Software Inspection for the Internet Age: how to increase effect and radically reduce the cost.

Monday June 18th 2001, 8 to 4pm

Software Special Interest Group (SSIG)
(ASQ Section 0509)

Location: Woodfin Suites
1380 Piccard Drive,
Rockville, MD
(301) 590-9880

Lecturer Tom Gilb
TomGilb@Result-Planning.com
Www. Result-Planning.com

The old concepts of inspection
### Difference to ‘Conventional’ Inspections

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>⇒ No sampling</td>
<td>⇒ Sampling to measure doc.</td>
</tr>
<tr>
<td>⇒ Inflexible bureaucracy</td>
<td>⇒ ‘Intelligent Inspections’</td>
</tr>
<tr>
<td>⇒ Focus on ‘Cleanup’</td>
<td>⇒ Focus on Time &amp; Control</td>
</tr>
<tr>
<td>⇒ Focus on ‘software code’</td>
<td>⇒ Systems, upstream focus</td>
</tr>
<tr>
<td>⇒ Poorly documented process</td>
<td>⇒ Richly documented (Book)</td>
</tr>
<tr>
<td>⇒ “My Way”&lt;--MEF/IBM</td>
<td>⇒ ‘Our Way’ &amp; ‘Your Way’</td>
</tr>
<tr>
<td>⇒ Do the defined process!</td>
<td>⇒ Do what pays off, only</td>
</tr>
<tr>
<td>⇒ ‘Interpret’ the document</td>
<td>⇒ Check against Rules, Sources</td>
</tr>
</tbody>
</table>

---

### The new concepts of inspection
10 Top Inspection Principles

- **Pr1.** *Prevention* is more effective than Cure (a stitch in time …)
- **Pr2.** *Avoidance* is more efficient than removal (an ounce of prevention ..)
- **Pr3.** *Feedback* teaches effectively
- **Pr4.** *Measurement* gives facts to control the process
- **Pr5.** Priority to the *Profitable*
- **Pr6.** *Forget perfection*, you can’t afford
- **Pr7.** Teach *fishing*, rather than ‘give fish’
- **Pr8.** *Framework for Freedom* beats bureaucracy
- **Pr9.** *Reality* rules
- **Pr10.** *Facts* beat intuition

SOME DETAILS OF NEW CONCEPTS
Optimum Checking Rate: new in ‘Preparation’

- **Optimum Checking Rate**
  - The most effective individual speed for ‘checking a document against all related documents’.

  - **Notes**
    - Not ‘reading’ speed
    - Correlation speed
    - Range = 1 Logical Page
      - ± 0.9/hour
    - Failure to use it, gives ‘bad estimate’ for ‘Remaining defects’

  - Fault Density versus Checking Rate:
    - Raytheon 95 <- SEI website

  - Action Items/1,000 Source lines
    - Thousands of Statements Checked per hour by a person

  - This area is the ‘illusion of quality’

---

**Issue**

- **Issue**
  - A potential ‘violation of a ‘Rule’ (=Defect)
  - As viewed by a Checker

  - **Note**
    - The ‘Editor’ decides if it really violates a Rule
    - Saves conflict at meetings
    - Saves meeting time
    - Avoids parallel investigation by checkers
Primary Checking Role

- **Primary Role:**
  - The set of checking roles assigned in the Master Plan, and agreed to, by Kickoff time, by the Checker.
  - Notes:
    - Not ‘exclusive’ territory
    - Can be intentionally given to other Checkers too
    - Desired that we report issues which we find outside our Primary Role.

Inspections and Reviews

- **Inspections**
  - Judgement based on conformance to standards
  - Well written, clear, complete, trustworthy
  - Can be used by any of ‘intended readership’
  - Should be done to guarantee decision-makers a good basis for a decision.
  - “I=Consistency of Content”
  - with Sources

- **(Go No-Go) Reviews**
  - Judgement based on goodness in real world
  - Content (‘Value’), not format (‘clarity’);
  - Value, not clarity
  - Approval by authorized ‘managers’/Senior Engineers
  - Should not be Entered if document not Exited from Inspection.
  - “R= Value of Content”
See any ‘defects’ with this?

• “The objective is to get higher adaptability using advanced architecture”

Quality Control to a standard is not new. The term ‘Inspection’ is on 50% pages in this book
Here is a standard “Rules” for quality objectives

1. They should be unambiguously clear to the intended reader.
2. They shall specify a SCALE of measure to define the Quality/Cost concept.
3. They shall break down complex concepts into a set of measurable elementary concepts.
4. To define ‘relative’ terms like ‘higher’ they shall specify at least two points of reference on the defined SCALE.
5. They shall specify exactly when a quality level is to be available.
6. They shall not mix design ideas in the specification of objectives/requirements.
7. The process input or “source” (like contract, standard, marketing plan) of the requirement shall be given.
8. Fuzzy unclear concepts shall be marked with <angle brackets> for improvement.

Defects in a Statement

The objective is to get higher adaptability using advanced architecture

- ambiguous, unclear (1), (8)
- no <fuzz>
- no 2 points of reference to define ‘higher’ (4)
- no SCALE (2)
- complex concept not broken down (3)
- source not given (7)
- a design idea is mixed into the objective (6)

QOBJ.1. They should be unambiguously clear to the intended reader.
QOBJ.2. They shall specify a SCALE of measure to define the concept.
QOBJ.3. They shall break down complex concepts into a set of measurable concepts.
QOBJ.4. To define ‘relative’ terms like ‘higher’ they shall specify at least two points of reference on the defined SCALE.
QOBJ.5. They shall specify exactly when a quality level is to be available.
QOBJ.6. They shall not mix design ideas in the specification of objectives.
QOBJ.7. The process input (like contract, standard, marketing plan) of the requirement shall be given.
QOBJ.8. Fuzzy unclear concepts shall be marked with <angle brackets> for improvement.
‘Editing’ to follow the rules
(this might not be a ‘good’ plan but it contains no ‘defects’)
We are using ‘Planguage’ here.

- Adaptability:
  - Maintenance:
    - SCALE: Clock time to fix a bug and <validate> fix.
    - PAST [Product X, last year] 5 hours <- Internal stats.
  - Portability:<- Marketing Plan Dec 15th. M.P.
    - SCALE: Conversion cost for [defined ports].
    - PAST [Prod. X, Any UNIX, 1999] 100 hours/1000 Lines

New Inspection Objectives

- Central Objectives
  1. Engineering Process Control *(the organization)*
  2. Measuring Specification Quality vs ‘Standards’ *( = ‘Rules’)-- *(level= the project, the spec)*
  3. Reduce Project Time & Cost

- Secondary Objectives
  4. Identify and *(possibly!)* Remove Major Defects
  5. Reduce Service/Maintenance *(Reuse?<-Juha)* Costs

- NOT Objectives
  - Approve document ‘content’ versus ‘Real World’
  - Remove minor defects
  - ‘Improve’ Quality of your end product
**Overall Process Map**

- **Entry**
- **Planning**
- **Kickoff**
- **Checking**
- **Logging**
- **Process Brainstorming**
- **Edit**
- **Edit Audit**
- **Exit**

**Product Sources**
- Rules
- Checklists

**Data Summary**

**New Inspection Paradigms**

- **P1.** Engineering process control
- **P2.** Cleanup is ineffective
- **P3.** Teamwork beats ego
- **P4.** Data beats guessing
- **P5.** Real Time Control
- **P6.** Author Responsibility
- **P7.** Checker Consultants
- **P8.** Author is Client
- **P9.** Optimize Checking speed
- **P10.** Quantified Gatekeepers
- **P11.** Rules Rule Objectively
- **P12.** Structure satisfies objectives
- **P13.** What works is right

See detailed comment on each paradigm in slides at the rear of the Team Leader Slides at www.result-planning.com collection
Generic Rules examples:
Defect=Rule violation paradigm

- G1-CONSISTENT. Statements must be consistent with other statements in same or related documents.
- G2.-ONE. All specifications shall be unambiguous to the intended readership, unless clearly marked (for example using the <angle parenthesis>).
- G3-NOTE. All form of comment, note, suggestion, idea which does not form an official part of the plan shall be clearly distinguished as such (for example by quotes, italics, footnotes, prefacing words.)
- G4-EXTRA. All specifications shall be as brief as possible, to support their purpose.
- G5-CLEAR. All specifications shall be crystal clear to all intended readers as to intent. The burden is on the planner not the reader. Clear enough to test.
- G7-UNIQUE. Specifications shall be stated once only in plans and thereafter referred to by their unique tag. Use comments (") to paraphrase.
- G8-SOURCE. Specifications shall contain information about their exact and detailed sources. Normally use the “<->” source arrow, but also “evidence” and comments. This applies to modifications as well.
- G9-EL. Specifications shall be broken up into their most elementary form (called “statements”), to permit separate analysis, costing, and implementation.
- G10-TAG. All elementary statements shall have their own identity tag for direct reference from other parts of any larger plan set. Parameter name and qualifiers can be used as sub-tags. e.g. USABILITY.PAST[1994].

Sampling
How big a sample?

- **Purpose**
  - To get some accurate idea of major defect density at low cost

- **Method**
  - Sample big enough to be credible (1-3 pages)
  - Sample where it is credible (critical text)
  - High precision not required (not 'science')
    - Enough to know under or over exit borders
    - Order of magnitude is a good beginning

Sample a definition

- **Sample**
  - *Part* of a document, which we want to use to measure the defectiveness of the whole document with.

  - **Notes**
    - Sampling can follow ‘sampling principles’
      - Size
      - Representative
      - Convincing
    - Cheaper but adequate.
Purpose of Sampling

• Sampling is useful
  ⇒ to permit optimum checking rates when there is not enough resource to check entire document
  ⇒ when initial chunks tell you that there is no point in continuing checking other chunks
  ⇒ to demonstrate the power of inspection for skeptical peers and managers

• Sampling is not valid when
  ⇒ The document is very small (less than 10 pages)- because you might just as well do the whole document
  ⇒ it is not done on representative sections
  ⇒ it is necessary to identify (and possibly also to fix) all possible defects, even if they are fewer than 1 per sample

Guidelines for Sampling

• Formally same as any guidelines for statistical sampling (see for example “Juran’s Quality Handbook”), Deming: Some Theory of Sampling” (Wiley 1950)

• Informal guidelines may be sufficient for many purposes
  ⇒ Size of sample as percentage of whole
    → What does it mean if you find no Major defects in a one page sample?
    → What does it mean if you find 30 Majors on a page?
  ⇒ Sample representativeness
    → does anybody agree that the sample is representative before you begin?
    → what could you do if people dispute the sample’s representativeness?
  ⇒ What confidence would your target audience for any conclusion have in it?
    → will they ‘buy’ the conclusion?
    → are they participants in drawing the conclusion themselves?
  ⇒ What is your exact purpose at this moment
    → to exit-or-not conclusions
    → provoke further inspections this document
    → impress managers with the method
A Sampling Case Study: Air Traffic

- **1986 Northern Europe**
  - Air traffic control trainer system for export
  - 80,000 pages contracted documentation before code
  - 40,000 pages already written
  - Project seriously late already (customer informed)
  - About 7 management signatures approving the 40,000 pages (pseudocode for coders)
  - Inspection of a sample of three pages
    - chosen by random numbers
    - declared to be representative
    - 19 Major defects found in half day inspection by the 7 managers
    - divisional director personally checks the defect log and confirms “I would be fired by the Chairman if I let these get to a customer” (he was later fired)
    - If we found 30%, how many were there in total?
  - What would have happened if we had inspected the first 100 pages?
  - Would you feel safe over that airspace?

---

How to Inspect a large amount of specification or code!
Sample “During” Authoring 1

The Author is expected to write about 45 pages.

Then sample one page with Inspection.

4 Major

Too many defects

Good Enough

Write New Pages

Sample

4 Major

Exit?

No

Re-Write all 5 pages

Sample “During” Authoring 2

The Author is expected to write about 45 pages.

The Author re-writes all 5 pages based on the feedback from the one inspected page.

Sample

1 Major

Then sample another page with Inspection.

1 Major

Too many defects

Good Enough

Write New Pages

Sample

1 Major

Exit?

Yes

No

Re-Write all 5 pages
Sample “During” Authoring 3

Exited Pages

The Author is expected to write about 45 pages

Sample

1 Major

Then sample one page with Inspection

Write New Pages

1 Major

Sample Exit?

Yes

No

Re-Write all 5 pages

Sample Exit?

Yes

No

Re-Write all 5 pages

Sample Exit?

Yes

No

Re-Write all 5 pages

Sample Exit?

Yes

No

Re-Write all 5 pages

Version 16/4/2001 TomGilb@Result-Planning.com Slide 29

Sample “During” Authoring 4

Exited Pages

The Author is expected to write about 45 pages

Sample

1 Major

Then sample one page with Inspection

Write New Pages

1 Major

Sample Exit?

Yes

No

Re-Write all 5 pages

Sample Exit?

Yes

No

Re-Write all 5 pages

Sample Exit?

Yes

No

Re-Write all 5 pages

Version 16/4/2001 TomGilb@Result-Planning.com Slide 30

© TomGilb@Result-Planning.com
Sample “During” Authoring 5

Exited Pages

The Author is expected to write about 45 pages

5 Major

Sample

Now the Author can write 5 more pages.

Then sample one page with Inspection

Too many defects

Good Enough

Write New Pages

5 Major Sample Exit? Yes No

Re-Write all 5 pages

Sample Exit?

Re-Write all 5 pages

Too many defects

Good Enough

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?

Re-Write all 5 pages

Sample Too many defects

Good Enough

Sample Exit?
What Do We Check for?

- Mainly Majors
- Minors if in doubt
- Improvements to process
- Majors in any project document
  ⇒ Especially sources
  ⇒ Even Kin
- Inconsistencies
  ⇒ Not our job to ‘convict’
  ⇒ Just to ‘arrest’ on suspicion

Measurement
• Purpose
  ⇒ To determine the set of forms or computer templates necessary
  ⇒ To gather Inspection data
  ⇒ To report Inspection data

• Method
  ⇒ Access standard set
    ⇒ Master Plan Sl p 401
    ⇒ Data Summary Sl p 402
    ⇒ Editor Advice Log Sl p 404
    ⇒ Change Request Sl p 406
    ⇒ Process Brainstorming p 405

---

Forms Planning

- **Purpose**
  - To determine the set of forms or computer templates necessary
  - To gather Inspection data
  - To report Inspection data

- **Method**
  - Access standard set
    - Master Plan Sl p 401
    - Data Summary Sl p 402
    - Editor Advice Log Sl p 404
    - Change Request Sl p 406
    - Process Brainstorming p 405
The “Data Collection”
Part of the Master Plan

NOTE: the data is filled in by each individual checker after checking, and before the logging meeting.

Purpose of Planning

- RESULTS WE EXPECT (REAL PURPOSE)
  ⇒ plan Inspection to be ‘cost effective’.
  ⇒ to help product exit on first check (mostly)
  ⇒ to get “effectiveness”, at finding Majors
  ⇒ to get “efficiency” (cost per Major)

- STRATEGIES (HOW WE GET THE RESULT)
  ⇒ to gather all necessary documents and ‘tools’
  ⇒ to form a team of people, able and willing

- Main Elements of Planning
  ⇒ 1. Which Documents to check?
  ⇒ 2. Which Standards to apply?
  ⇒ 3. Team Size?
  ⇒ 4. Specialist role?
  ⇒ 5. How big a sample?
  ⇒ 6. How much time to use?
### Purposes of Data Summary Form

- To collect basic **costs and results**
- To put these data in a **database for all checking activity**
- To provide data for **judging exit and entry conditions of sub-processes**
  - including “real time” control of the process
  - for example entry to “Logging”
- To advise team leaders
- To provide a basis for **improvement of the process** (ex. optimum rate)
- To help us **evaluate the profitability of the process**

### Data Summary filled out after entry to logging

#### Table: Data Summary

|-----------------|---------|----------|--------------------|---------|------|

| DATA SUMMARY | | | |

**Checker Checking Pages (P)**
- Major + SM
- Minor
- Improvement
- ?s
- Checking Reporthours (t) studied
- issues
- issues
- improvement
- ?s
- New issues found

**Checking rate - average P/t.**
- (could this be a failed exit condition?)

**LOGGING MEETING DATA**

- No. of people
- Logging-duration (clock hrs)
- Logging-time (work-hours)
- Major + SM issues logged
- Minor issues logged
- Improvement suggestion of intent
- in the meeting

**Item - Logging rate**
- (items/minute)

**Logging-meeting-rate**
- (pages per hour checked)

**Detection-time**
- (Plan+Kickoff+Check+Log)

### EDITING, FOLLOWUP, EXIT, BRAINSTORMING AND FINAL SUMMARY DATA

- No. Major + SM defects:
- No. minor defects:
- No. Change Requests:
- Edit-time:
- Follow up time:
- Exit-time:
- Exit date:
- Control-time:
- Defect-removal-time:
- Est remaining Maj + SM defects/page:
- Est. effectiveness (% Maj defects found/page):
- Efficiency (Maj/wk-hr)

### Notes

- Total Brainstorming time (in work-hours)
- Development time probably saved by this check
  - based on net 8 [or ___] hrs./Major)
Data Summary (based on SI page 403 (improved))

<table>
<thead>
<tr>
<th>Inspection ID</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inspection Leader</th>
<th>Mail/Tel Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Document Reference</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total logical (300 Non-Commentary words/page) Checked Pages</th>
<th>of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date/time Inspection Requested</th>
<th>Date Entry criteria passed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checker Report</th>
<th>Pages Studied (P)</th>
<th>Checking hours (t)</th>
<th>Major+SM issues</th>
<th>minor issues</th>
<th>Improvement suggestions</th>
<th>% of intent</th>
<th>New issues found in the meeting</th>
<th>Checking Rate (P/t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDIVIDUAL CHECKING DATA (to be reported really during the entry process for Logging meeting)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOGGING MEETING DATA (fill in at the end of the logging meeting)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(N) Number of people</td>
</tr>
<tr>
<td>(D) Logging-duration</td>
</tr>
<tr>
<td>(6) Edit-time</td>
</tr>
<tr>
<td>(7) Edit Audit-time</td>
</tr>
<tr>
<td>(8) Exit-time</td>
</tr>
<tr>
<td>(9) Control-time</td>
</tr>
<tr>
<td>(10) Defect-removal-time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOGGING MEETING SUMMARY (All items logged during the logging meeting)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major +SM issues logged</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXIT RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major +SM defects fixed per logical page</td>
</tr>
<tr>
<td>Did the Inspection Process meet the Inspection Exit Criteria:</td>
</tr>
<tr>
<td>Yes or No</td>
</tr>
<tr>
<td>Did the document Exit the Inspection Exit Criteria:</td>
</tr>
<tr>
<td>Yes or No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EDITING, FOLLOWUP, EXIT, BRAINSTORMING AND FINAL COST SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>(6) Edit-time</td>
</tr>
<tr>
<td>(7) Edit Audit-time</td>
</tr>
<tr>
<td>(8) Exit-time</td>
</tr>
<tr>
<td>(9) Control-time</td>
</tr>
<tr>
<td>(10) Defect-removal-time</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FINAL FINDINGS AS REPORTED BY EDITOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Major +SM defects</td>
</tr>
<tr>
<td>fixed per logical page</td>
</tr>
<tr>
<td>Did the Inspection Process meet the Inspection Exit Criteria:</td>
</tr>
<tr>
<td>Yes or No</td>
</tr>
<tr>
<td>Did the document Exit the Inspection Exit Criteria:</td>
</tr>
<tr>
<td>Yes or No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EFFICIENCY ESTIMATES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Learning
Why have a meeting, Helena?

- **Motivation**
  - To do a great checking work, avoid being embarrassed by poor defect finding
  - Editor will be motivated to respect the opinions of people when given face to face as opposed to some “behind the back” attack.

- **Learning**
  - How good others are at finding defects (understand the potential)
  - How to be as good as others are at finding defects (which methods used)
  - Learning engineering practice (observe design and ideas)
  - Learning what has priority (Majors not minors)
  - Learning teamwork and meeting practices, how to productively exploit limited time (it spills over to all other meetings, is one experience)
  - Collating
  - Provide editor with a single log of issues (not multiple scratching)

- **Double checking, 15% more defects.**
  - Each “New” Major found is worth about 9 work hours saved
  - So, it is easy to justify the extra time needed as profitable
  - Swap to new checking roles, get fresh people to look at things with fresh perspective

- We do not do meeting, or parts of it (like double checking, or logging minors), if judged not profitable on basis of checking data!

---

Individual learning Curve

- **Individual Learning Curve**
  - The speed which the individual learns to follow the Rules,
  - As measured by reduced Major Defects found in Inspections

  → Notes:
  - Faster, earlier and more dramatic than “process improvement”
  - Never mentioned in literature as a measurable

Marie Lambertsson’s Learnability Curve, Ericsson, Stockholm, 1997
Motivation

Motivation ‘is Everything!’

- People are ‘sensitive’
- Avoid all ‘threats’
- Give ‘positive’ motivators
- Very many ‘details’ support this attitude
- You will respect this, or fail with Inspection!
- Do unto others, as you would have them Inspect your work!
Positive Motivators

• Group-work
• Team
• Freedom
• Learning
• Game
• Experiments
• Challenge
• Numeric Feedback
• Process Improvement
• Positive Leadership

Negative Motivators

• Time Pressure
• Result Pressure
• Personal Attacks
• Bureaucracy
• Small-minded Leader
• Personal-fault blaming
• Process corruption
• High volume/cleanup
Motivational Philosophy in New Inspections

- Intelligent Inspection
- Maximum Leverage
- Process causes Defects
- Trust people
- Empower people
- Allow experiment
- Let results decide
- Continuously improve

Exit Control
Overall Process Map

Exit

- Kickoff
- Checking
- Logging
- Process Brainstorming
- Edit
- Edit Audit
- Change Requests to project
- Data Summary
- Exit
- Master Plan
- Exit Conditions
- Correct conditions preventing exit
- OK all Exit Conditions
- Failed Exit Condition
- Do it Again
- Exited Product

Getting numeric control over all your work. Policy level

- All software engineering specifications (from contract to code) will be subject to formal entry and exit control.
- This is primarily numeric, and based on ‘Major defects remaining’ levels,
  - i.e. based on ‘economic suitability for downstream work processes’.
- Default level maximum 1 major defect remaining per page.
  - Rationale: to make sure that poor specification practices do not pollute downstream activity, and threaten time to market, human resources or product quality.

Example of real policy suggestion to top management.

September 2000
Is the Inspection Process Trustworthy?

- **If the Process cannot be trusted**
  - Then we cannot trust it to tell us how bad the Product document is
  - Then we cannot make a safe decision on release of the Product to other work processes

- **The main reason we cannot trust the Inspection process is ‘fast checking rates’**

    Check Process Exit Conditions:
    1. Has Editing Been Completed
    2. Were Checking rates within 20% of Optimum?
    3. Is Data Summary complete & in database?
    4. Do you as Leader feel the Process was done correctly; so data is valid?
    If so, go to next Exit-stage: is the document OK?
Is the Product Document ‘Economic’?

- The central decision is whether the Product document is going to cost us
  - More if we release it now
  - than if we ‘fight the Defects’ before release
- There is no question of ‘zero defects’
- There are many later ‘nets’ to catch Defects, including Inspection of ‘Sources’

Check Document Exit Conditions:

1. Do ‘Remaining Major Defects/Page’ exceed the Exit Limit?
   (0.2 or 2.0)
2. Any other data that is suspicious?
   (very large number of ‘minors’)
3. Was your sample Representative and large enough to make this decision?
4. Is the Author personally comfortable with release?

If you pass all these tests, then Exit the Product document.

Calculation of Remaining Defects for Exit

Remaining Major Defects/Page =

\[ \frac{\text{Major Defects (in Data Summary after Edit)} \times ((1 - \text{Effectiveness\% expressed as decimal}))}{\text{Effectiveness\%}} + \text{Fix-fail-rate \((17\%) \times \text{Major defects}\) (round up)} \]

\[ \text{divided by Pages (of 300 words, non-comment)} \]

**Example**

16 Major defects (edited - correction ‘attempts’- in 36 checked logical pages).
36 Logical Pages (of 300 words non-commentary)
60% Effectiveness for this document type (from your history. Range is 30% to 90%)
16 x \((1 - .60)/.60\) = 10.7 Majors remain unnoticed
17% x 16 = 2.72 (3 rounded up)
10.7 (remain) + 3 (fail fix) = 13.7 Majors total after edit remain in 36 page document.
13.7 Remaining / 36 = 0.37 Majors per page

This could exit (after edit at maximum 0.4 but not 0.3 per page)

Uncertainty of calculation
- we guess about ±30% of remaining Maj./Page
- it seems to work in checking predictions

Assumptions
- effectiveness is like history
- Inspection procedures are done as written
- checking rates are ±20% optimum
- or at least are near historic rates

Defect insertion rate during fixing is intentionally not included here for simplicity (about 3% of fixes?)
Can you exit on a sample?

- If a representative sample is clean enough (less than 0.2 Majors/Page))
  ⇒ you could decide to exit the entire document with no corrections

- If a sample is “garbage” (> 10 Maj./Pg.)
  ⇒ then it might be economic to “burn the document” and “make a new one”

- If the sample is “moderate quality” (0.2 to 10 Majors/page total)
  ⇒ then you might consider using Inspection to clean defects and find and clean the rest.

What Exit Means, and What it Doesn’t

- Perfection costs infinity!
- Imperfection has a price too!
- So Exit is an attempt to ‘balance’ these extremes
- We should Exit when we have more to gain than to lose.
- We should not exit when this will result in undesired delays and costs.

If we do not meet the formal Exit Conditions, can we release the Product Document?

- We can, but we should announce the risks in writing and quantitatively.
- Make sure management accepts the risks of delay and cost.
- If you choose to NOT EXIT
  - Explain the costs now, and alternative costs later.
Generic Exit Conditions 1 of 4

Process Validation Conditions

GXI.1 (Optimum Checking Rate)

The Average Team Checking Rates (at both Checking and Logging) are within 20% of the established optimum for that Product type.

In default of an Established Optimum Checking Rate, a similar document type Rate can be used.

In no case is an average Team checking rate greater than 600 Non-Commentary words per hour (2 logical pages/hr.) acceptable.

---

Generic Exit Conditions 2 of 4

GXI.2 (Team Leader OK)

The Team leader evaluates the entire inspection process in relation to taught, and defined process and is willing to state and be held personally accountable, that an acceptable process has been carried out, and that the numeric results are valid for the stated Inspection Purpose.

Note: the ‘deadly sin’ here is allowing non-optimum rates, and then drawing false conclusions about Remaining Major Defects (below) to be the basis for Exit.

GXI.3 (Facts Captured)

The Inspection data Summary sheet is competently filled out and the data is transmitted to the Inspection Database, and accepted as Valid.

Note: This is so that we can learn about our real process and improve it and our own practices. It is necessary to have Exit pressure here to get it done at all!
Generic Exit Conditions 3 of 4

Document Exit Conditions

GXI.4 (Remaining Majors)
An estimate of Maximum Average Remaining Major Defects per Logical Page (300 NC words) is made, based on known Effectiveness of a Valid Process (assume 30% in default of positive knowledge).
The Remaining Majors shall not exceed the Standard Limit for the Document Type.
The default Limit is 3 Majors/Logical page for immature Document Processes; and maximum 0.3 Majors/Logical Page for Mature Document Processes.

Calculation of Remaining Defects for Exit

Remaining Major Defects/Page =
Major Defects (in Data Summary after Edit) x \((1 - \text{Effectiveness}\% \text{ expressed as decimal}) / \text{Effectiveness}\% + \text{Fix-fail-rate \((17\%) x \text{Major defects} \text{ (round up)} \text{ divided by Pages (of 300 words, non-comment)}\})

Example:
16 Major defects (edited - correction 'attempts' in 36 checked logical pages).
36 Logical Pages (of 300 words non-commentary).
60% Effectiveness for this document type (from your history. Range is 30% to 90%).
16 x \((1 - .60)/.60 = 10.7\) Majors remain unnoticed
17\% x 16 = 2.72 (3 rounded up).
10.7 (remain) + 3 (fail fix) = 13.7 Majors total after edit remain in 36 page document.
13.7 Remaining / 36 = 0.37 Majors per page.
This could exit (after edit at maximum 0.4 but not 0.3 per page).

Uncertainty of calculation
we guess about ±30% of remaining Maj./Page
it seems to work in checking predictions

Assumptions
Effectiveness is like history.
Inspection procedures are done as written.
Checking rates are ±20% optimum
or at least are near historic rates.

Defect insertion rate during fixing is intentionally not included here for simplicity (about 3% of fixes?)
Generic Exit Conditions 4 of 4

GX1.5 (Author OK).

The Responsible Product Document Author/Editor approves Exit.

GX1.6 (Majors Edited)

The Edit Audit Process, conducted by the Leader, has satisfactorily verified that the intended Editing work has been completely and competently completed. There are no Known Majors unresolved through lack of time or effort.

GX1.7 <to be determined sometime, maybe by you, if you are the process owner>.

Can you exit on a sample?

- If a representative sample is clean enough (less than 0.2 Majors/Page)
  ⇒ you could decide to exit the entire document with no corrections
- If a sample is “garbage” (> 10 Maj./Pg.)
  ⇒ then it might be economic to “burn the document” and “make a new one”
- If the sample is “moderate quality” (0.2 to 10 Majors/page total)
  ⇒ then you might consider using Inspection to clean defects and find and clean the rest
Data Collection from “Checking”

Individually Checking Results (to be reported during the entry process for Logging meeting)

<table>
<thead>
<tr>
<th>Checker</th>
<th>Hours Checking</th>
<th>Pages Inspected</th>
<th>Major issues</th>
<th>Minor issues</th>
<th>Improvements</th>
<th>Questions</th>
<th>Checking rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>1.5</td>
<td>1.5</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>~0.83</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.5</td>
</tr>
</tbody>
</table>

- This data will be reported orally at the beginning of the logging meeting to the Leader.
- The leader will fill it in the data summary form.
- The data will be evaluated to determine whether, or how, the main part of the logging meeting should be carried out.

Entry Control
Simplified Inspections

Predicting bugs in test
Predicting bugs in field

Predicting Delays to Project
The costs of getting started

The costs of keeping it going
Doing it as
your system
evolves

Inspecting
maintenance
Changes
Process Brainstorming

- **Purpose:**
  - To get ‘grass roots’ insights as to root causes and cures for defects.

- **Method:**
  - Use up to 30 minutes
  - Log opinions about up-to 10 Major defects (3 min. each)
  - Don’t go deeper!
  - Leave deeper analysis to Process Improvement Teams
The PB Process

- Team Stays together after ‘Logging’
  - Same room
  - Same people
  - Maybe a break first
  - Same documents
  - Up to half an hour

- Shift mentality!
  - Not the project
  - The process, our organization
  - How we feel it can be improved for us
  - So we are not ‘forced’ to make mistakes

Quick Break from Logging

Leader Picks a real Sample Major Issue from Log, tells Team (1 min)

Team Brainstorms ‘Root Cause’ (1 min)

Team Brainstorms ‘Cause Cure’ (1 min)
Process Brainstorming Log (filled out)

<table>
<thead>
<tr>
<th>Brainstorm Item No.</th>
<th>Issue Ref.</th>
<th>Classification of Cause (StA)</th>
<th>Root Cause (keywords)</th>
<th>Improvement Suggestions (keywords)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>Communication Oversight Transcription Education</td>
<td>No Source used like Corp. Policy</td>
<td>Entry requires proper source</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>Communication Oversight Transcription Education</td>
<td>Failure to use main strategy</td>
<td>Slower check rate</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>Communication Oversight Transcription Education</td>
<td>Author did not understand generic rules (quantity)</td>
<td>Re-train Leaders to understand generic rule use</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Communication Oversight Transcription Education</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Communication Oversight Transcription Education</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Communication Oversight Transcription Education</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The PB Log

<table>
<thead>
<tr>
<th>Item</th>
<th>Issue</th>
<th>Classify</th>
<th>Root Causes</th>
<th>Improvement Suggestions</th>
</tr>
</thead>
</table>
| 1    | 10    | Oversight | • Time pressure  
• no tools  
• no info | • optimum time  
• build tool  
• give info on PC |
| 2    | 8     | Education | • trainees don’t know  
• manual not updated | • special meeting for trainees  
• manual on Web |
| 3    | 3     | Communication | • authors are unknown | • publish their email on doc. head |

- Brainstorming Rules: no criticism, flow ideas in
- Getting ‘Grass Roots’ opinions, investigation later
Process Brainstorming: the bigger picture (DPP)

The Road To Wisdom : Piet Hein

"The road to wisdom is plain and simple to express, to err, and err, and err again, but, less, and less, and less."

Piet Hein, (Danish Philosopher)
Defect Prevention Experiences

% of usual defects prevented
90%
80%
70%
60%
50%

(Why not?) example Raytheon 95
8 years 45% rework down to 5%

North Carolina


- 2X to 3X reduction in defects within 2 years.
- ROI Overall, incl. production fix saving 13 to 1
- 1985-94, 200 people dev.org 1,822 actions sugg. 61% impl.
DPP total costs 1% of org. resources (for 227 person org.)

Robert Mays
- P O Box 12195
- Dept. A82, Bldg 8503
- RTP NC 27709
- (919) 254 5210
rmays@us.ibm.com

Prevention + Pre-test Detection

Test & Use

90%
80%
70%
60%
50%
40%
30%
20%
10%
0%

70% Detection by Inspection
<- Mays 1993, 70% prevented
<- Mays & Jones 50% prevented (IBM) 1990

95% cumulative detection by Inspection (state of the art limit)

“Detected”
“Prevented”

- Prevention data based on state of the art prevention experiences
  ⇒ (IBM RTP, NC)
- Cumulative Inspection detection data based on state of the art Inspection
  ⇒ (in an environment where prevention is also being used, IBM MN, Sema UK, IBM UK)
The Process Brainstorming Aftermath

- Brainstormed suggestions
  - Are input to Process Improvement Teams.
  - Are part of the inputs
    - & cost of defect data
    - & frequency of defect.
  - PB Insights are
    - Accurate
    - Decentralized
    - Real time
    - Socially acceptable
  - Proven (Mays) to work better than centralized efforts (Fagan's Method 1973)

Defect Prevention Process within 'Inspection'

- Process Brainstorming
- Field Operation Fault Data
- Quality Assurance Database
- Product Testing & Fixing Fault Data

PROCESS CHANGE MANAGEMENT TEAM

Select Improvement Target (Pareto analysis)
Delegate Analysis and Design of Improvement to 'Process Investigators'
Evaluate Effect of Trial Improvements on real project
Spread Improvements with your organization

See 'Software Inspection' Chapters 7 and 17 for detail.
Policy for Continuous Improvement

See SI page 431 Appendix E: Policies

- Simplified version
- CQI-1. Use formal Defect Prevention Process (Ch.17)
- CQI-2. Track DPP profitability
- CQI-3. Measure all quality improvement (process and product)
- CQI-4. Use Entry and Exit
- CQI-5. Use 'process owners'
- CQI-6. Improve by 40% a year
- CQI-7. No exit > 3 Majors/page
- CQI-8. Management is responsible.
- CQI-9. Grass roots suggests changes daily
- CQI-10. Priority to defect avoidance

Source: Software Inspection page 431 (Inspection->Document Quality Improvement)

Defect Prevention Process (DPP) at Raytheon

- Prevention is Better Than Cure
- Prevention = 'avoiding defect insertion'
- Detection = 'find & fix'
- Key concept: Decentralization, Many small wins, Measurable progress towards goal

Raytheon 1995 SEI Report

Start of effort

Individual Learning effect

Bad Process change

Cost for doing it right COC

Cost for doing it wrong(ly)
(Cost of Non-Conformance)

Raytheon 1995 SEI Report

Figure 8: Cost of Quality Versus Time
CASE STUDIES

CASE OF INSPECTION ON SILICON VALLEY STARTUP E-BUSINESS BUSINESS PLANS 2000

COURTESY OF BILL B.
ONE HOUR INSPECTION RESULTS

1.1. Aggregate Summary

- Round #1 Averages (n=8)
  - Majors/Page: 35
  - Majors/Page (according to author): 19
  - Average Edit Time: 2.2 hours
- Round #2 Averages (n=7, omit outlier marked by *)
  - Majors/Page: 17
  - Majors/Page (according to author): 9
  - Average Edit Time: 1.25 hours

1.2. Major Defect Counts

<table>
<thead>
<tr>
<th>Document</th>
<th>Major Defects (Inspection #1)</th>
<th>Major Defects (Inspection #2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>25</td>
<td>17</td>
</tr>
<tr>
<td>B</td>
<td>31</td>
<td>17</td>
</tr>
<tr>
<td>C</td>
<td>45</td>
<td>24</td>
</tr>
<tr>
<td>D</td>
<td>44</td>
<td>11</td>
</tr>
<tr>
<td>E</td>
<td>35</td>
<td>15</td>
</tr>
<tr>
<td>F</td>
<td>45</td>
<td>46*</td>
</tr>
<tr>
<td>G</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>H</td>
<td>49</td>
<td>25</td>
</tr>
</tbody>
</table>
1.1. What went well

- One page is actually useful: impact felt throughout rest of doc. Focus on one section identifies blind spots
- Commitment of inspection team
- Good spreadsheet
- Raised awareness of doc quality

1.2. What did not go well

- One page is too short: not enough background
- Inflexible process
- Lots of comments were not helpful
- Rules not useful
- "Enough" rule was a Trojan horse putting focus on concept rather than clarity. Authors would take out context to fix an Enough comment
- "Clear" rule makes authors add more text
- Don't do inspections during crunch time
- Paper trail too oppressive
1.3. Suggested Improvements

- Certified writer program
- Awards (Dunce Cap)
- Make inspections ongoing, don't tie to Plan Complete
- Provide more "source" information: use most of doc
- Read all of doc and circle problems
- Skim doc before starting inspection
- Offline reading: split logging meeting into a separate meeting
- Enable a constant feedback loop
- Inspect after concepts solidified, then focus on clarity
- Docs should be approved pending inspection
- Pre-inspect before biz tech?
- Look for high level feedback across doc
- Improve rule lists. Use 3-4 rigid rules: inconsistent, ambiguous, not measurable
- Provide example of majors
- Biz Tech Review -> Inspect -> Biz Tech Approve
- More cross-functional review teams
- Audit product plans and review to see whether requirements fit
- Key question: when in doc lifecycle?

OPINIONS ABOUT THE MINI INSPECTIONS

Averages for Quantitative Questions
A score of 5 means "Strongly Agree" and a score of 1 means "Strongly Disagree." Sixteen surveys were returned.

<table>
<thead>
<tr>
<th></th>
<th>Inspections improve doc quality</th>
<th>Inspections improve system quality</th>
<th>WE should continue using inspections</th>
<th>Inspections were a good use of time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>3.94</td>
<td>3.75</td>
<td>3.81</td>
<td>3.31</td>
</tr>
<tr>
<td>Author Only</td>
<td>4.20</td>
<td>4.00</td>
<td>4.20</td>
<td>3.80</td>
</tr>
<tr>
<td>All Author</td>
<td>4.00</td>
<td>3.71</td>
<td>3.71</td>
<td>3.29</td>
</tr>
<tr>
<td>Checker Only</td>
<td>3.25</td>
<td>2.75</td>
<td>3.25</td>
<td>2.25</td>
</tr>
<tr>
<td>All Checker</td>
<td>3.82</td>
<td>3.64</td>
<td>3.64</td>
<td>3.09</td>
</tr>
<tr>
<td>Leader Only</td>
<td>4.40</td>
<td>4.60</td>
<td>4.40</td>
<td>4.20</td>
</tr>
</tbody>
</table>
What did they like best about the Inspections?

<table>
<thead>
<tr>
<th>Author</th>
<th>As an author, I got focused attention from key areas on my document and writing style.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author</td>
<td>Getting input on readability and clarity on my documents from many members of a group.</td>
</tr>
<tr>
<td>Author</td>
<td>Getting review from ops, technology, and biz process – each distinct perspectives.</td>
</tr>
<tr>
<td>Author</td>
<td>Pointed out a number of items that needed strict clarification.</td>
</tr>
<tr>
<td>Checker</td>
<td>Excellent feedback from peers.</td>
</tr>
<tr>
<td>Checker</td>
<td>I think it gives the folks in Biz Process a better understanding of the types of gaps that often appear in their documents. If they can do a better job up front, we can do less work in Development related to closing these gaps after the product plans are frozen.</td>
</tr>
<tr>
<td>Checker</td>
<td>Open, constructive feedback process.</td>
</tr>
<tr>
<td>Checker</td>
<td>Keep employing the sampling approach – and rotate through documents.</td>
</tr>
<tr>
<td>Checker</td>
<td>That we were taking a hard look at people's work.</td>
</tr>
<tr>
<td>Checker-Author</td>
<td>Received comments back from individuals who were closely tied to the requirements (technology/operations). Feedback is always great to receive and makes the document that much more robust.</td>
</tr>
<tr>
<td>Checker-Author</td>
<td>That people actually proof the document.</td>
</tr>
</tbody>
</table>

Sample of what people liked least

- **Checker-Author**

- Very time consuming, and a process that should simply be done when documents are sent out for review by peers. I've sent out the same document to the same people that were in the inspection previous to the inspection. However, because it wasn't under the guise of a formal session, they did not take the time to read over the underlying issues that were uncovered in inspection. As a result, I don't think inspections are necessarily the reason, I just think people need to take the time to proof-read each other's documents and read through them with a more fine-tooth comb.
The “Jet” Review

This USA Multinational makes Jet components. Case from Fall 1999

Inspection of System Requirements Specification (SRS) of 82 pages

This presentation shows how to carry out a one-half hour inspection with senior/middle managers. The purpose is to show to make managers aware that they play a key-role in creating project delays by approving poor quality of requirements documents.

The inspection results shown in this real-life example successfully predicted a project delay of at least 2 calendar years.

Poor quality marketing requirements documents prove time and again to be a good predictor of project delays. The clue is that requirements documents with a high defect density are an indicator of a truly unprofessional engineering culture.
Rules

Introduce the following three rules for inspecting a requirements document:

Three Rules for Requirements:

⇒ 1. Unambiguous to intended Readership
⇒ 2. Clear enough to test.
⇒ 3. No Design (how to) mixed in
   → with Requirements (how well)
   → MARK Design as “D”

Defect

Explain the definition of a Defect:

• A Defect is a violation of a Rule

• Note: If there are 10 ambiguous terms in a single requirement then there are 10 defects!
Severity

Explain:
- the definition of **Major** Defect
- the checkers must focus on finding Major Defects

- **Major**: a defect severity where there is potential of
  ⇒ high (10 lost engineering hours) loss
  ⇒ later downstream (test, field).

Exit?

Agree with the management team on a numeric exit condition

- **Exit Conditions**: (Requirements can go to Design, Test etc with little risk)
  ⇒ *Maximum 1 Major Defect/ (Logical) Page*

*Logical Page* = 300 Non commentary words.
The Job

- You have up to 30 minutes for checking one requirements Logical page from an 82 pages document
- Count all Rule Violations = Defects
- Classify Major and minor

Report for Page 81

(Reported inspection results on requirements document, 4 managers)

- Tot=24, Maj=15, D=5
- 44, 15, D=19
- 55, 20, D=4
- 22, 4, D=2

- Defect Density
- Total for group 20 x2 = 40 Majors assume are unique
- If 33.333% effective, total in page = 3 x 40 = 120
- Of which 2/3 or 80 were not yet found.
- If we fix all we found (40), then the estimated remainder of Majors would be 80 (not found) + 8 not fixed for real = 88 Majors remaining.
Report for page 82

(reported inspection results on requirements document, 4 other managers)

- Total Defects, Majors, Design (part of Total and M&m)
  - 41, 24, D=1
  - 33, 15, D=5
  - 44, 30, D=10
  - 24, 3, D=5

- Team would log unique Majors about 2x30=60
- Which is 30% of total, so total this page is about 180 Majors

- If we attempt to fix 60 we log, and correctly fix 5/6 then 10 are failed fixes, so:
  - The total remaining after inspection and editing = 10+120 =130 Majors per page.

Extrapolation to Total Majors in Whole Document

- Page 81: 120 majors/p
- Page 82: 180 Majors/p
- Average 150 Majors/page x 82 page = 12,300 Majors in the document.

- If a Major has 1/3 chance of causing loss
- And each loss is avg 10 hours then total project Rework cost is about 41,000 hours loss.
- (This project was over a year late)
  \[ \Rightarrow 1 \text{ year} = 2,000 \text{ hour} \times 10 \text{ people} \]
ASSUMPTIONS when extrapolating total majors in the entire document

⇒ **TEAMS WILL FIND DOUBLE OF THE BEST SINGLE OBSERVATION**

→ Largest Majors, estimate team would find about

⇒ **TEAM IS 30% EFFECTIVE (for inexperienced teams)**

→ Total is thus about 3 x
→ = Major defects/Pages

⇒ *If Exit required less than 1 Major/page we are 3 orders of magnitude (factor 1000) away from safe economic exit.*

---

**Letter to Your Boss**

31 March 2000, NL

Boss!

We have 2 options for the 82 page Requirements document.
Our sample shows that we have 180 Majors/Page.
We can spend 180 hours per page removing them with Inspection
We can rewrite the pages at a cost of 10 hours each.
Or we can suffer 30% of these as bugs and fault, at an average removal cost of about 10 hours each (test and field debugging and re-testing). 1/3 of 180 x 10 = 600 hours per page if we do not rewrite (10 hours/Page) or remove before test (180 hours/Pages).

We suggest rewrite (changing something to avoid defect injection rate). But you have said you are against this. So we have to tell you that your option will delay our project by 600 hours x 82 = 49,200 hours.
Our project has 10 people on it, and they can do about 2,000 hours per year. So that is 20,000 work hours per year for our team. The approximate delay for your decision not to rewrite is this about 2.5 years worse Time To Market.

We will of course do what you say, but we wanted to be sure that you understood what your boss will blame you for later.

Your Loyal Servant,  Tom
CASE SILICON VALLEY
CHIP DESIGN COMPANY

• APRIL 2001

Real Example: Clear & Ambiguous rules: 12:57 Wednesday

• One cycle consist (sic!) of 46 different phases (outermost LEDs are counted only once).

• Issue log
  ⇒ Outermost: both unclear and ambiguous
  ⇒ Why Unclear (‘not testable’): there are no parameters defining what is ‘outermost’
  ⇒ Fix = Outermost:Defined As: the single LEDs which are at the physical extreme both sides of the circuit board, for any given straight line row of LEDs
  ⇒ Severity:Major
  ⇒ Why Ambiguous:
    → Interpretation 1: the right most outermost
    → Interpretation 2 : the left most outermost

• Rules
  • Unclear: All specifications must be clear enough to test
  • Ambiguous: All specification must be Unambiguous to the Intended Readership.
The consequence of a few rules...

- Using 10 minutes
  - Mark on the sheet all potential violations (issues -> defects) of the Unambiguous/Clear rules, classify as probably (M) ajor or (m)inor
  - 10 FIM cash tax free to finder of most Majors!
  - Reports Majors: 10, 20, 15, 20, 12, 13, 12, 7, 4, 7, 12, 11, 11, 16, 7, 14, 6, 12, 8, 11, 19, 7, 8, 3
  - Estimate the number of Majors on the entire page,
  - Highest score 20
  - Team score probably about ~ 2(3?) (for small teams of 2 -5 people) x 20 =40 for 22 people I alter the estimate to 3x 20 +10=60+15, also we used only 15 minutes. Optimum effectiveness (maximum find) needs about 1 hour: so add at least 50% 60 +50% ~ 90.
  - Now this kind of SQC (Spec Quality Control) is 30% to 90% effective, the highest % take 5 to 8 years of culture improvement. The lower number is more realistic and conservative. 33.3% 3x 90= 270 Majors ±100
- And this is only for TWO RULES: if 20 rules maybe we have 1800 Majors/page??
- If we fixed all 50 we found, we woud still have 180 (not found yet) +18 not fixed correctly = ~200 Majors per page.
- If about 1/3 of these caused delay or fault (~67) and the delay was about 10 hours then this page, after correcting would delay you project by 10 x 67 = 670 engineering hours,
- We have a 20 page data sheet so the total project delay projected is about 20 x 670 hours = ~ 14,000 (@ 2,000 hours year) 7 work years lost as a result of this sloppy work.
- EXPERIENCE: at GE, IBM, Ericsson we have proven this calculation works
- ASSUMPTION: these are really Majors: the defects can really cause delay!
  - In this case we suspect the documentation is ignored (with good reason!).
- CASE STUDY: RON RADICE (IBM) PAPER FOR LUIS IN THE CD
- TECHNICAL NOTE: BELLCORE (HON PENCE 93) 42% BUGS DUE SPECS DEFECT, TRW (78) 62% BUG IN MILITARY SPACE SOFTWARE DUE TO SPECIFICATION DEFECTS GIVEN TO CODERS!

(Specific) Rules for **Structural Diagrams**: Thursday 11:41

- 1. (Generic) The Generic Rules apply (Clear, Unambiguous, etc.)
- 2.< (Naming) Name (uniquely, descriptively) <all> blocks and **HDL-readable** signals. It is not required to name the \{non HDL readable signals\} signals. This name will be used in the HDL code.>
- 3. (Hierarchy) Each diagram, which represents one level of a hierarchy, should have all (a complete set) of its sub-blocks included.
  - The sub-blocks must be detailed in other diagrams, if there is more detail to express.
  - Unanimous temporary vote for this 12:34
More Structure Diagram rules

1. (Generic) The Generic Rules apply (Clear, Unambiguous, etc.)
2. (Naming) Name (uniquely, descriptively) all blocks and HDL-readable signals. It is not required to name the {non HDL readable signals} signals. This name will be used in the HDL code.
3. (Hierarchy) Each diagram, which represents one level of a hierarchy, should have all (a complete set) of its sub-blocks included. The sub-blocks must be detailed in other diagrams, if there is more detail to express.
4. (Sub-block presentation quantity per page)
   Definition: There should never be more sub-blocks on a single page than the number which gives a clear overview to the intended readership. This number is generally not more than ten.
   Severity: minor
5. All structure diagrams, at any one level, should fit into an A4 page and still be readable (without microscopes). (approved 100%)
6. (Heading) A structure diagram must have the following headings:
   ⇒ Author, Date last change, Version, Name of Block
   ⇒ But these headings are not mandatory.
   * QC Status (Exited?), Major Defect level Remaining, Confidentiality, Project Number, Owner, Approver, Version,
   ⇒ Severity:?
7. Data Flow Direction: left to right for main average data flow, not feedback loops.
8. Clearly distinguish between combinatorial and sequential logic by specifying the graphic clock symbol (>) inside lower left of rectangle for sequential logic.
9. Signals: Asynchronous and pulse shall be identified by.....
10. <naming rules we are going to make will cover this and more>

SIMPLIFICATION RULES
10. (Marko Set Symbols) Do not draw clock and reset symbols if you only have one set of them. Do so when there are many sets on the same diagram.
11. Use <symbols which are well understood but contain a lot of logic> (example a counter):
   ⇒ Reference: Marko’s Set of Well-Understood-but-Contain-a-Lot-of-Cool-Logic Symbols.
   ⇒ Defect: symbols outside this set which could be expressed using the current defined Marko’s set.
QC exercise: 15:15 Thursday

- 3 different diagrams allocated to 3 teams of about 15 people each (Hand, Big, Simple)
- Agree on a team strategy to maximize major defect harvest per diagram
  - 1. Use 15 minutes per diagram
  - 2. Mark out every suspected violation of the rules
  - Write the rule number violated ‘7’. Next to the defect
- NOTE DOWN: ANY IDEA OF BETTER, MORE RULES AS YOU WORK!
- 3. Classify and count all Major Defects (downstream cost potential defects)
- 4. Give me your Major count at the end.

RAW DATA COLLECTION 15:50 THURSDAY

- HAND
  - Boxes: ~7
    - 5
    - 5
    - 8
    - 8
    - 8
    - 8
    - 6
    - 8
    - 10
    - 7
    - 6
    - total unique majors 20-30?
    - Team says 16

- Simple
- Majors
  - Boxes: ~30
    - 15
    - 8
    - 4
    - 39-> NOT MAJORS 10 WERE
      - 5
      - 5
      - 4
      - 5
      - total Unique? 15
- Density 0.5/Box

- Big: xx Boxes: ~36
  - USING EXTREME ROLE SPECIALIZATION
    - 5 TEAMS EACH OWN RULES
  - 25
  - 8
  - 1
  - 1
  - 34
  - 36 -> 72±10
  - 36
  - 3
  - (team of 3)
  - 1 (for 3)
  - TOTAL 66 UNIQUE MAJORS (137 M&m)
  - Density FOUND=66/36=~2
  - Hypothesis: Clever role allocation can improve FIND by 4x ?
  - Complicating factors: different documents
Extrapolation of the sample

- 66 Majors by the team
- 36 Boxes =~2 Majors/Box
- NO QUESTION OF EXIT,
  ⇒ VOTE (ALL - 1 (WHOLE PICTURE ARGUMENT)
- ESTIMATED EFFECTIVENESS OF THIS QC 33.3%±10%
- IF SO THEN THERE ARE 3 X 66 MAJORS IN THE DIAGRAM = ~198 ±33?
- IF YOU -> ~0 (<1 MAJORS/DIAGRAM) GILB WILL GIVE 10,000 FMK FOR A PARTY (MAY EVEN COME HERE MYSELF) SIGNED TOM S GILB
- + ONE FREE ADVANCED COURSE

EXIT CONDITIONS FOR STRUCTURE DIAGRAMS

- Maximum 1 Major/Defect remaining per A4 Diagram before exit to any other process.

- Intended audience/readership of structure diagrams:
  ⇒ The designer who wrote the diagram,
  ⇒ Review committee who will help the designer find bugs
  ⇒ Manager of designer
  ⇒ Customer reads
  ⇒ Other designers later, redesign
  ⇒ New student trainee engineer,
  ⇒ new employee
Various unsorted unfinished rule ideas

4. (Sub-block Creation)
   ⇒ Definition: Sub-blocks must be created when it is necessary in order to maintain engineering control of any critical aspects of the design; such as implementation, testing, cost, qualities etc.
   ⇒ Test [Sub-block Creation] if you can imagine potentially any requirement of the system or our engineering process, or any stakeholder concern, which is threatened by lack of sub-block specification, then the rule is violated (a potentially defective specification is found).
   ⇒ Justification: do deep enough engineering to control the technical and economic characteristics of the product.
   ⇒ Severity: Major

5. The structure diagram should represent all the {interfaces, } <which ones> amongst the technical functions described in the technical specification (Data Sheet). Talk to IIW about this one!

End slide, Tom Gilb