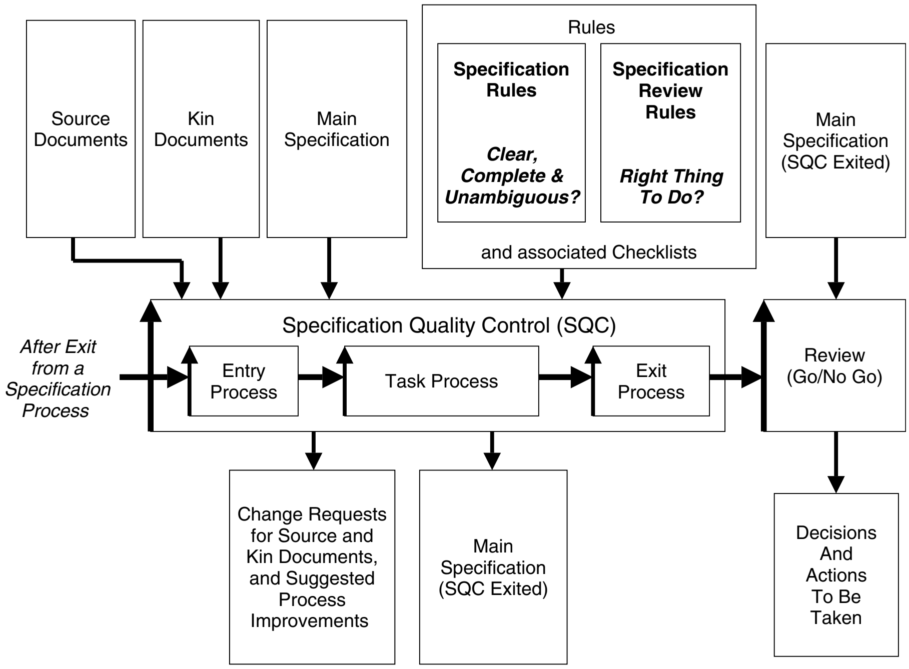

Source Documents
Kin Documents
Main Specification
Rules
Specification Rules
Clear, Complete & Unambiguous?
Specification Review Rules
Right Thing To Do?
and associated Checklists
Main Specification (SQC Exited)
Specification Quality Control (SQC)
After Exit from a Specification Process
Entry Process
Task Process
Exit Process
Review (Go/No Go)
Change Requests for Source and Kin Documents, and Suggested Process Improvements
Main Specification (SQC Exited)
Decisions And Actions To Be Taken

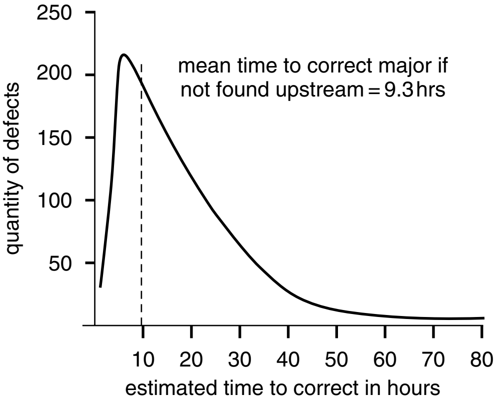
Figure 8.8
Overview of the SQC process showing how Specification Review Rules fit alongside Specification Rules.

current business or technical demands. For example, maybe a certain level of ROI is demanded? Maybe a specific safety margin must be shown to be present? By creating a different set of rules, called Specification Review Rules, the SQC process can also be used to carry out pre-review quality control, in advance of a review meeting. This will probably result in better quality control than would be carried out in a hurried executive review meeting.

8.8 Further Example/Case Study: A Stitch in Time Saves Nine

Trevor Reeve made use of SQC (at that time he called it ‘Statistical Quality Control applied to Software and Documentation’ or ‘Documentation Quality Improvement’) in all industrial aspects of a 1,500 person defense electronics manufacturer (later a part of Racal). He documented four years of experience after this author ran a course on-site (Gilb and Graham 1993, Pages 305–316).

Reeve carried out an analysis of the first 1,000 major defects logged by the SQC process to investigate the cost savings of using SQC. Test



(Source: Thorn EMI June 1989–January 1990

Figure 8.9
MEL/Thorn EMI (later RACAL) UK, Factory and Lab-wide SQC gave order-of-magnitude savings. About 1,000 major defects found by using SQC with multi-disciplinary teams were analyzed. The alternative cost to fix majors, if caught downstream, was nine times greater than if caught upstream by SQC. This is a frequency chart for the 1,000 defects.

and field staff were asked when these defects would have been found in their test or field reports. They were also asked to indicate what the cost of finding and fixing them would be. The frequency curve in Figure 8.9 was drawn based on their answers. The mean time to correct these defects downstream would have been 9.3 hours. The mean time to find and fix them using SQC was about one hour. The defects would *otherwise* have been found by test and by customers. The result of this was that it was acknowledged by top management that removing a major defect using SQC gave a net saving of about 8 hours, or a 9.3 to 1 ratio of engineering hours ‘return’ on investment in SQC.

Compare this to the Raytheon return of 7.7 to 1 (*see Section 1.8*). Six hundred inspections had been done at Thorn EMI by 1992, and over a thousand by early 1993. Savings were conservatively estimated at £500,000 each year, after one-time startup costs of £50,000. External consultants are said to have estimated real savings at double this figure. “Quality increased and development time has been reduced significantly.”

Use of SQC on many different types of document

SQC experience at Raytheon was limited to software, but at MEL/ Thorn EMI, the documents

> “ranged from system, hardware and software design documents to software code and change notes . . . test specifications, proposals, program management documents (for example,

> configuration and program plans), contracts and purchase specifications, printed circuit board design and test specifications, and procedures and standards.
>
> Further application (1993) all contractual documents, drawings and internal specifications (for example, information technology and financial requirements) . . . with all contracts using it to a lesser or greater degree by end 1992. . . .
>
> The product appraisal process was revised to incorporate the technique into the normal activities performed by the organization on all types of document from contract to lowest level design and test specification including drawings.'' (Gilb and Graham 1993: P310)

Note: The use of SQC on the upstream documents will produce the greatest benefits because defects will be caught earlier, and do less damage.

The Organization Must be Supportive and SQC Needs a Champion

Since 1993, Trevor has confirmed many times within different organizations and various parts of the world, that two of the main factors for SQC to succeed are as follows:

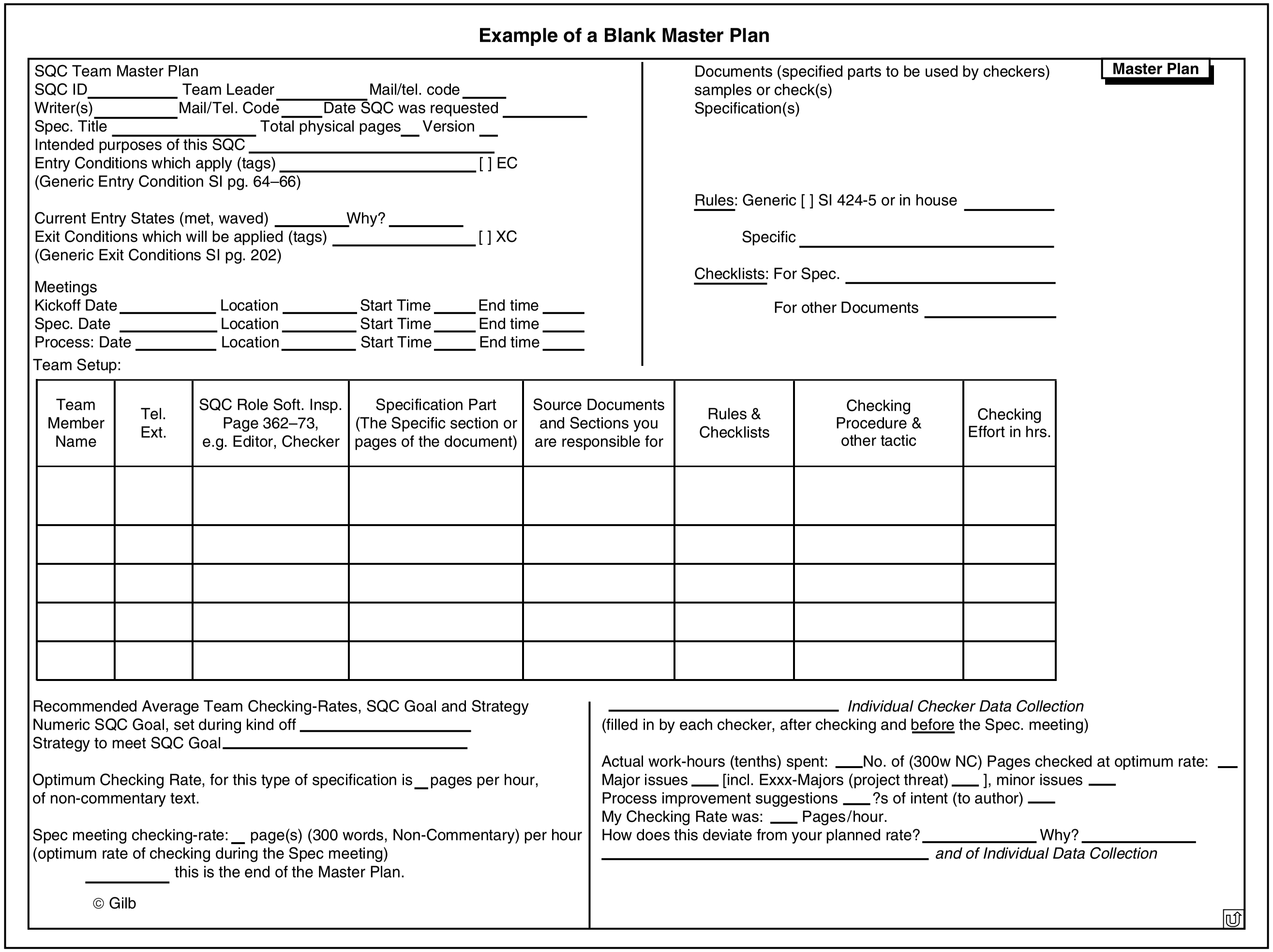
1. An organization really needs to be willing to change, and
2. The continuous presence of a totally committed champion of the method is necessary, for many years after the initial introduction of the method, to help the necessary culture change to take place. (This was also the experience in the same period of another client, Hewlett-Packard (Grady and Van Slack 1994).)

8.9 Diagrams/Icons: Specification Quality Control

This section shows the SQC forms as follows:

- Figure 8.10: Master Plan
- Figure 8.11: Editor Advice Log
- Figure 8.12: Data Summary
- Figure 8.13: Process Meeting Log
- Figure 8.14: Simplified SQC Form

These are the blank versions of the filled-in forms given earlier in this chapter.



Example of a Blank Master Plan

Master Plan

SQC Team Master Plan
SQC ID________ Team Leader________ Mail/tel. code____
Writer(s)________ Mail/Tel. Code____ Date SQC was requested ________
Spec. Title ____________ Total physical pages__ Version __
Intended purposes of this SQC ______________________
Entry Conditions which apply (tags) __________________ [ ] EC
(Generic Entry Condition SI pg. 64–66)

Current Entry States (met, waved) ______ Why? ______
Exit Conditions which will be applied (tags) ____________ [ ] XC
(Generic Exit Conditions SI pg. 202)

Meetings
Kickoff Date ________ Location ______ Start Time ____ End time ____
Spec. Date ________ Location ______ Start Time ____ End time ____
Process: Date ________ Location ______ Start Time ____ End time ____

Documents (specified parts to be used by checkers)
samples or check(s)
Specification(s)

Rules: Generic [ ] SI 424-5 or in house ________

Specific ______________________

Checklists: For Spec. ____________________

For other Documents ____________

Team Setup:

Team Member Name	Tel. Ext.	SQC Role Soft. Insp. Page 362–73, e.g. Editor, Checker	Specification Part (The Specific section or pages of the document)	Source Documents and Sections you are responsible for	Rules & Checklists	Checking Procedure & other tactic	Checking Effort in hrs.

Recommended Average Team Checking-Rates, SQC Goal and Strategy
Numeric SQC Goal, set during kind off ________________
Strategy to meet SQC Goal______________________

Optimum Checking Rate, for this type of specification is__pages per hour, of non-commentary text.

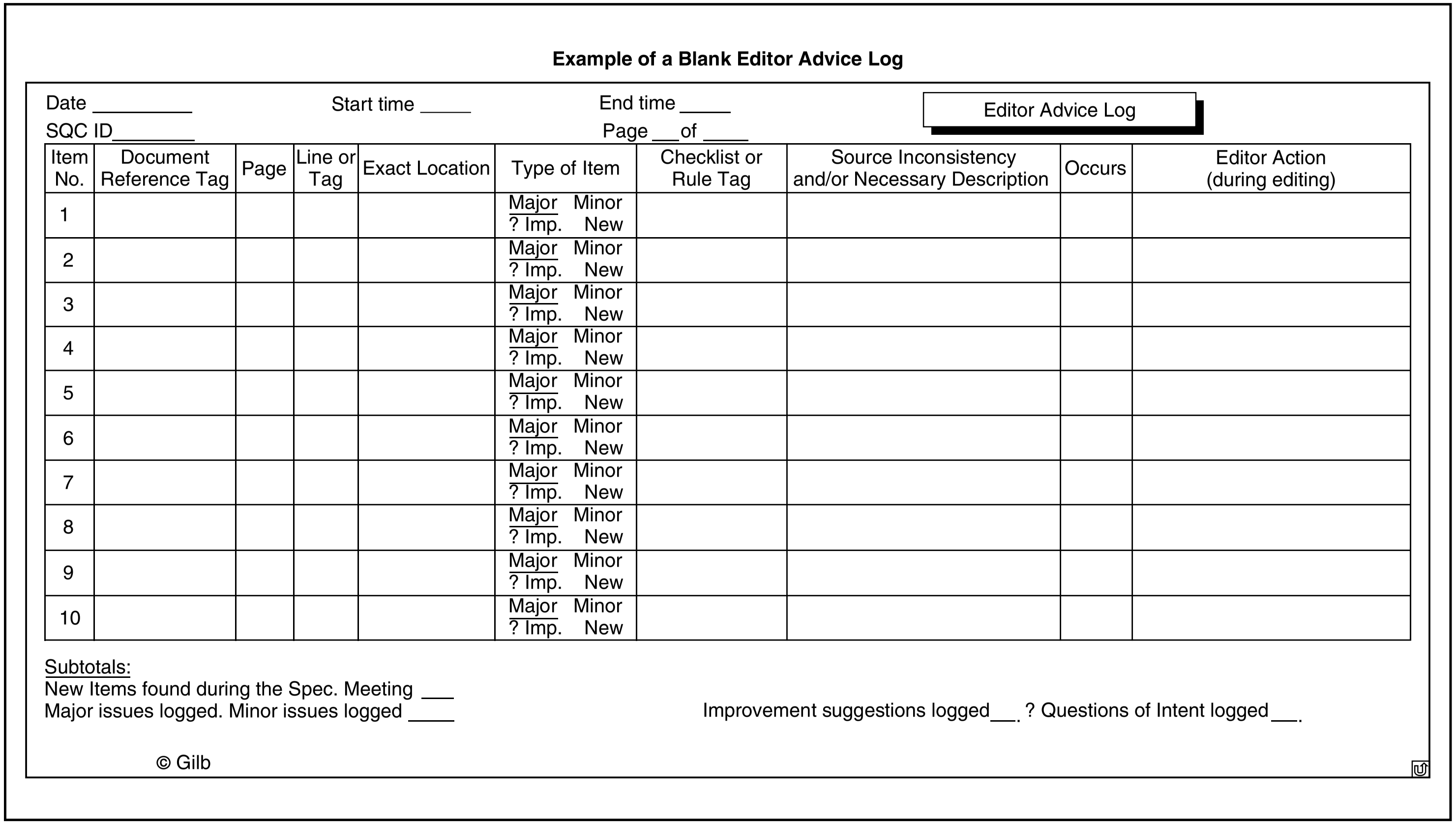
Spec meeting checking-rate:__ page(s) (300 words, Non-Commentary) per hour
(optimum rate of checking during the Spec meeting)
________ this is the end of the Master Plan.

© Gilb

____________________ *Individual Checker Data Collection*
(filled in by each checker, after checking and before the Spec. meeting)

Actual work-hours (tenths) spent: ___No. of (300w NC) Pages checked at optimum rate: ___
Major issues ___ [incl. Exxx-Majors (project threat) ___], minor issues ___
Process improvement suggestions ___ ?s of intent (to author) ___
My Checking Rate was: ___ Pages/hour.
How does this deviate from your planned rate?__________ Why?__________
______________________________ *and of Individual Data Collection*

Figure 8.10 Blank Master Plan.



Example of a Blank Editor Advice Log

Date __________ Start time _____ End time _____

SQC ID________ Page __of ____

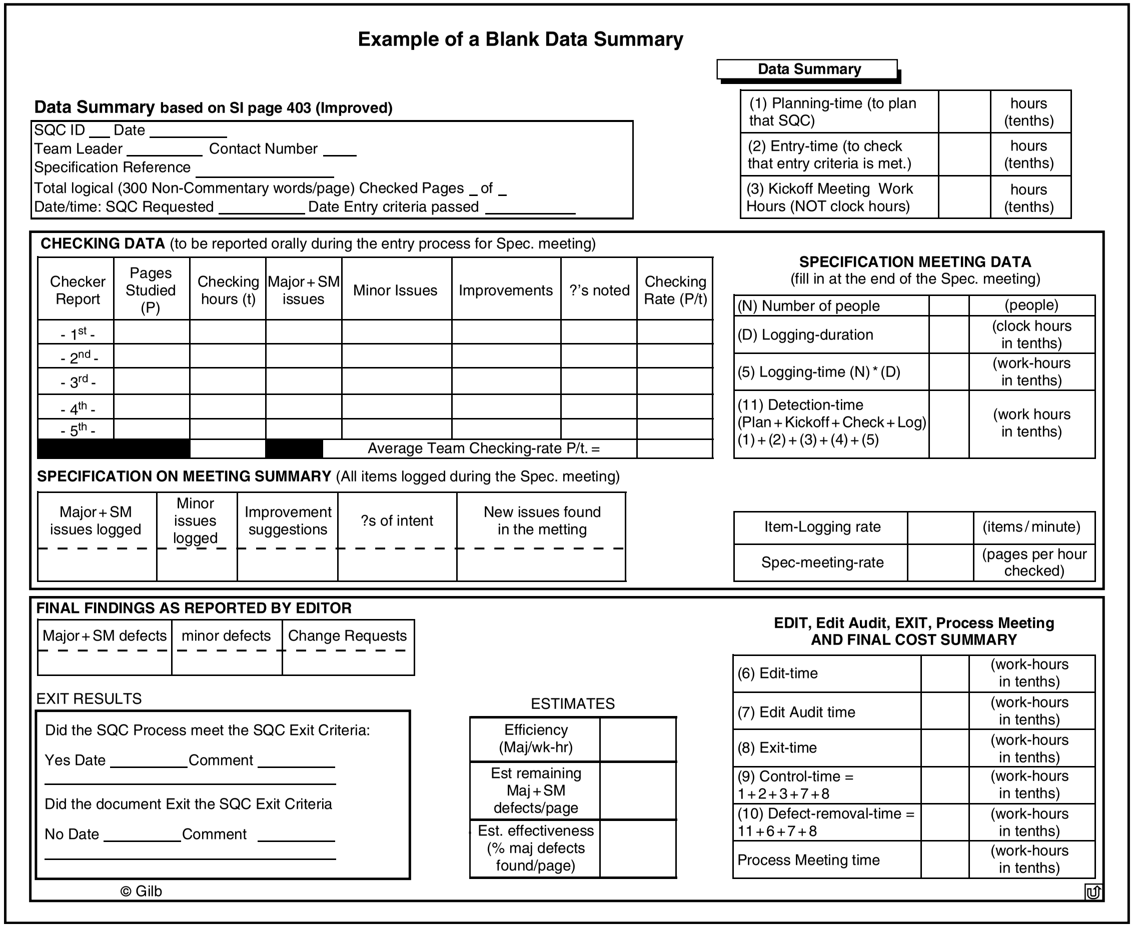
Editor Advice Log

Item No.	Document Reference Tag	Page	Line or Tag	Exact Location	Type of Item	Checklist or Rule Tag	Source Inconsistency and/or Necessary Description	Occurs	Editor Action (during editing)
1					Major Minor ? Imp. New				
2					Major Minor ? Imp. New				
3					Major Minor ? Imp. New				
4					Major Minor ? Imp. New				
5					Major Minor ? Imp. New				
6					Major Minor ? Imp. New				
7					Major Minor ? Imp. New				
8					Major Minor ? Imp. New				
9					Major Minor ? Imp. New				
10					Major Minor ? Imp. New				

Subtotals:
New Items found during the Spec. Meeting ___
Major issues logged. Minor issues logged ____ Improvement suggestions logged__. ? Questions of Intent logged__.

© Gilb

Figure 8.11 Blank Editor Advice Log.



Example of a Blank Data Summary

Data Summary **based on SI page 403 (Improved)**

SQC ID ___ Date _________
Team Leader _________ Contact Number ____
Specification Reference ________________
Total logical (300 Non-Commentary words/page) Checked Pages _ of _
Date/time: SQC Requested __________ Date Entry criteria passed ___________

Data Summary

(1) Planning-time (to plan that SQC)		hours (tenths)
(2) Entry-time (to check that entry criteria is met.)		hours (tenths)
(3) Kickoff Meeting Work Hours (NOT clock hours)		hours (tenths)

CHECKING DATA (to be reported orally during the entry process for Spec. meeting)

Checker Report	Pages Studied (P)	Checking hours (t)	Major + SM issues	Minor Issues	Improvements	?'s noted	Checking Rate (P/t)
- 1st -							
- 2nd -							
- 3rd -							
- 4th -							
- 5th -							
■	■		■	Average Team Checking-rate P/t. =			

SPECIFICATION MEETING DATA
(fill in at the end of the Spec. meeting)

(N) Number of people		(people)
(D) Logging-duration		(clock hours in tenths)
(5) Logging-time (N) * (D)		(work-hours in tenths)
(11) Detection-time (Plan + Kickoff + Check + Log) (1) + (2) + (3) + (4) + (5)		(work hours in tenths)

SPECIFICATION ON MEETING SUMMARY (All items logged during the Spec. meeting)

Major + SM issues logged	Minor issues logged	Improvement suggestions	?s of intent	New issues found in the metting

Item-Logging rate		(items / minute)
Spec-meeting-rate		(pages per hour checked)

FINAL FINDINGS AS REPORTED BY EDITOR

Major + SM defects	minor defects	Change Requests

EXIT RESULTS

Did the SQC Process meet the SQC Exit Criteria:

Yes Date _________Comment _________
____

Did the document Exit the SQC Exit Criteria

No Date _________Comment _________
____

ESTIMATES

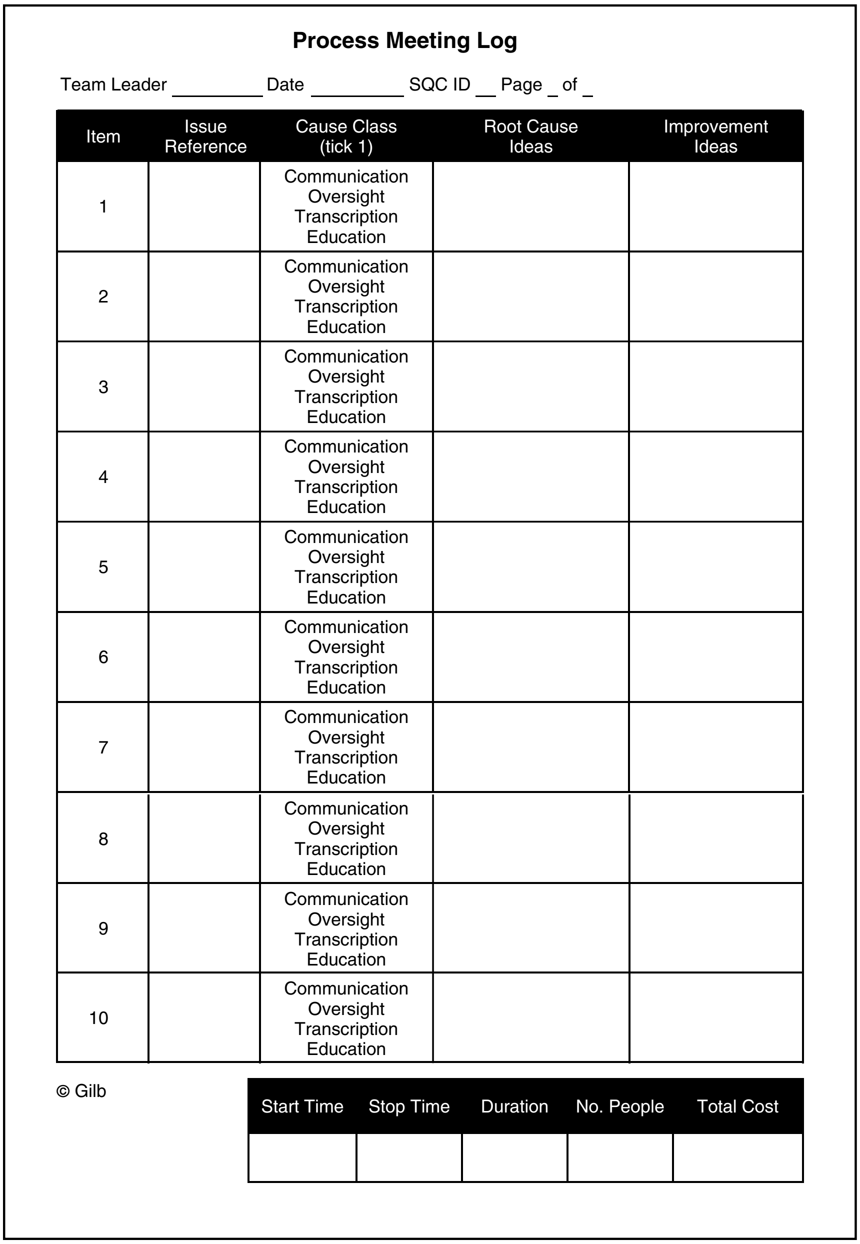
Efficiency (Maj/wk-hr)	
Est remaining Maj + SM defects/page	
Est. effectiveness (% maj defects found/page)	

EDIT, Edit Audit, EXIT, Process Meeting AND FINAL COST SUMMARY

(6) Edit-time		(work-hours in tenths)
(7) Edit Audit time		(work-hours in tenths)
(8) Exit-time		(work-hours in tenths)
(9) Control-time = 1 + 2 + 3 + 7 + 8		(work-hours in tenths)
(10) Defect-removal-time = 11 + 6 + 7 + 8		(work-hours in tenths)
Process Meeting time		(work-hours in tenths)

© Gilb

Figure 8.12 Blank Data Summary.



Process Meeting Log

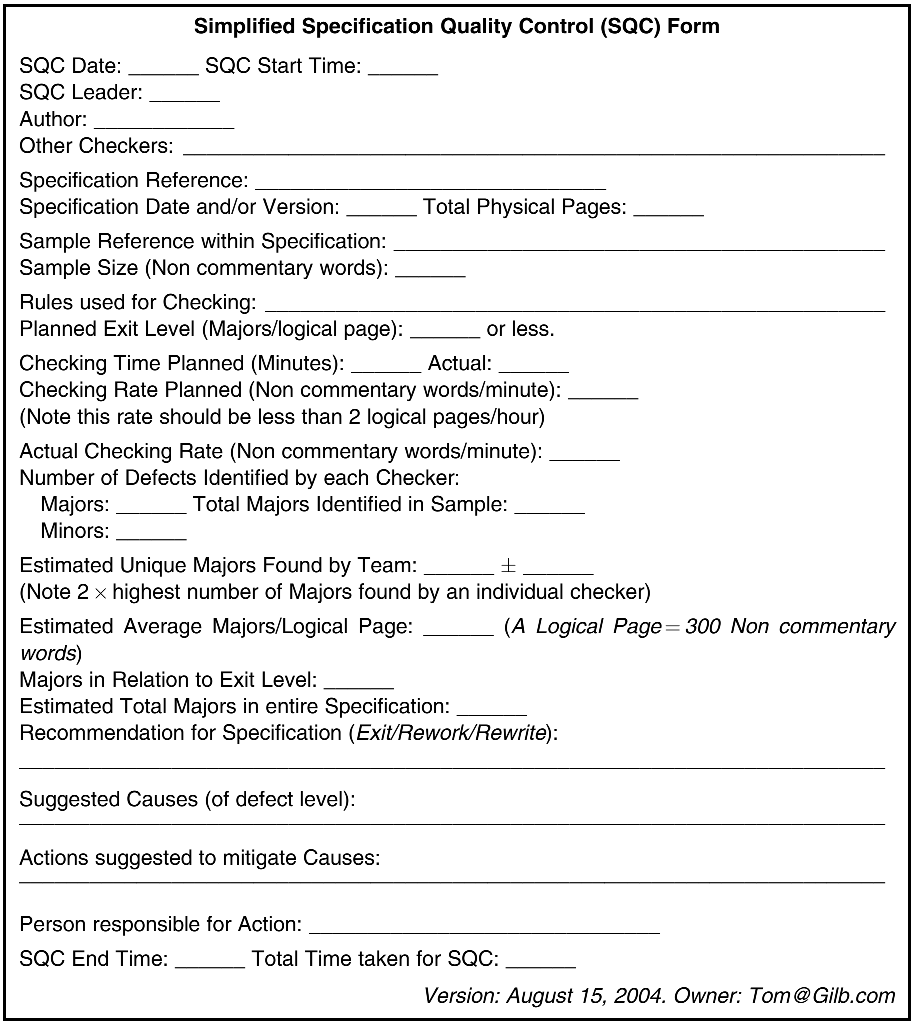
Team Leader _________ Date _________ SQC ID __ Page _ of _

Item	Issue Reference	Cause Class (tick 1)	Root Cause Ideas	Improvement Ideas
1		Communication Oversight Transcription Education		
2		Communication Oversight Transcription Education		
3		Communication Oversight Transcription Education		
4		Communication Oversight Transcription Education		
5		Communication Oversight Transcription Education		
6		Communication Oversight Transcription Education		
7		Communication Oversight Transcription Education		
8		Communication Oversight Transcription Education		
9		Communication Oversight Transcription Education		
10		Communication Oversight Transcription Education		

© Gilb

Start Time	Stop Time	Duration	No. People	Total Cost

Figure 8.13
Blank Process Meeting Log.



Simplified Specification Quality Control (SQC) Form

SQC Date: ______ SQC Start Time: ______
SQC Leader: ______
Author: ____________
Other Checkers: ____

Specification Reference: ______________________________
Specification Date and/or Version: ______ Total Physical Pages: ______

Sample Reference within Specification: ____
Sample Size (Non commentary words): ______

Rules used for Checking: ____
Planned Exit Level (Majors/logical page): ______ or less.

Checking Time Planned (Minutes): ______ Actual: ______
Checking Rate Planned (Non commentary words/minute): ______
(Note this rate should be less than 2 logical pages/hour)

Actual Checking Rate (Non commentary words/minute): ______
Number of Defects Identified by each Checker:
Majors: ______ Total Majors Identified in Sample: ______
Minors: ______

Estimated Unique Majors Found by Team: ______ ± ______
(Note 2 × highest number of Majors found by an individual checker)

Estimated Average Majors/Logical Page: ______ (*A Logical Page = 300 Non commentary words*)
Majors in Relation to Exit Level: ______
Estimated Total Majors in entire Specification: ______
Recommendation for Specification (*Exit/Rework/Rewrite*):
____

Suggested Causes (of defect level):
____

Actions suggested to mitigate Causes:
____

Person responsible for Action: ______________________________

SQC End Time: ______ Total Time taken for SQC: ______

Version: August 15, 2004. Owner: Tom@Gilb.com

Figure 8.14
Simplified Specification Quality Control (SQC) form.

8.10 Summary: Specification Quality Control

The basic ideas of SQC are simple:

- "A stitch in time saves nine": fix defects at *early* design stages (DDP), before they cause damage and/or require a costly 'defect removal' process, during test or operation,
- "An ounce of prevention is worth a pound of cure": learn from defects, which have common underlying causes, and continuously improve your work processes (DPP).

Finding Defects

The Defect Detection Process (DDP) is more powerful than similar processes, such as 'checking'[9] (of engineering drawings, as proven at Douglas Aircraft 1988, Boeing 1989 and Thorn EMI 1990 on), 'reviews ', 'walkthroughs ', meetings and management approval. This is mostly due to a series of tactics, the most critical of which are probably the use of a proven *optimum defect-searching time* (optimum checking rate) and, the total focus on *finding and fixing 'major' defects* (which saves time downstream).

Understanding Document Quality

One of the most important opportunities using SQC is to be able to measure the degree to which systems engineering and management documents of all types really do correspond to the required standards of quality. The concepts of 'entry' to, and 'exit' from, all systems engineering and management processes are enabled by our ability to measure 'probable remaining major defects/page' and to decide by estimation if a specification is economic enough to release downstream ('exit'), or economic enough to start work on (allow 'entry'). If necessary, sampling of large documents is an economic way to measure quality levels before making decisions of consequence concerning those documents.

The fact that the SQC process is universally applicable to any readable specification (in any intellectual, administrative, project management, planning, systems engineering, software or user specification task), means that any group of people can use it wherever they want control over quality-in-relation-to-standards. However, SQC has some limitations in understanding 'how well' specifications will work in practice. Even if specifications exit according to any rules you use to analyze them, there can still be catastrophic defects in them in practice. So, we need to use additional methods to see 'how good' a specification is and, if necessary, adjust the specification. *That* is the mission of other tools in this book (like Impact Estimation and Evolutionary Project Management).

The SQC ability to measure quality, in relation to standards, is also important when the standards are a major part of continuous process

[9] Do not confuse with the SQC 'Checking' sub-process! The aircraft factory traditionally used the term 'checking' for a process done by a group of people who specialized in this, called 'checkers.' The process checked engineering drawings against the official engineering drawing specification rules, which were in a large handbook – so large that copies were not given to inform individual engineers what the rules were! In 1988 we proved, with hard data on a large scale, for the engineering directors, that the SQC process was far more effective at finding interesting engineering defects than the traditional 'checking' process. We ended up within the first year with sixty times better quality in terms of rejected and reworked drawings (0.5% versus earlier about 30% reworked).

Possible Purposes For Using SQC

-Reducing Time-to-Delivery
-Measuring the Intelligibility of a Document
-Measuring the effectiveness of engineering specifications
-Measuring the ability of the Process producing the Document to follow best practice rules.
-Enabling Estimation of the Number of Remaining Major Defects
-Identifying Major Defects
-Removing Major Defects
-Preventing consequential 'Downstream' Defects being generated by removing existing Defects
-Improving the Engineering Specification Process (better standards, like rules)
-Improving the SQC Process (better rates, better entry exit conditions, better procedures)
-On-the-Job Training for the Checkers
-Training the Team Leader
-Certifying the Team Leader
-Peer Motivation (getting people to learn, and follow the rules)
-Motivating the Managers (to deal with problems early)
-Helping the Writer (learn to write clearly and have effective ideas)
-Reinforcing Conformance to Standards
-Capturing and Re-using Expert Knowledge (by use of Rules and Checklists)
-Reducing Costs
-Team Building
-Fun – a Social Occasion

Figure 8.15
Possible purposes for using SQC. Any one or several can apply at anytime.

improvement. We can use SQC to measure process improvement efforts! The measurement of defects is a measure of whether people have actually learned, practiced, and understood the continuous improvements intended to increase productivity.

Continuous Process Improvement

The Defect Prevention Process (DPP) exploits grass-root everyday experience with Defect Detection Process (DDP), as well as making use of your data about defects from 'test' and 'field' situations. DPP is the 'engineering and management' version of what Deming (1993) and Juran (1974) taught manufacturing industry, starting in Japan. Experience (Dion 1993; Haley et al. 1995; Kaplan, Clark and Tang 1994) shows that 40% annual productivity improvements are possible, when this is done properly (which is rare).