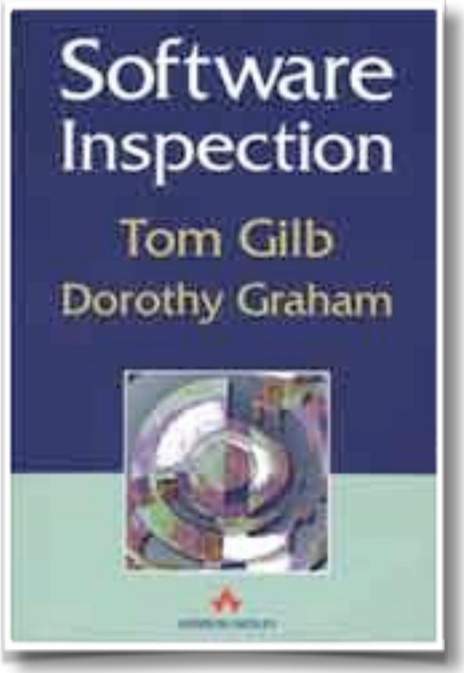
Lean Inspection (Spec QC)

ソフトウェア
インスペクション
Tom Gilb，Dorothy Graham／著
伊土 誠一・富野 壽／監訳
発行：株式会社構造計画研究所　発売：共立出版株式会社
INSPECTION

By Tom & Kai Gilb
www.Gilb.com



© 2011 Kai Gilb

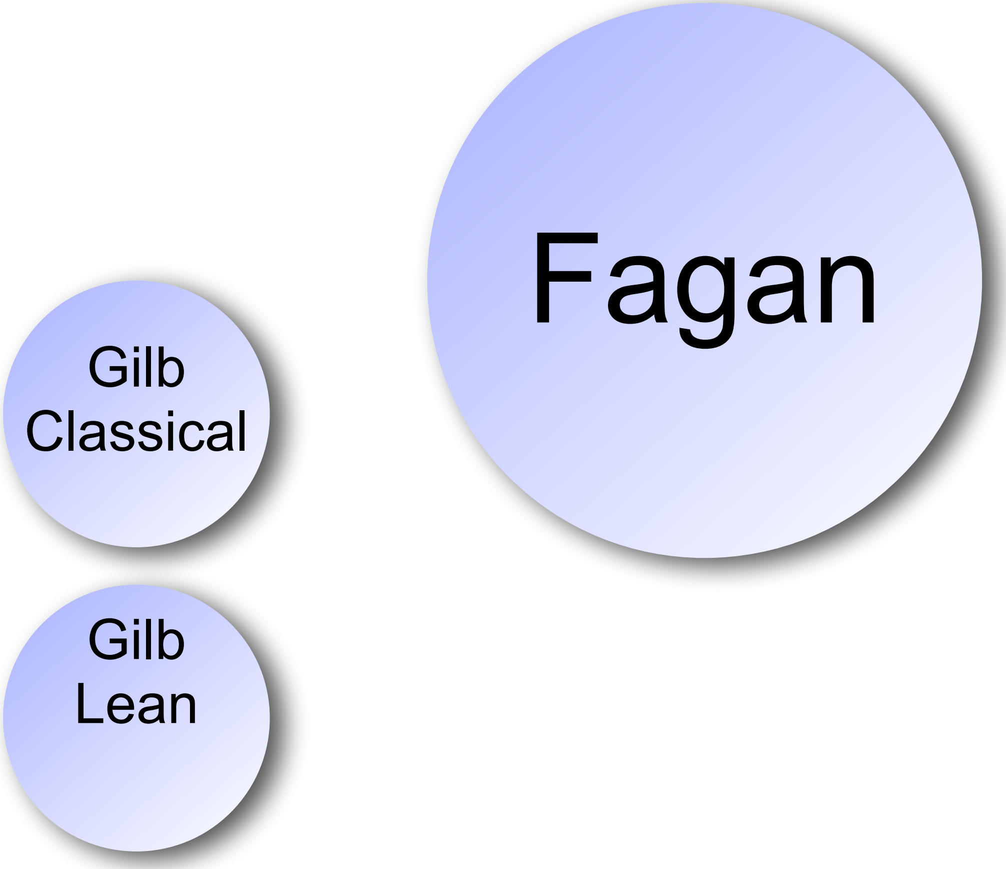
4 Types of Inspection

Fagan

Gilb Classical

Gilb Lean

4 Types of Inspection



Fagan

Gilb Classical

Gilb Lean

Conventional (IBM, Fagan, 1973)

Good Solid Practice

Optimum Checking Rates

Numeric Entry / Exit

etc.

But also:

No sampling

Inflexible bureaucracy

Focus on ‘Cleanup’

Focus on ‘software code’

Poorly documented process

“My Way”<--MEF/IBM

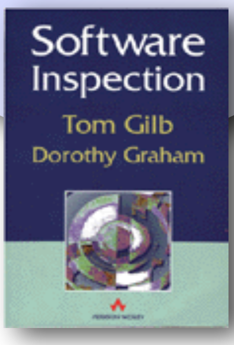
Do the defined process!

‘Interpret’ the document

4 Types of Inspection

Fagan

Gilb Classica


Software Inspection
Tom Gilb
Dorothy Graham

Gilb Lean

‘Advanced’ Inspections (Gilb 1993-1998)

Sampling to measure doc quality to decide if large cleanup is economic.

‘Intelligent Inspections’

Focus on Time & Control

Systems, upstream focus

Richly documented (Book)

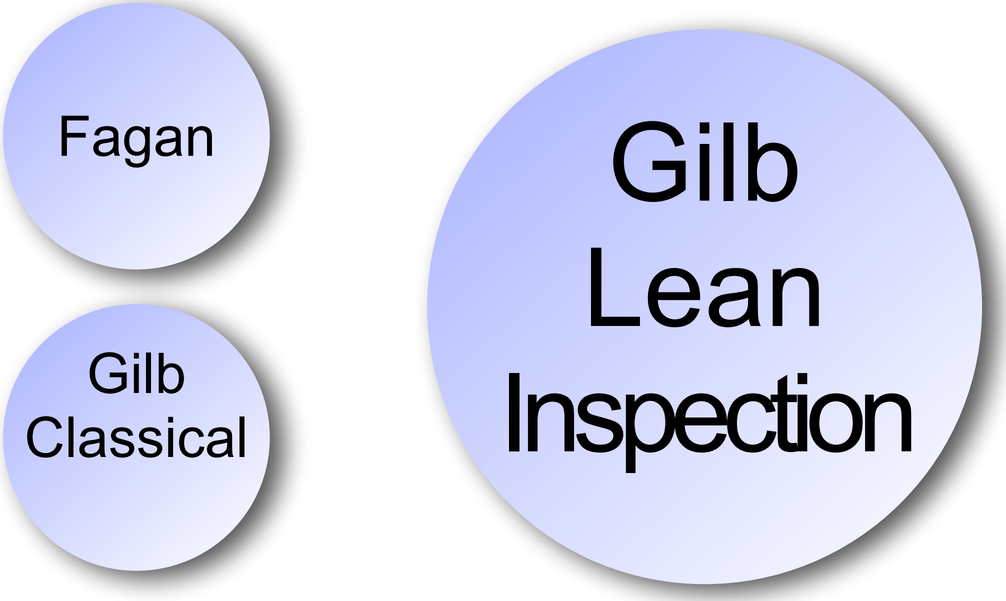
‘Our Way’ & ‘Your Way’

Do what pays off, *only*

Check against Rules, Sources.

Defect Prevention

4 Types of Inspection



Fagan

Gilb Classical

Gilb Lean Inspection

Gilb Lean / Agile

High Value for Effort

ONLY Sampling to *measure* doc quality.

NOT about cleaning bad work.

Focus on Exit Level

Motivate people to not insert defects.

Focus on Time & Control

Do what pays off, *only*

Check against Rules, Sources.

4 Types of Inspection

Fagan

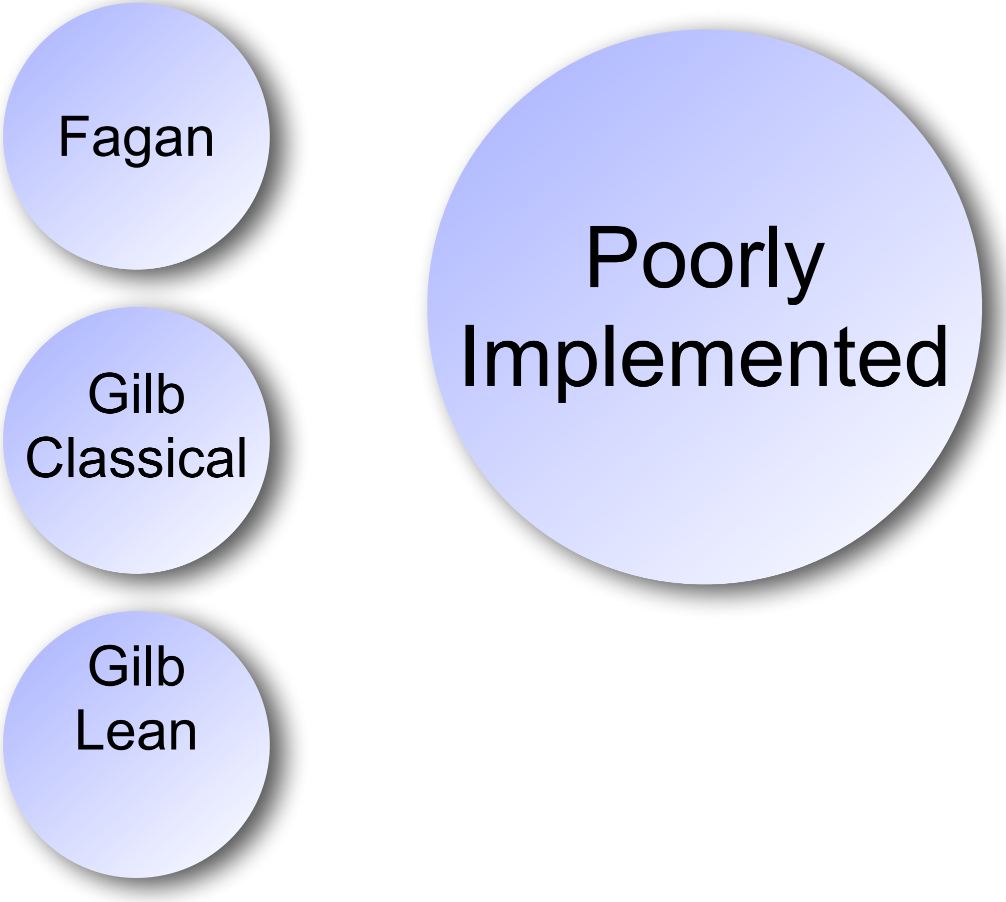
Gilb Classical

Gilb Lean

4th type

?

4 Types of Inspection


Fagan

Gilb Classical

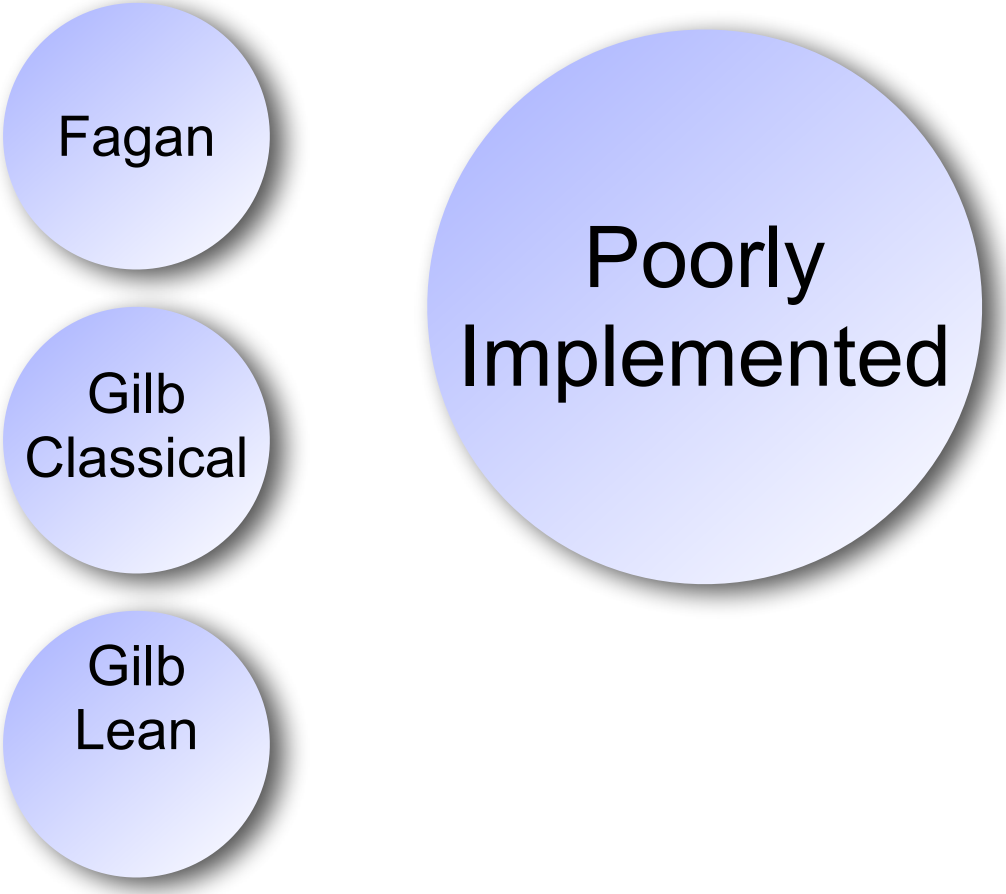
Gilb Lean

Poorly Implemented

With Poorly Implemented, I mean.

- No optimum checking rates.
- No numeric Entry / Exit Criteria.
- Not rule based
- No Sampling
- Only operating at ‘downstream’ level.
- Unknown Effectiveness
- Unknown Project Savings
- No Defect Prevention

4 Types of Inspection



Fagan

Gilb Classical

Gilb Lean

Poorly Implemented

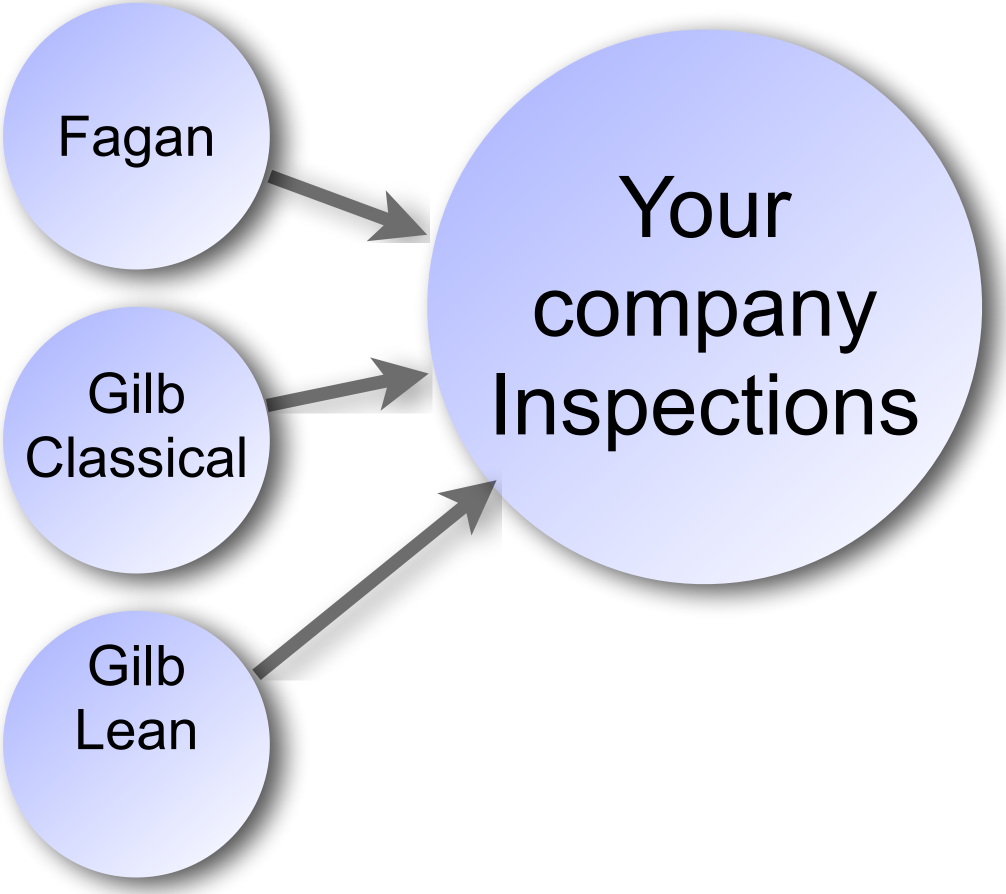
Why Poorly Implemented

Many companies base their Inspections on Fagan’s Inspection.

Fagan did not document it well, and does not prefer to share.

Unless you are trained by him, chances are you are doing Inspections poorly.

4 Types of Inspection



Why use Lean Inspection?

Many different types of benefits can be achieved by using Lean Inspection.

Two objectives that are central and typically focused on are:

Internal: Defect Density per Page.

External: Time to successfully complete a project.

External Goal

Project Efficiency

Scale: Total project time to successfully complete a project

Past [Jan. this year] **xx**

Goal [Jan. next year] = **70% of Past**

Goal [Jan. two years from this year] = **50% of Past**

NOT Objectives of Lean Inspection

Find and Fix Defects.
Approve document ‘content’ versus ‘Real World’.
‘Improve’ Quality of your end product.

Threat / Defect

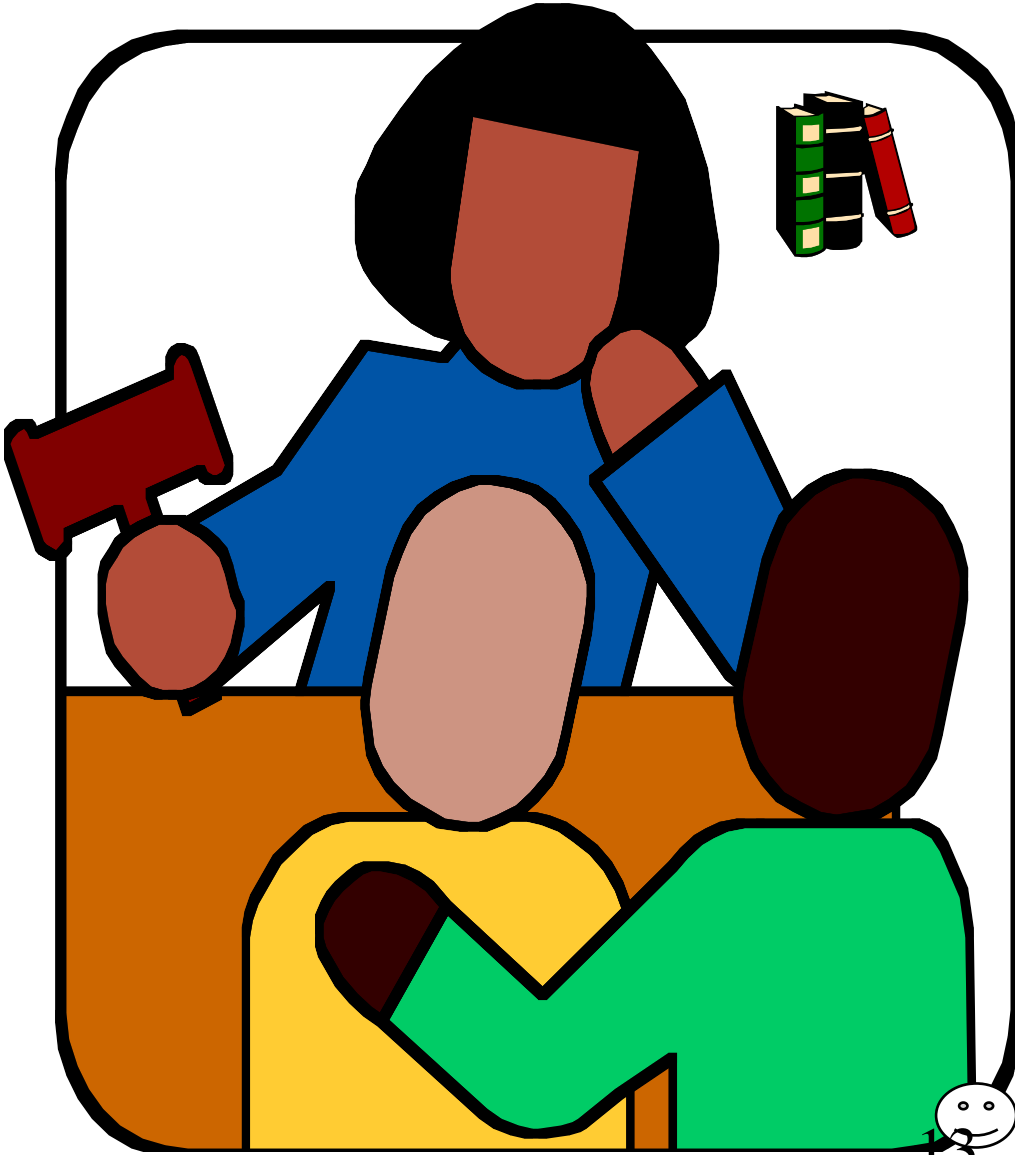
Definition:

Violation of a Rule

Notes:

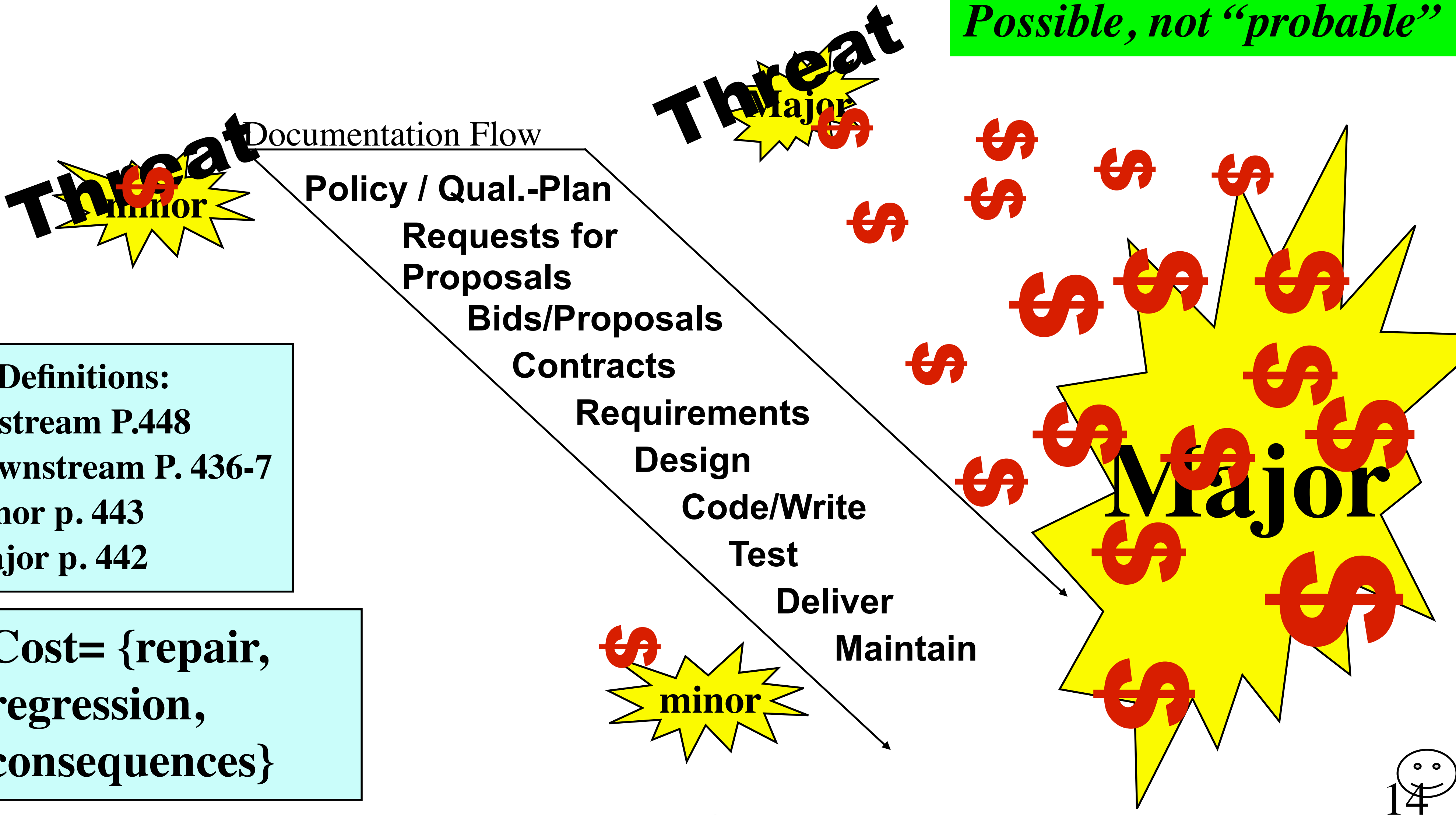
Rules are *official* standard for how to write documents.

We are primarily concerned with 'Major' Threats , rather than minor Threats .



Severity Concepts (minor, Major “cost later”)

Possible, not “probable”



Documentation Flow

Policy / Qual.-Plan

Requests for Proposals

Bids/Proposals

Contracts

Requirements

Design

Code/Write

Test

Deliver

Maintain

SI Definitions:
Upstream P.448
Downstream P. 436-7
minor p. 443
Major p. 442

Cost= {repair, regression, consequences}

Internal Goal

Density:

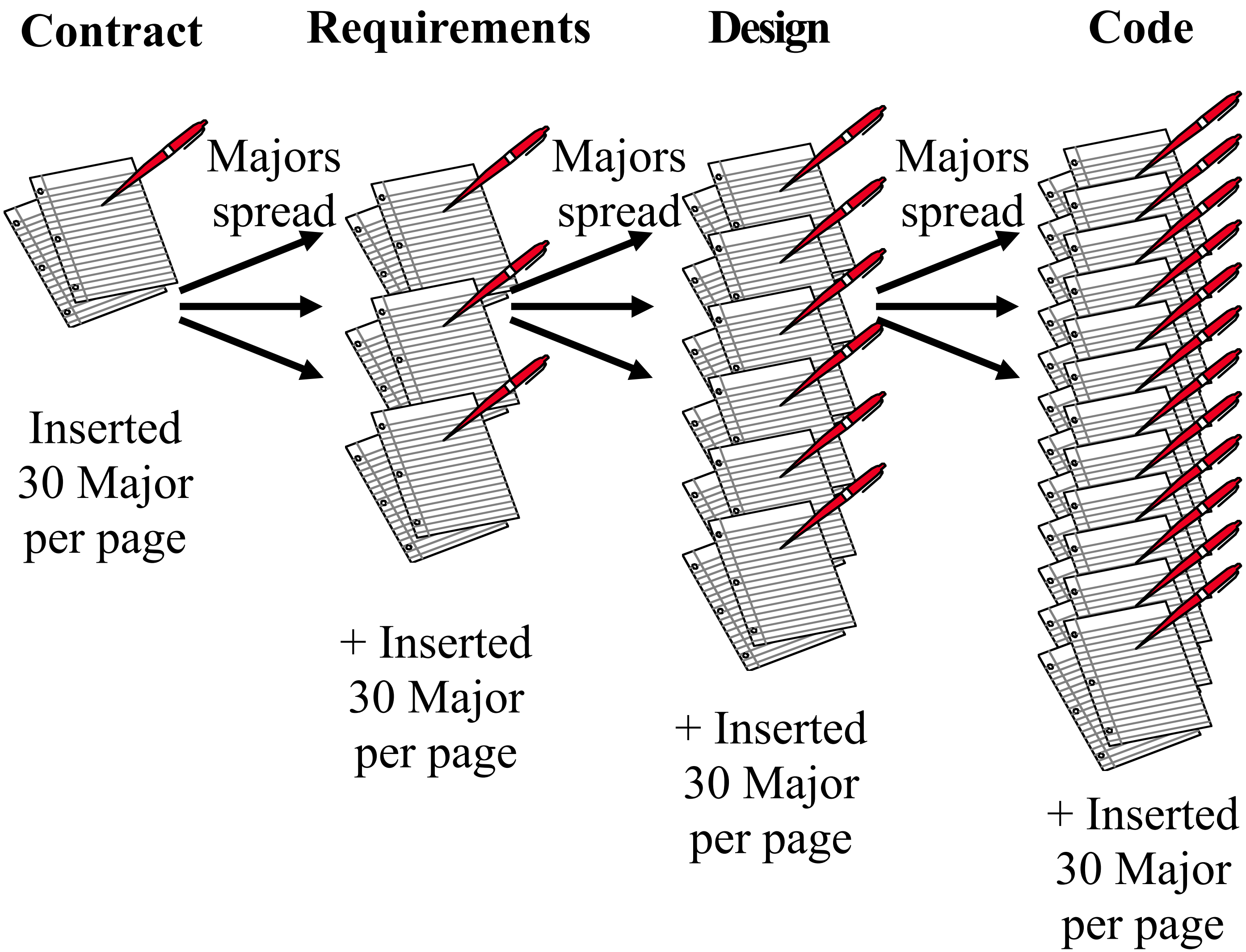
Scale: Estimated remaining, Major Threats, per logical page (300 Non Commentary words)

Past [March this year, Req] **70-140** Majors/Page. <- Multiple sample inspections

Goal [May next year] less than **<10-30>** Majors/Page.

Goal [Dec. next year] **optimum exit**

Software Development without Exit



Contract

Majors spread

Inserted 30 Major per page

Requirements

Majors spread

+ Inserted 30 Major per page

Design

Majors spread

+ Inserted 30 Major per page

Code

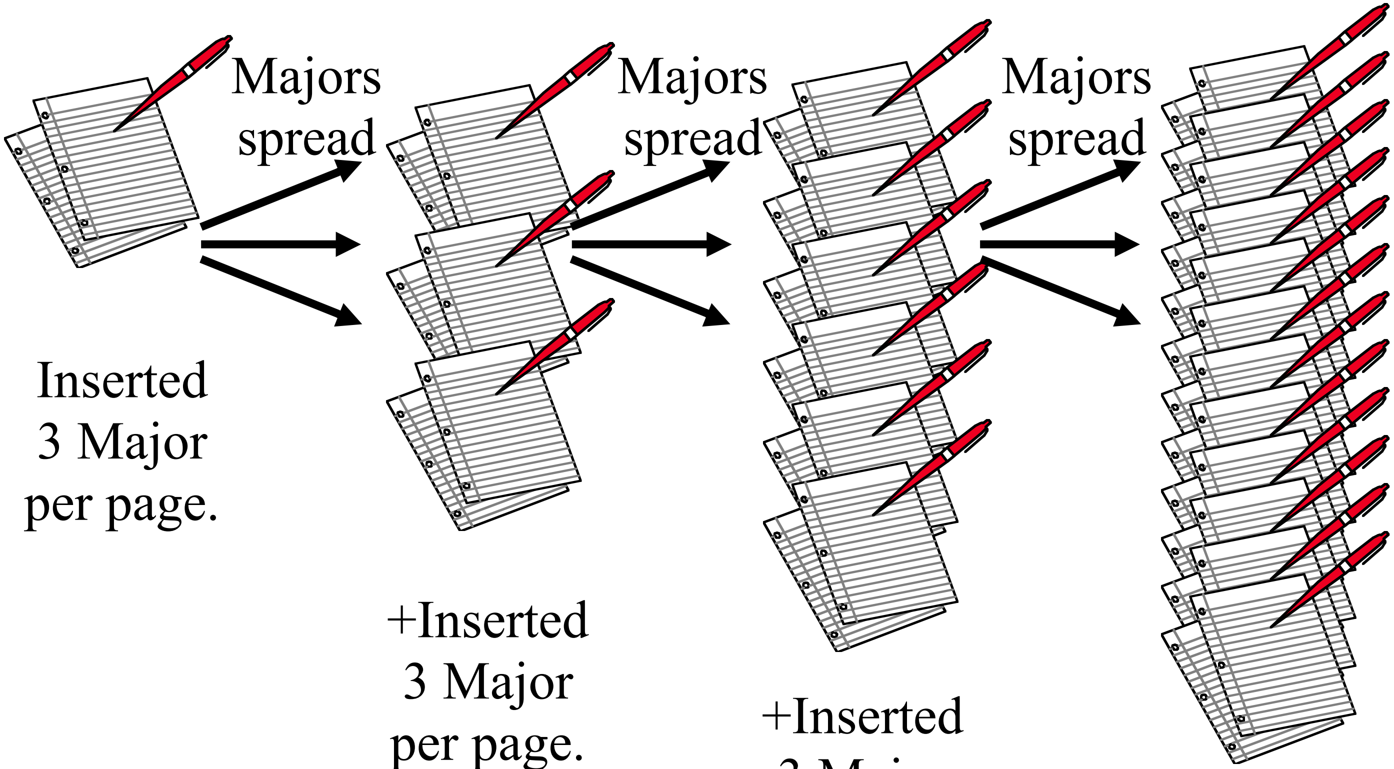
+ Inserted 30 Major per page

Test
Finds enormous number of Threats .
High Cost in time, people and money

Operation
Unhappy Customers

Maintenance
High Cost
(Much Higher than development costs)

Development with Lean Inspection & numeric Exit



Contract

Inserted 3 Major per page.

Majors spread

Requirements

+Inserted 3 Major per page.

Majors spread

Design

+Inserted 3 Major per page.

Majors spread

Code

+Inserted 3 Major per page.

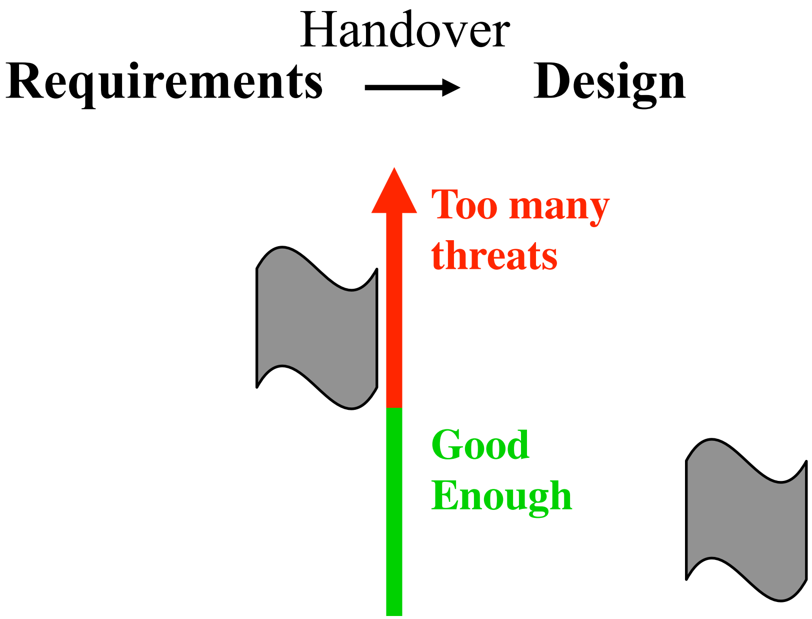
Test
Verifies low number of bugs.
Low Cost in time, people and money

Operation
Happy Customers

Maintenance
Low Cost

How does Lean Inspection ensure that a low number of threats are released to next levels?

Answer: Entry and Exit conditions



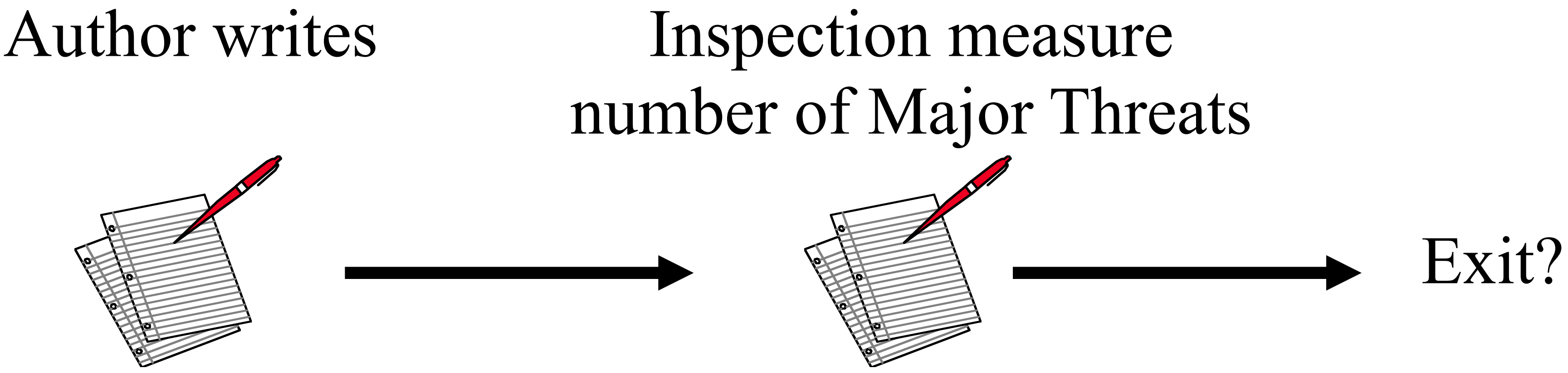
Handover

Requirements → **Design**

Too many threats

Good Enough

Lean Inspection is used to measure the number of threats per page

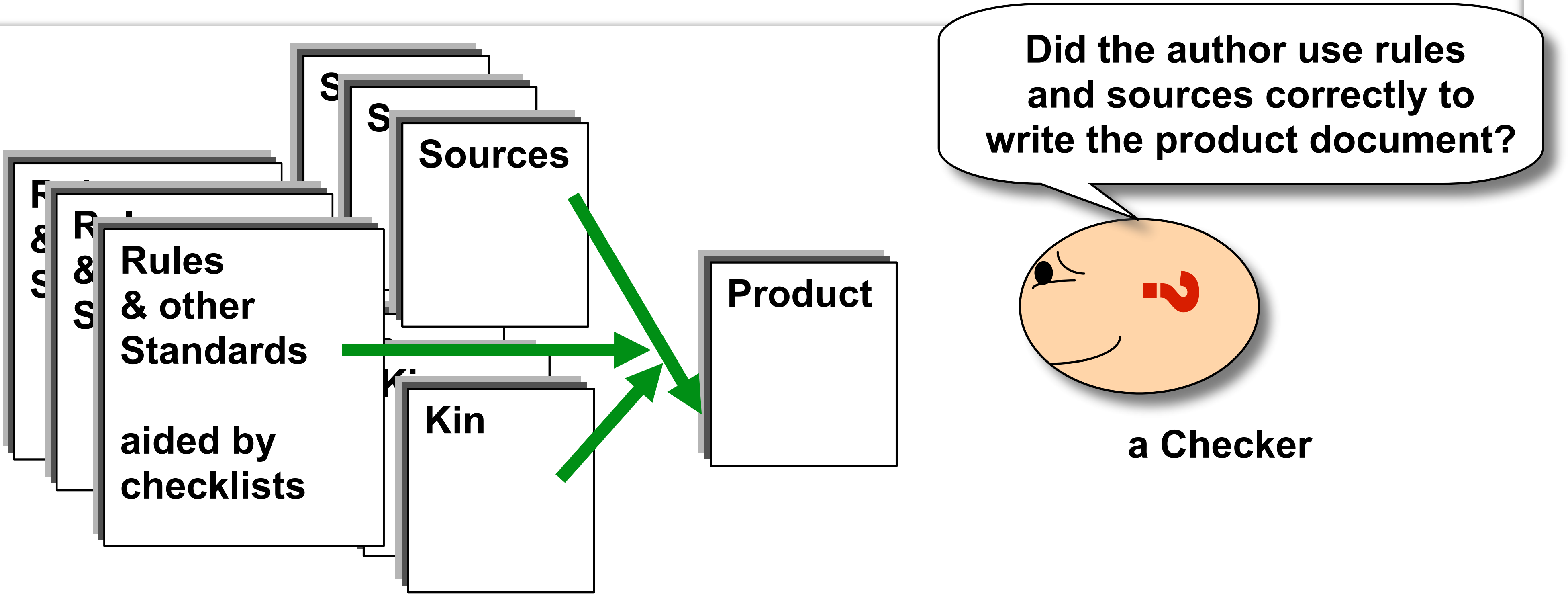


Author writes

Inspection measure number of Major Threats

Exit?

How does Inspection find a Threat?


Sources
Rules
& other
Standards
aided by
checklists
Kin
Product
Did the author use rules and sources correctly to write the product document?
a Checker

Source: Example, a policy "Avoid complexity for users".

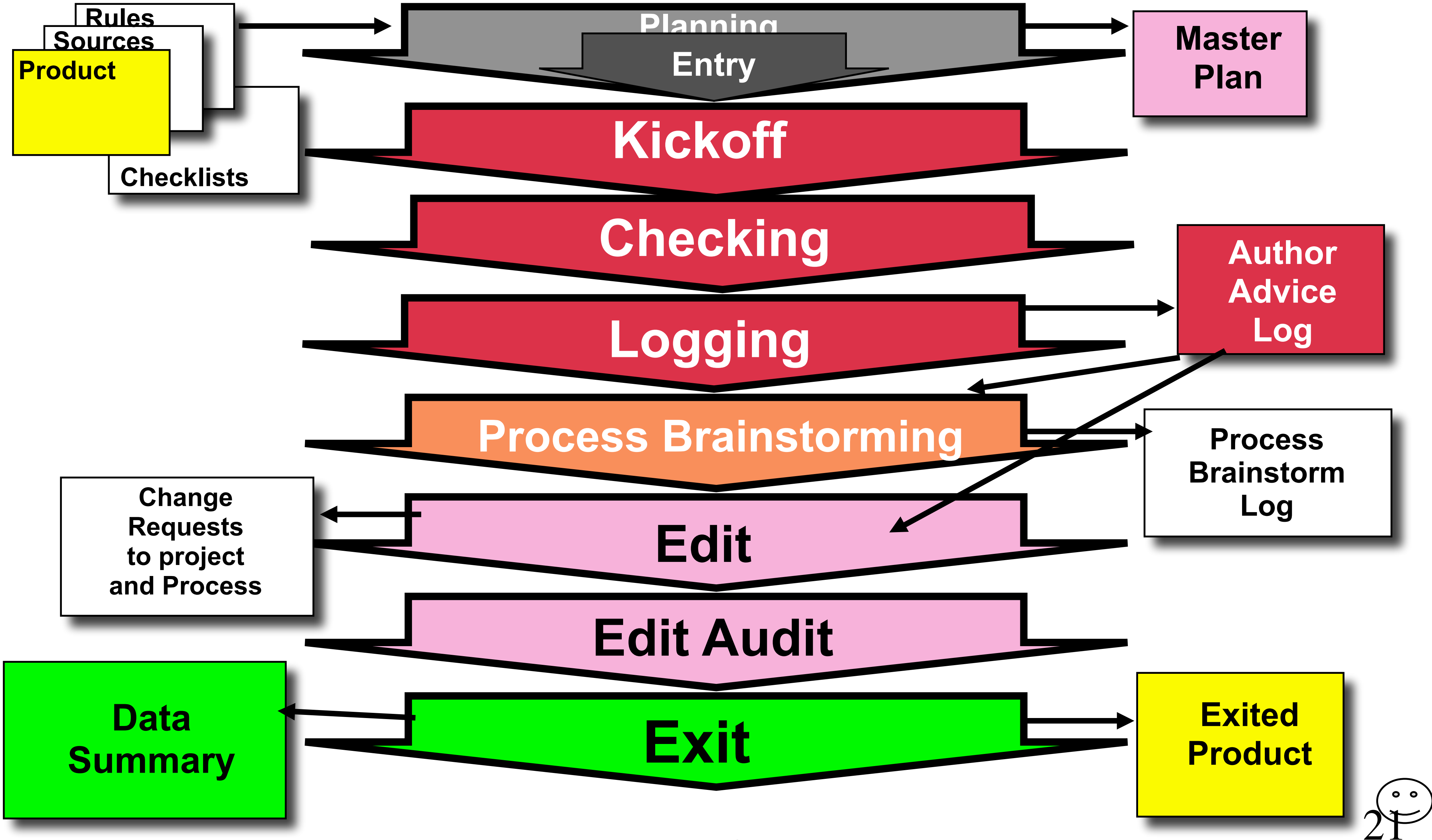
a Rule example for Requirements; Rule 1: "All Quality Requirements shall be specified numerically".

A statement in Requirement Product document: "It shall be state-of-the-art easy for the customer to use".

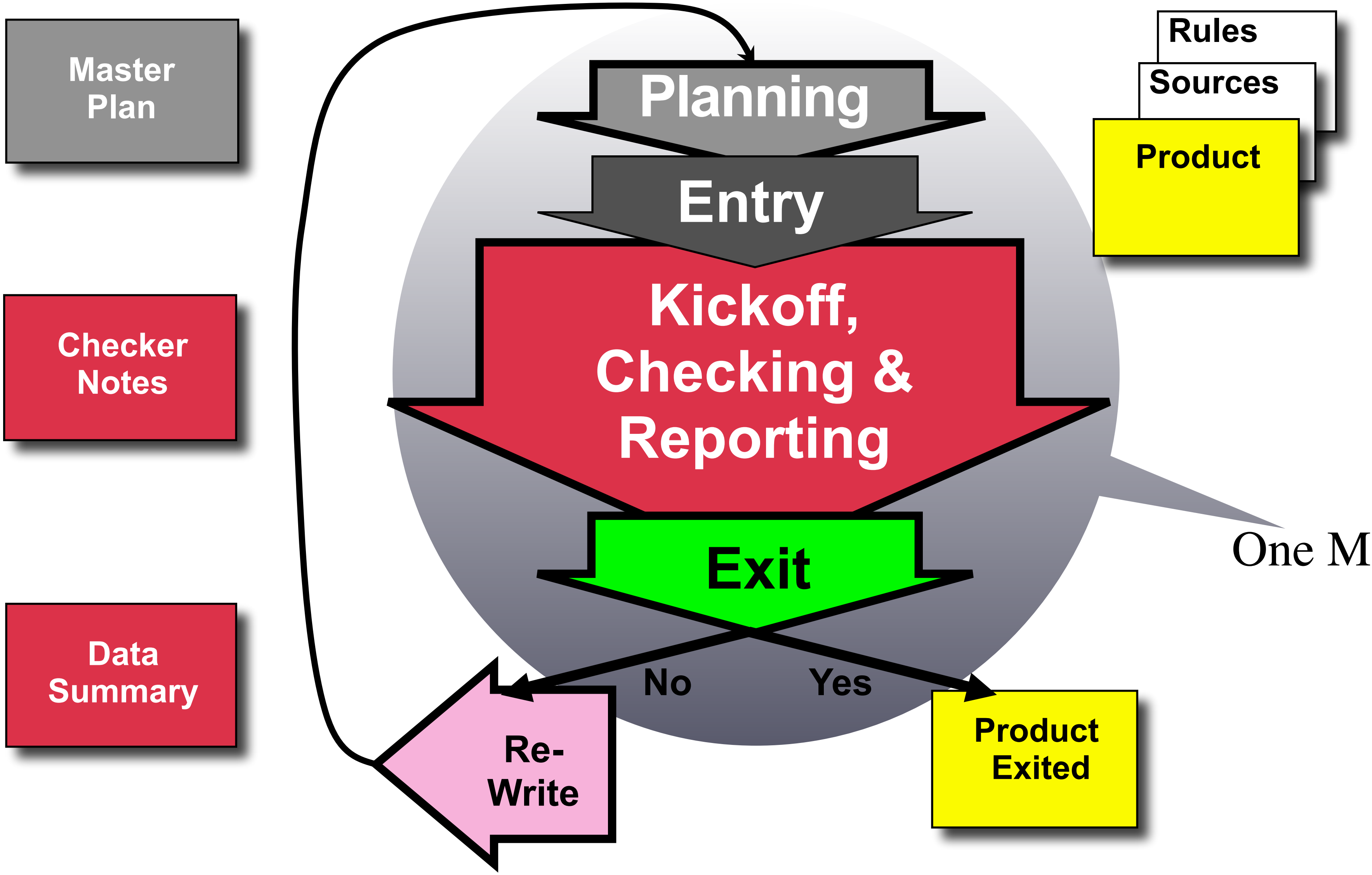
Checkers asks them self: Is Rule 1 followed? (if not, you found a "Threat")

"Kin" documents (like test plans) can also be used to check correctness.

Full blown Inspection Process Map



Lean Inspection Process Map



One Meeting

Rule

Rule

A best practice specification standard.

Notes

Rules are set by engineering process ‘owners’

Rule violation = ‘Threat’

Rules should focus on ‘Major’ (severity) practices.

Rule:
All quality requirements must be expressed quantitatively.

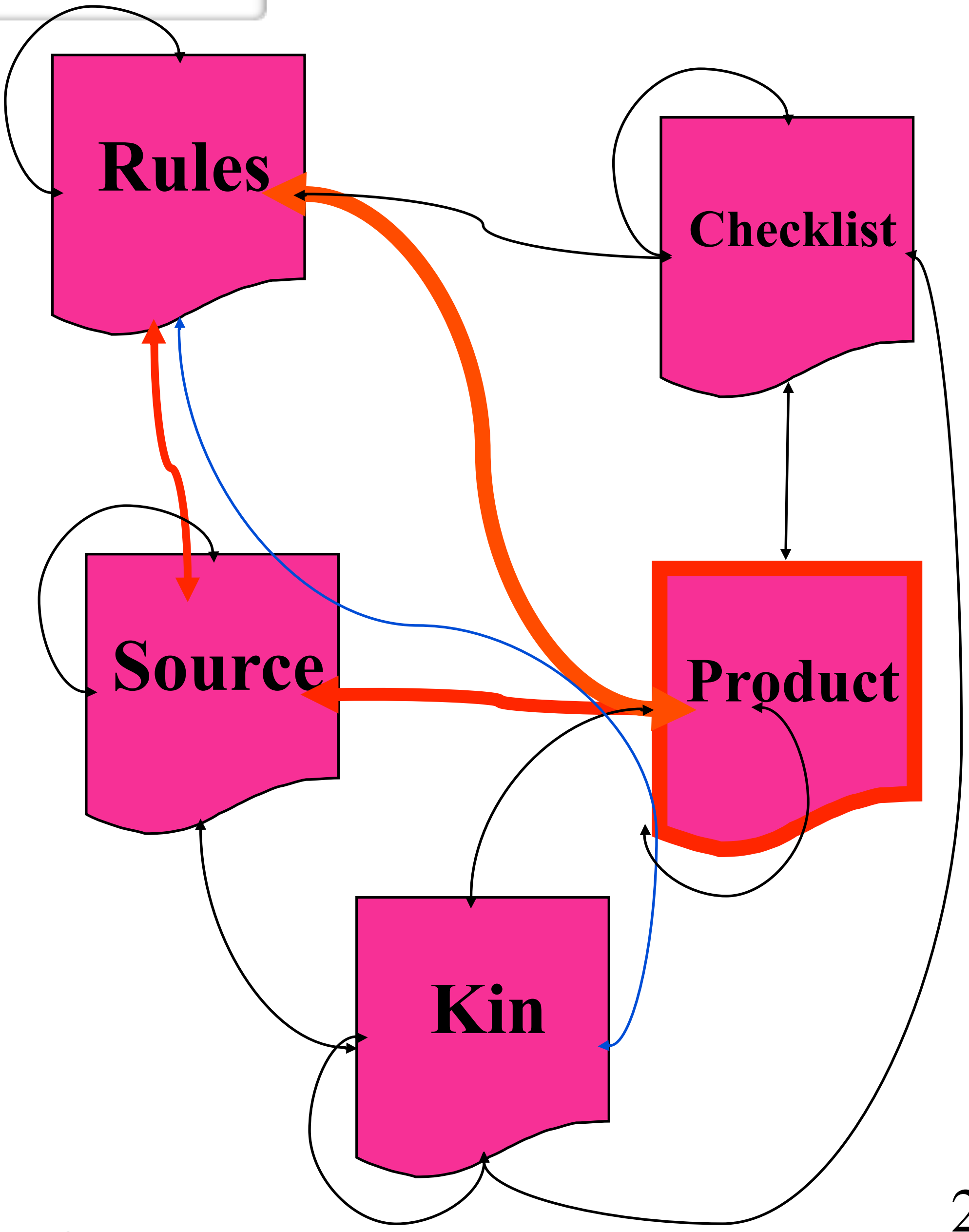
Rule ≠ Specification

Requirements:

1. The system will be extremely User-Friendly.

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



How do you check?

The “Burger King” ™ Principle

(Any way you like it)

Check all relevant **rules** are followed

“Rules” are ‘checklists’ during checking

note “Improvement suggestions” for rules

if rules are unclear

if you have a better or new rule idea Using

optimum rate
(assigned, or adjusted
by you personally)

Check against related source/kin documents

according to the rules and checklists

Check internal consistency

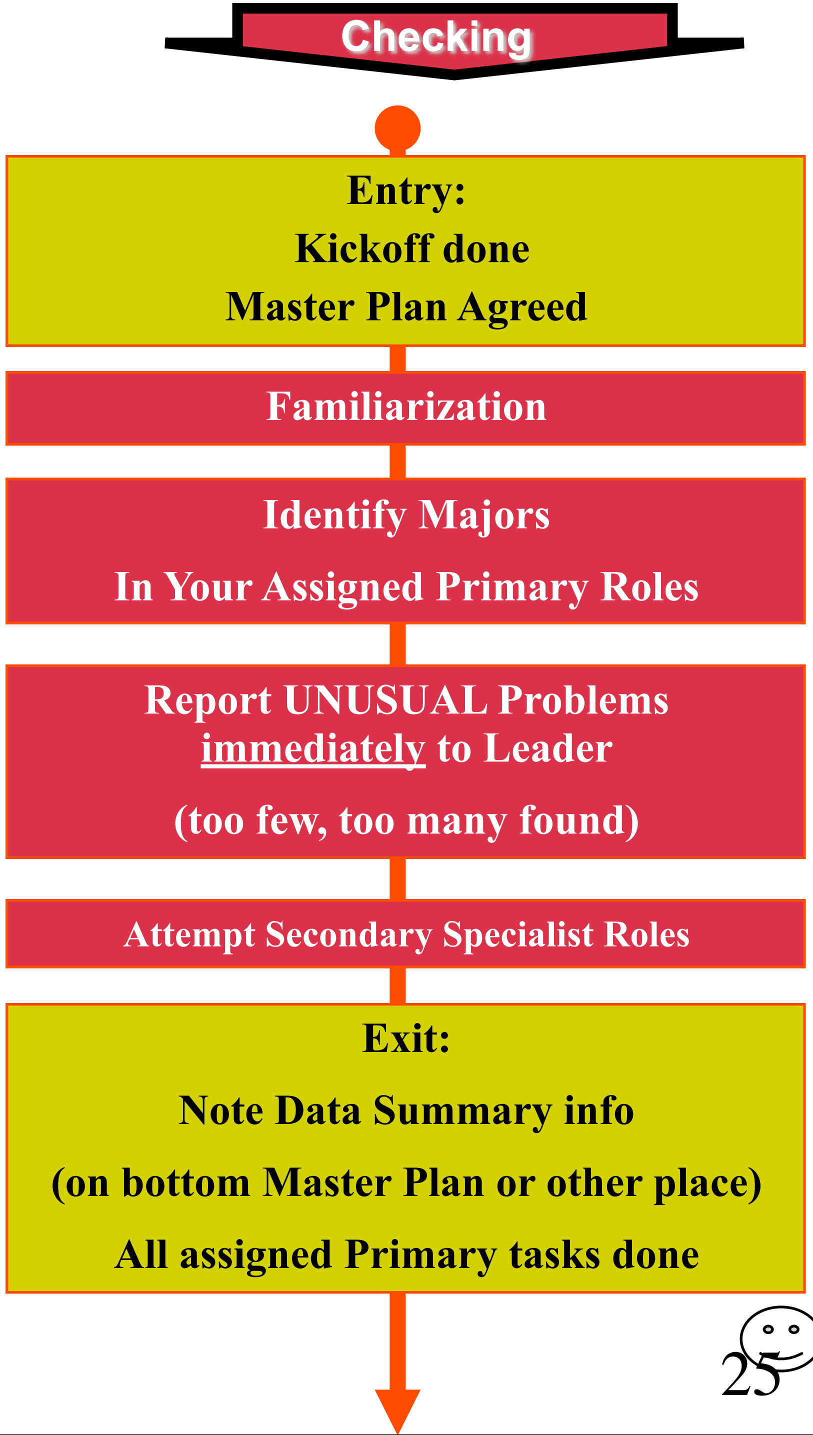
Full blown Inspections

Check against formal checklist questions

Improvements can be noted for checklists

Note, all checklist questions support a “rule”

You give priority to your
assigned **roles** (where to
check, what to seek)



Checking

Entry:
Kickoff done
Master Plan Agreed

Familiarization

Identify Majors
In Your Assigned Primary Roles

Report UNUSUAL Problems
immediately to Leader
(too few, too many found)

Attempt Secondary Specialist Roles

Exit:
Note Data Summary info
(on bottom Master Plan or other place)
All assigned Primary tasks done

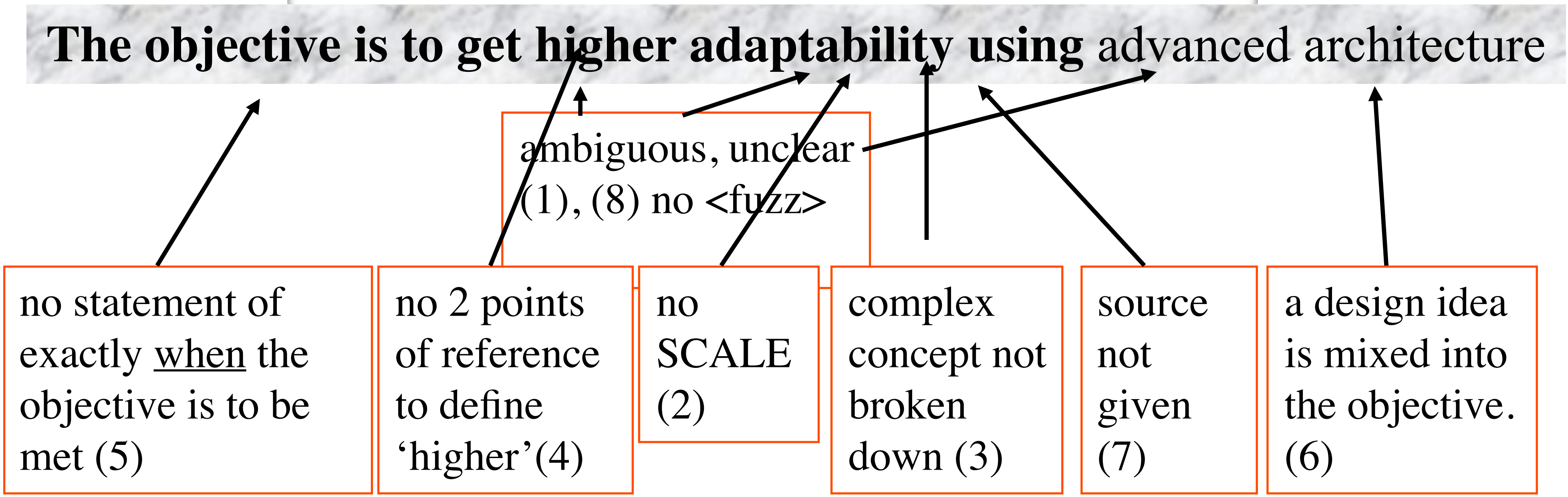
See any ‘Threats’ with this?

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

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 statement of exactly when the objective is to be met (5)

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 few ‘Threats’

Adaptability:

Maintenance:

Scale: Clock time to fix a bug and <validate> fix.

Past [Product X, last year] 5 hours <- Internal stats.

Goal [Product Y, At <Launch>] 10 minutes <- Mkt. Dir.

Portability:<- Marketing Plan Dec 15th. M.P.

Scale: Conversion cost for [defined ports].

Past [Prod. X, Any UNIX, this year] 100 hours/1000 Lines

Goal [Prod. Y, Any UNIX, next year] 20 hours/1000 Lines

The Power Set!
Generic Editing Rules

P1 (CON) Statements must not contradict any statement in the same or related documents, or any information in sources.

P2 (CLEAR) All statements must be crystal clear to the intended readership.

P3 (ENO) Documents must contain enough information to fulfill their purpose.

P4 (BRIEF) Documents must be as brief as possible.

P5 (TRGET) Documents must contain text or a reference that describes their intended readership and purpose.

Generic Editing Rules (Rule.G)

Planning

G1 (INCON) Statements must not be inconsistent with any statement in same or related documents, unless motivated by a statement.

G2 (ONE) All documents shall be crystal clear to the intended readership.

G3 (NOTE) Text or figures not intended for decisionmaking (e.g. comment, note, suggestion, example) shall be marked as such.

G4 (EXTRA) All documents shall be as brief as possible.

G6 (HEAD) Document headers shall follow current CompanyXX standard.

G7 (UNIQ) A statement shall exist on one place only in a document set.

G8 (SOURC) All statements shall contain information about their exact sources.

G9 (EL) All statements shall be broken down to their most elementary form.

G10 (TAG) All statements shall have a tag according to 11/00021-1/FEA 202 502.

G12 (CHANG) Changes in specific statements, from the previous approved version of the document, shall be stated in same or referred document, Example: in a "Revision History" chapter.

G13 (RISK) Any known or suspected uncertainty or risk shall be stated, with syntax: {Unsure: 0.6-0.7}, scale: 0.0=a random guess, to 1.0=a proven fact.

G14 (STATU) Documents shall show their DQI status, according to the Generic Exit Conditions, 2/170 01-1/FEA 202 502.

G16 (ENOUG) Documents shall contain enough information to fulfil their purpose.

G17 (FUZZY) Unsure information shall be marked e.g. with angle brackets, example: <the project has stable requirements>.

G18 (NOTAT) Specific notation used [see rule G3, G13, G17] shall be clearly explained.

G19 (READR) A document shall in text or by reference state its intended readership.

G20 (PURPO) A document shall in text or by reference state its purpose and application.

Generic Rules for On-Screen Viewing

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].

Rules
SI pp 424-5

Rules For Source Code: Correctness

COR.00 (SIGNIFICANCE). Violation of any correctness property means that the software may not behave as intended.

COR.01 (INCOMPUTABLE). All computations (including assignment) must follow the laws of arithmetic and be within limits defined by the program and the implementation platform (language, compiler, processor …).

COR.02 (INCOMPLETE). All elements of each code segment must be defined and implemented so that the code segment can fulfil its intended functionality.

COR.03 (UNASSIGNED). Each variable must receive a legitimate value (defined within the *domain* of the variable) before being used in assignment, computation, or comparison.

COR.04 (IMPRECISE). Each variable must be defined to a precision adequate for all values received through initialisation, assignment, or computation.

COR.05 (UNINITIALISED). Each variable used as a loop variable must be explicitly initialised prior to loop entry, and as late as possible prior to loop entry. A loop structure is initialised if all loop variables are initialised.

COR.06 (NON-PROGRESSIVE). Each branch of a recursive function, or iteration of a loop, must make clear progress towards the termination condition for the recursive or iterative process.

COR.07 (NO VARIANT). For each loop structure, there must be a *variant* loop guard that defines a relation (the variant condition) that is congruent with, and derivable from, the variant function used to prove termination of the loop.

COR.08 (INCONSISTENT). All uses of the code structure must maintain its properties and functionality, all its elements must contribute to its overall effect, and it must generate no side effects and not be misused.

Rules For Source Code: Structure

STR.00 (SIGNIFICANCE). Structural properties enforce structured programming rules. Main focus is within a module, but some properties also apply at higher levels of organisation.

STR.01 (UNSTRUCTURED). Each code segment must correspond to one of the three block types (iteration, selection, sequence), and must have a single clearly identifiable point of entry and single clearly identifiable exit point.

STR.02 (UNRESOLVED). The code control structure must be derived from the problem data structure or logic structure.

STR.03 (UNHOMOGENEOUS). Compound termination conditions for loops must use only AND connectives.

STR.04 (INEFFECTIVE). The code document must include all necessary elements and no unnecessary elements to define and implement its specified function.

STR.05 (INCOMPLETE). The code must include all conditions necessary for its function.

STR.06 (REDUNDANT). The code must include **no** conditions not necessary for its function.

STR.07 (COMPOUND). All computations must be expressed in their simplest form.

STR.08 (INDIRECT). Each computation must be represented in a way that derives directly from the problem it addresses.

STR.09 (UNADJUSTABLE). All constants used within the code must be declared by name, with the exception of 0, 1, and -1; no more variables and constants may be declared than necessary to execute the code function, and each must have a single purpose.

STR.10 (RANGE-DEPENDENT). The lower and upper bounds of each variable (especially arrays) must be expressed with variables rather than with constants.

STR.11 (UNUTILISED). Every defined code object (data structure, variable, constant, code block …) must be used within the scope it is defined in; no object may be defined but unreferenced.

STR.12 (UNINDENTED). Statements within a code segment must follow a consistent pattern of indentation.

STR.13 (EXTERNAL). Code must not be written to replicate functionality of external reusable components or library procedures.

STR.14 (SEQUENCE). Code segments must be executed in correct sequence.

STR.15 (TIMING). Code must be capable of meeting specified execution time constraints.

STR.16 (COMPLEX). No code procedure may have an extended cyclomatic complexity exceeding 15 except where there are documented adequate reasons.

Rules For Source Code: Modularity

MOD.00 (SIGNIFICANCE). Modularity properties address high-level module design issues and system interface issues, including how a module encapsulates its data; how it is coupled to other modules; how loose its internal structure and functionality are; how flexible it is; and its potential for re-use.

MOD.01 (UNPARAMETERISED). All inputs must be accessed via a declared parameter list which defines only the necessary and sufficient inputs and outputs for the module's function.

MOD.02 (COUPLED). The module must have no connection to calling modules except via its name and declared parameters.

MOD.03 (GLOBAL). All data used by the module is local to (declared within) the module.

MOD.04 (INCOHERENT). All internal elements of the module must contribute to achieving a single, well-defined, objective or function, and the sequence of statements must be such that the last in a sequence depends on all its predecessors (none could be executed in parallel).

MOD.05 (TYPE-DEPENDENT). The module's computations must not be dependent on the types of its inputs and outputs (i.e., input and output types must be parameterised).

MOD.06 (NON-ABSTRACT). There must be no obvious, useful, more generic concept that includes the code structure.

Rules For Source Code: Description

DES.00 (SIGNIFICANCE). Descriptive properties reflect how well source code is internally described. *Comments* document how a program realises its desired functionality; *preconditions and postconditions* specify the functionality of a code segment; *self-descriptive identifiers* contribute to the ability of readers to analyse code.

DES.01 (UNSPECIFIED). The functionality of each code structure must be described by precondition and postcondition specifications describing its initial state, inputs, outputs, function, eventual state, and side-effects (if any).

DES.02 (UNDOCUMENTED). The purpose, strategy, intent, and properties of the code structure must all be explicitly and precisely described within its context.

DES.03 (NON-DESCRIPTIVE). The name of the code structure, and the identifier(s) used within it, must be clearly consistent with its purpose, strategy, intent, or properties, as well as being descriptive of their own.

DES.04 (INCONSISTENT). All descriptive information must be consistent with the code it describes, and vice versa.

DES.05 (UNCOMMENTED). All descriptive information must be clearly identified as such so that readers cannot confuse it with executable code.

Workshop

Hold Lean Inspection

Team up 2 to 3 people on each team.

Plan

- Product Doc - Mother doc. - Rules
- Decide on a numeric Exit criteria.
- Decide on Sample.
  - Review Plan with Tom & Kai

Kickoff - Checking - Reporting - Exit

Review numeric results with class

- Estimated Major per Page
- Actual Checking Rate
- Total Cost of Inspection
- Exit yes or no
- Any other insights/comments

Optimum Checking Rate

Optimum Checking Rate

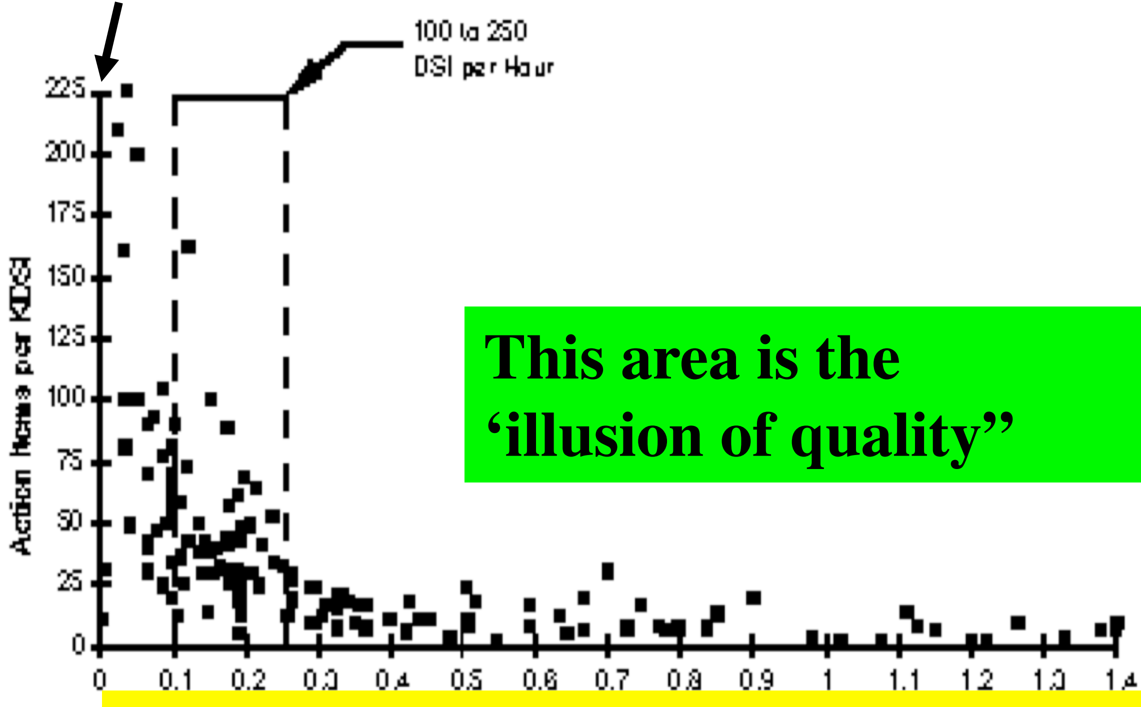
The most Effective individual speed for ‘checking a document against all related documents’.

Notes

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

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

Action Items/1,000 Source lines



This area is the ‘illusion of quality”

Thousands of Statements Checked per hour by a person

Exit Process
(at end of Inspection)

Exit

Purpose:

Determine if Product document is economic to release

Method: Exit Conditions

Is the Inspection ‘Trustworthy’?

Is the Product document ‘OK’?

- Remaining Majors =?
- Economic to Exit?
  - Or cost more later
  - Than fix now?

Exit Conditions

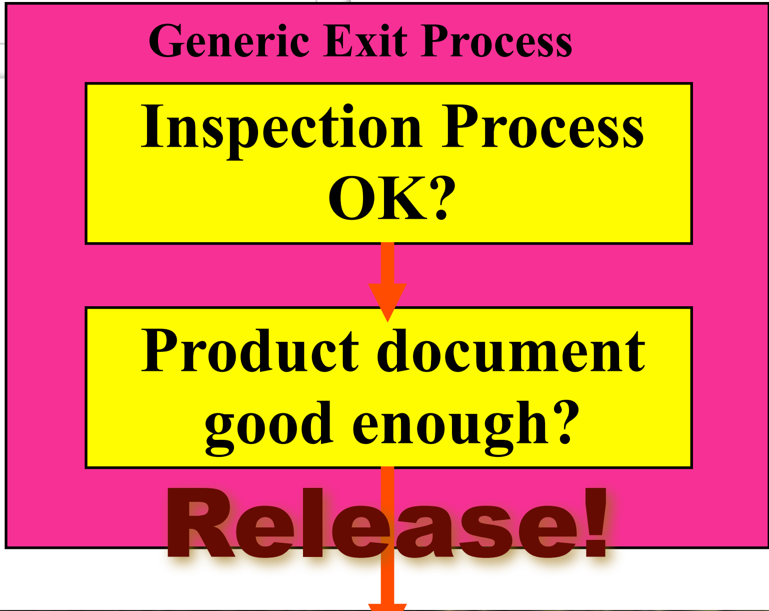
Definition

A formally written condition which must be met for a task to be successfully completed.

Notes:

- **Part of a process definition**
- **'owned' by process owner**
- **Learning from hard experience**
- **Based on 'economics'**

Generic Exit Conditions: ‘Release’ (SI p 202)



Generic Exit Process

Inspection Process OK?

Product document good enough?

Release!

Process OK?

- Optimum Check Rate?
- OK with Leader?
- Experience ->Database?

Product OK?

- Few Majors remain?
- OK with Author?
- Known Majors fixed? (not for Lean Inspection)

Exit Conditions

EI Exit Conditions

EI.X1: Defect Density Condition:

Estimated Major Defects remaining per page is less than 1 per 300 Non commentary words *(initially until end 2003 10 Majors, to get a lenient start)*.

FORMULA FOR ESTIMATION:

Assume 33% effectiveness of the 2-checker checking-process.

Total Unique Majors acknowledged by writer, found in the sample logical page, times 3, gives a reasonable estimate of Majors/Page. This is before writer correction of known Majors.

Note: the effectiveness for a 3 checker group is slightly higher say about 40%. This figure needs to be determined by your own measurement.

OPTION: we might manage the exit level at an individual writer level to gradually motivate them to improve by about 50% (defect injection) less per iteration of the write and check cycle. <- KM idea – TG likes it!

EI.X2: Writer Veto

The specification cannot exit if the spec writer wants more time to improve it.

Generic Exit Conditions

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.

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 threats (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!

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.

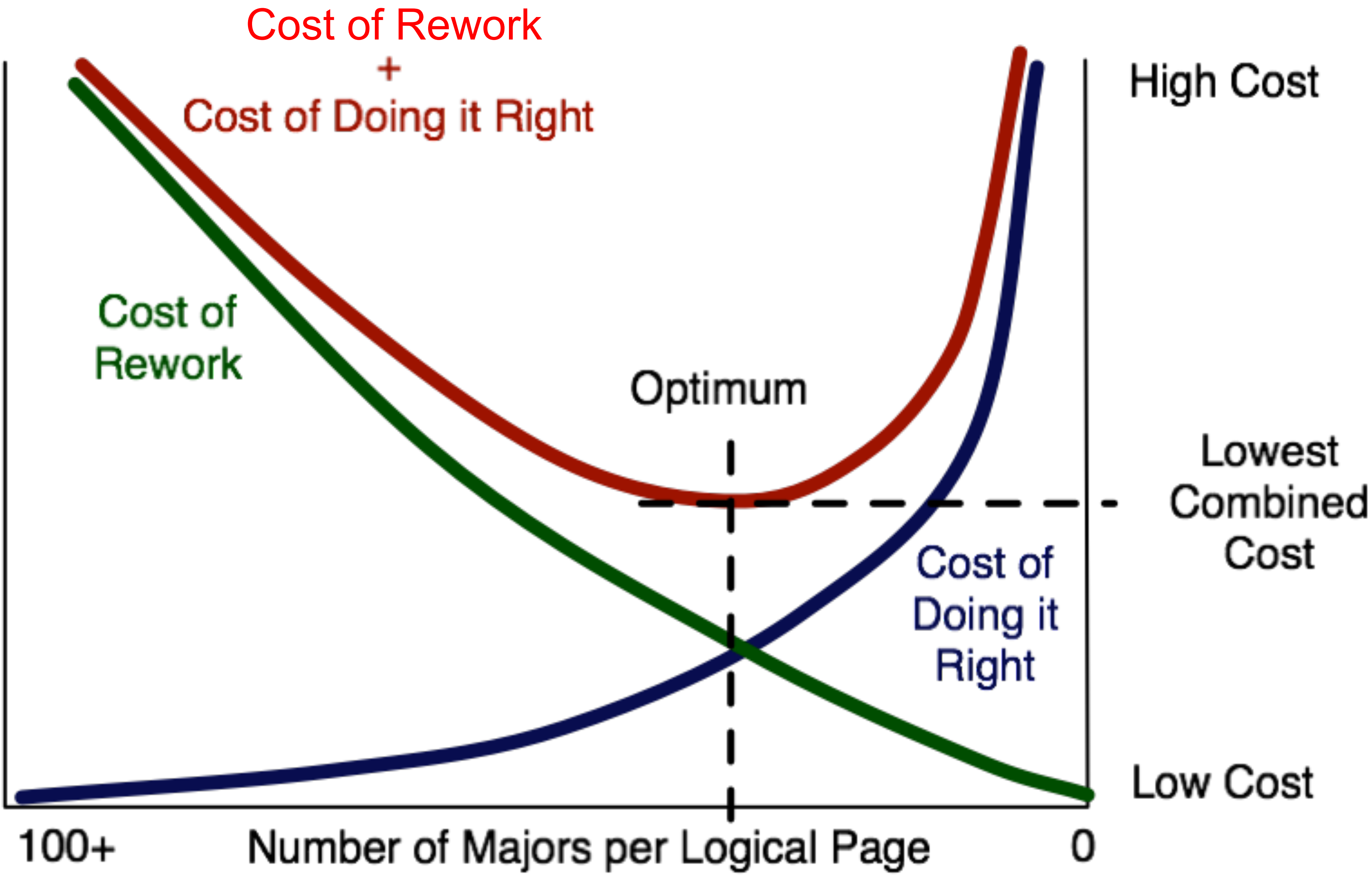
GXI.5 (Author OK).

The Responsible Product Document Author/ Editor approves Exit.

GXI.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.


Cost of Rework
+
Cost of Doing it Right
Cost of Rework
Optimum
High Cost
Lowest Combined Cost
Cost of Doing it Right
Low Cost
100+
Number of Majors per Logical Page
0

Workshop

Hold Lean Inspection

- Team up 2 to 3 people on each team.
- Plan
  - Product Doc - Mother doc. - Rules
  - Decide on a numeric Exit criteria.
  - Decide on Sample.
    - Review Plan with Tom or Kai

- Kickoff - Checking - Reporting - Exit
- Review numeric results with class
  - Estimated Major per Page
  - Actual Checking Rate
  - Total Cost of Inspection
  - Exit yes or no
  - Any other insights/comments

Generic Entry Conditions

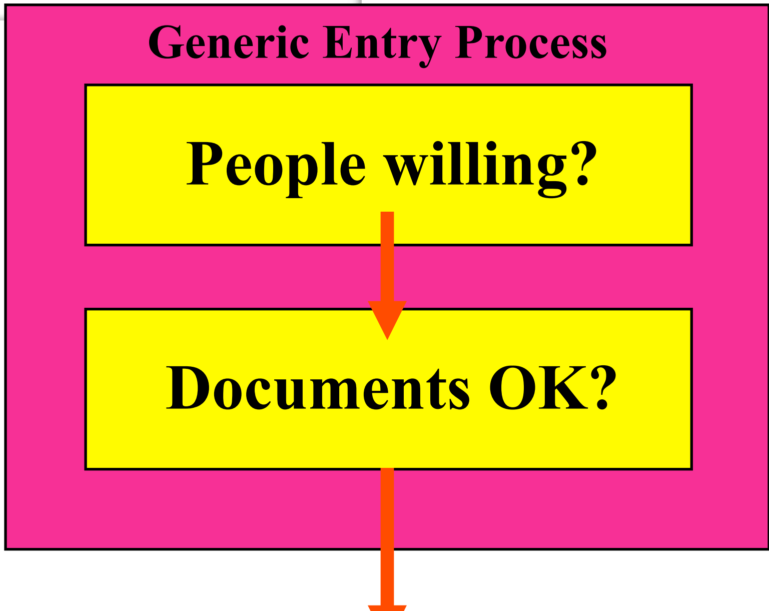
(SI p 64-65)

People-related Entry Conditions

- Author veto
- Leader veto
- Leader trained & certified
- Author/Editor = checker

Quality-related Entry Conditions

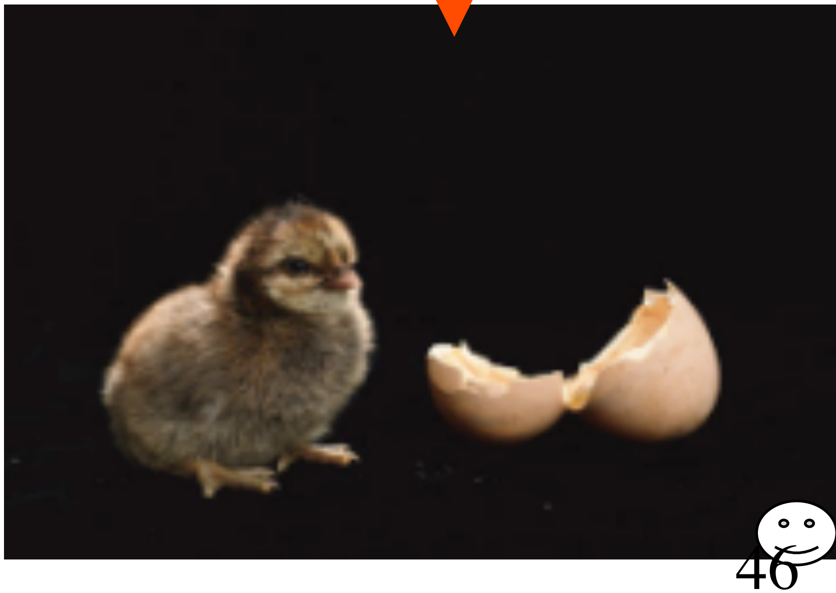
- Sources exited
- Rules available
- Master Plan available
- Product=Cursory OK
- Clean diagnostic/spell



Generic Entry Process

People willing?

Documents OK?



Entry Conditions

EI Entry Conditions

EI.E1: Experience. At least one of the participants has done a well conducted successful inspection once before, or been briefed by a competent practitioner, or will be guided through the process by a competent guide (ideally an expert in this process).
Rationale: people need to have some reasonable sense of how to do this process, otherwise it can become corrupted. We believe we can avoid formal training in the method, but we need some knowledge and experience of it in place.

EI.E2: Author. The specification writer sincerely believes that the defect level is low enough to exit. They have done personal checking against the rules themselves and find no defects.
Rationale: the writer should take the trouble to make sure the spec id as clean as possible before inspections. They should not misuse people and time to compensate for sloppy work.

EI.E3: Source. Exited copies of all source specifications are available.
Rationale: there is little point in checking consistency against highly polluted source specifications.
(example by using bad Business Requirements to check new System Requirements).

EI.E4: Toolkit. An updated 'Inspection Toolkit' (with specification Rules, Checklists (for learning to apply the rules in practice), Process descriptions, forms, electronic support, intended readership role information) is available and is understood by the participants.
Rationale: This tool kit is the real definition of the Inspection process. This really determines correct use of the method.

Generic Entry Conditions Page 1 of 4

GEI.1 (Author Checks)

The Product Author (or proposed Editor; essentially the current owner of the Product) agrees to participate fully as a Checker on the Inspection Team. Someone MUST be assigned this role.

GEI.2 (General Veto Power)

Both the Team Leader and the Product Author must agree to decide that we will enter the Inspection process, and later to continue the Inspection process beyond initial Entry to Planning.

> *Note: This does not excuse the Author from the ultimate responsibility of successfully Exiting their Product document sometime, as required by Exit Conditions.*

Generic Entry Conditions. Page 2 of 4

GEI.3 (Sources Exited)

All Source documents for the ’Product document to be Inspected’, shall themselves have Exited, with known Threat levels, from their own Inspection.

This means Sources shall have met all valid Generic and Specific Exit Conditions pertaining to them.

Conscious and Documented Failures to meet the Source’s Exit Conditions *can* be accepted, when the Team Leader is able to make provision for the exceptions. Example by sampling them, or special roles to Check them.

GEI.4 (Formal Rules)

There must be one or more sets of Formal Rules for writing the Product Document, which the Author and Leader agree apply to the Quality Control Process of Inspection.

Note: such Rules are the only valid Conditions for identifying a Defect in the Product document, or in other associated documents, such as Sources.

Generic Entry Conditions. Page 3 of 4

GEI.5 (Master Plan)

A written Master Plan for the Inspection has been faithfully completed, and is reasonable in all critical respects, so as to ensure 'probable success' of the Inspection effort expended, with regard to the stated Inspection Purposes.

Note: this includes people, timing, documents, optimum rates, special roles.

GEI.6 (Team Leader Capability)

The responsible team leader has been thoroughly trained in the correct practice of Inspection, and has got current Certification attesting to that fact.

Note: if not, a Certified Leader can take formal responsibility for a 'Co-Leader' when they are being trained or evaluated for Certification through practical work.

Generic Entry Conditions. Page 4 of 4

GEI.7 (Cursory Check)

The Leader, or competent expert they assign, shall do cursory checking, of at least 15 minutes duration, and of 100 Non-commentary words, of the Product document, against a selection of applicable Rules, searching for Major Threats .

If the quantity of Major Threats per Logical Page exceeds the Maximum remaining Majors at Exit, then the situation will be discussed with the Author, with a view to correcting the Product document before proceeding to use Team time. Use an expert if you can’t do it!

GEI.8 (Automatic Pre-Clearance)

The Author shall demonstratably have applied all available and valid automated checking tools prior to submission, and shall have cleaned up any diagnostics.

Note: this means Spell Checkers, Source Code Diagnostic Compiles, ”lint” (depth diagnostic for C).

How big a sample?

Purpose

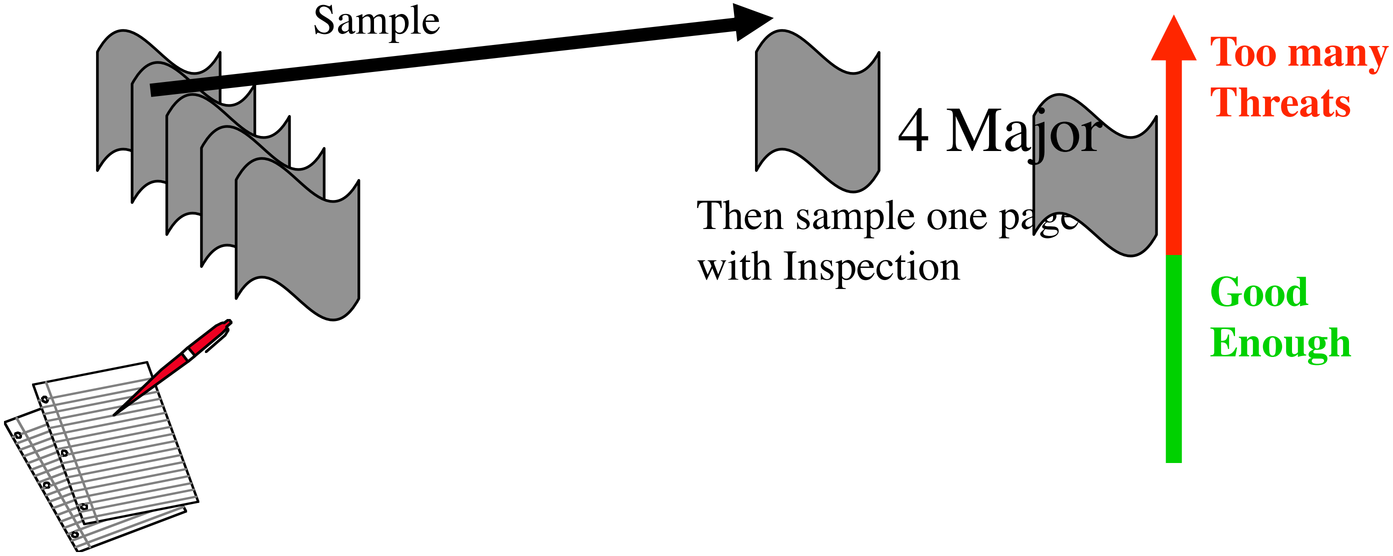
To get some accurate idea of Major Threat density at low cost

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

How to Inspect large amount of documentation!

Sample “During” Authoring 1



Sample

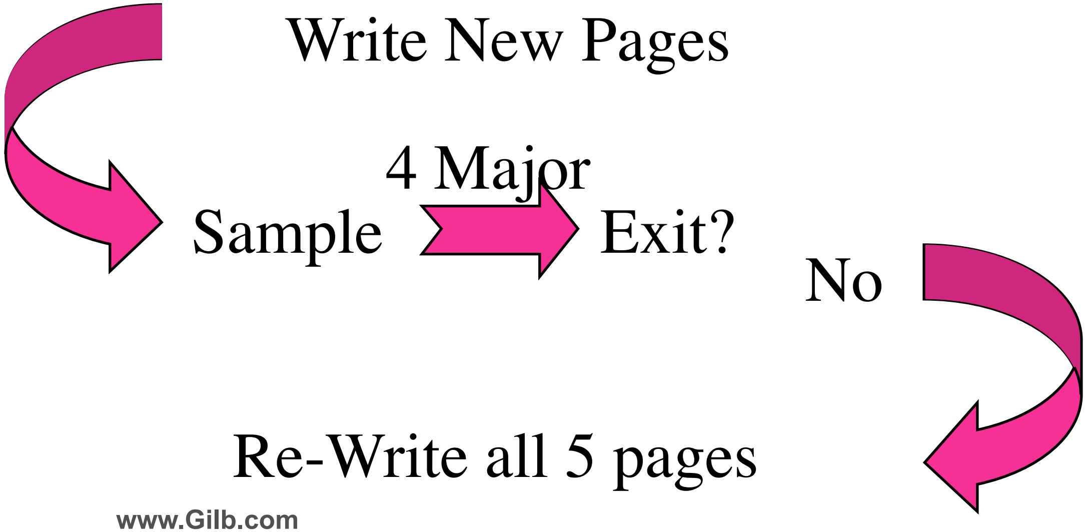
4 Major

Then sample one page with Inspection

Too many Threats

Good Enough

The Author is expected to write about 45 pages



Write New Pages

4 Major

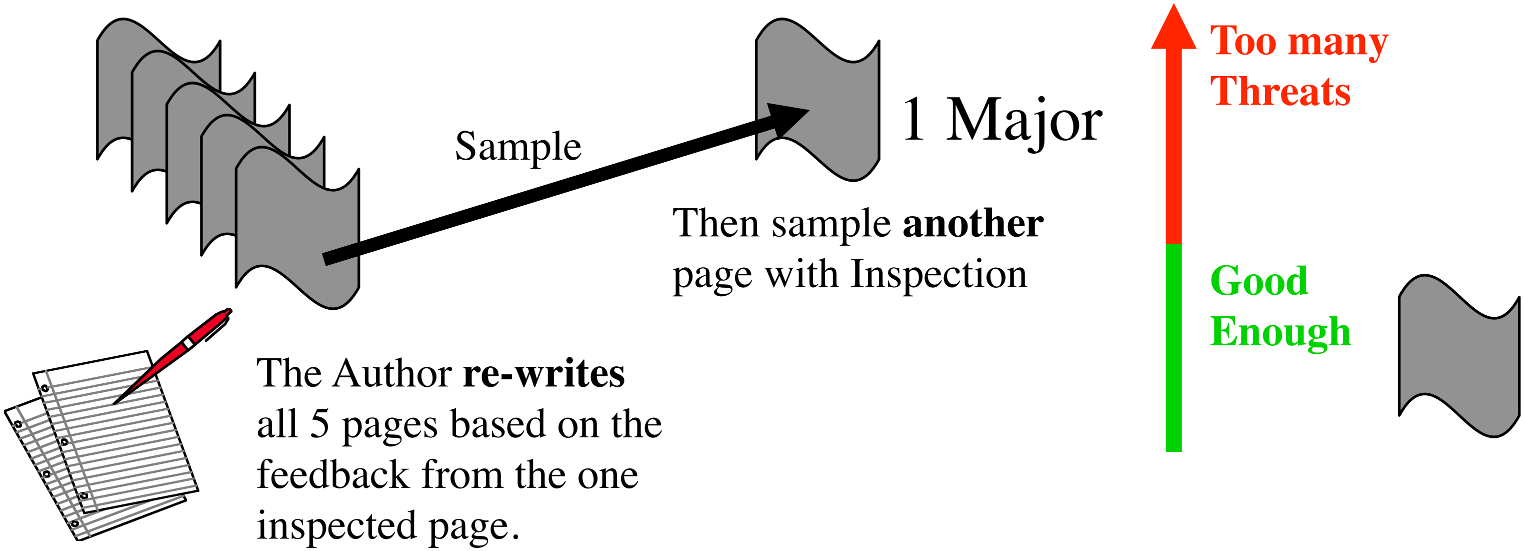
Sample

Exit?

No

Re-Write all 5 pages

Sample “During” Authoring 2



Sample

1 Major

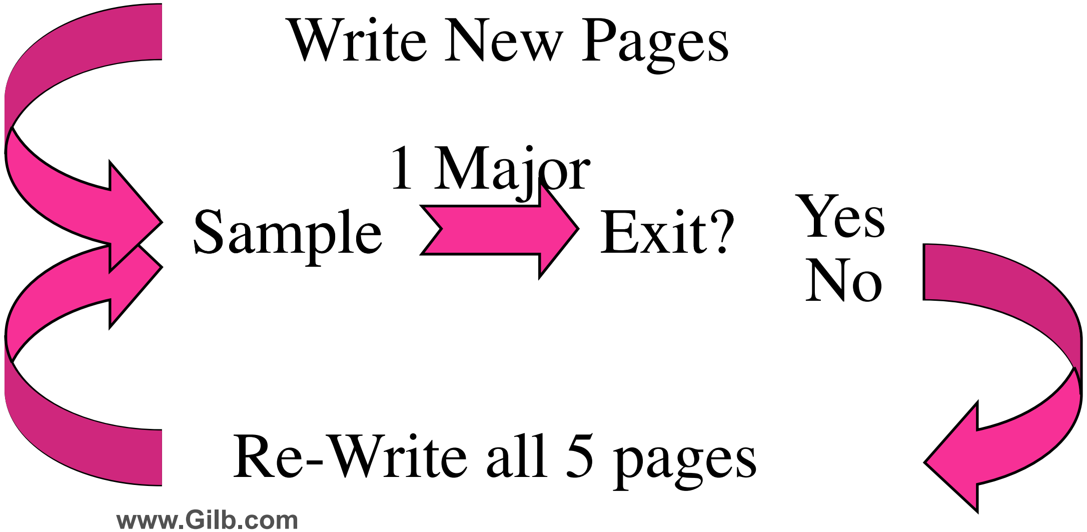
Then sample **another** page with Inspection

Too many Threats

Good Enough

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

The Author is expected to write about 45 pages



Write New Pages

Sample

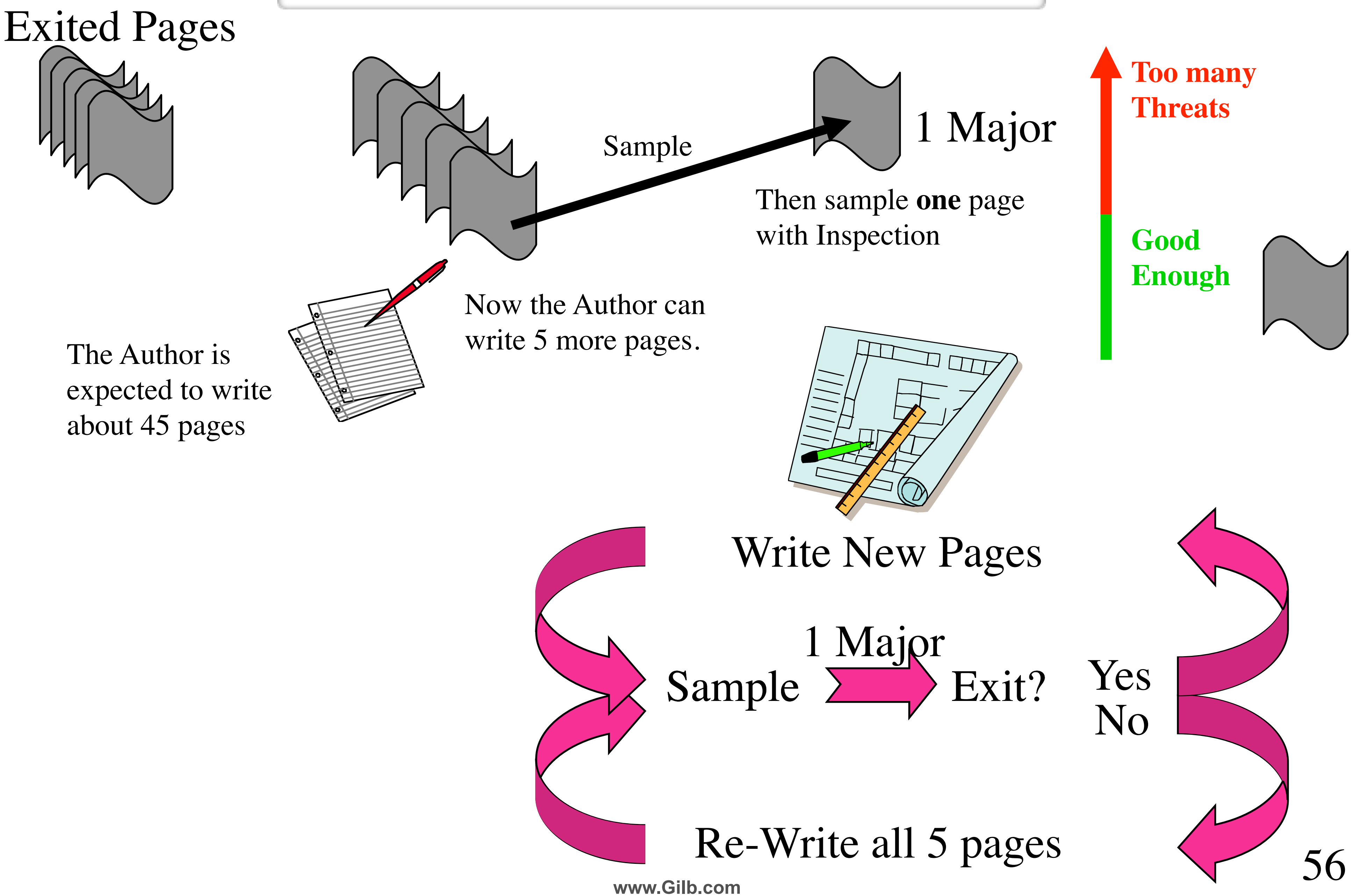
1 Major

Exit?

Yes
No

Re-Write all 5 pages

Sample “During” Authoring 3



Exited Pages

Sample

1 Major

Then sample **one** page with Inspection

Too many Threats

Good Enough

The Author is expected to write about 45 pages

Now the Author can write 5 more pages.

Write New Pages

Sample

1 Major

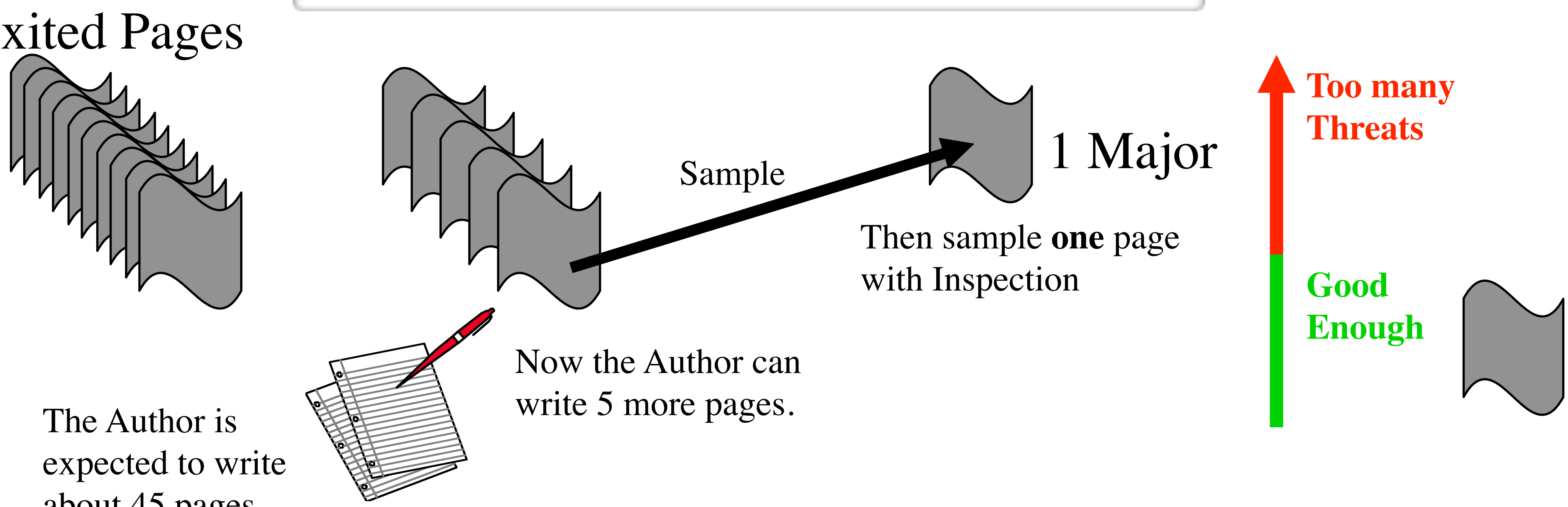
Exit?

Yes

No

Re-Write all 5 pages

Sample “During” Authoring 4



Exited Pages

Sample

1 Major

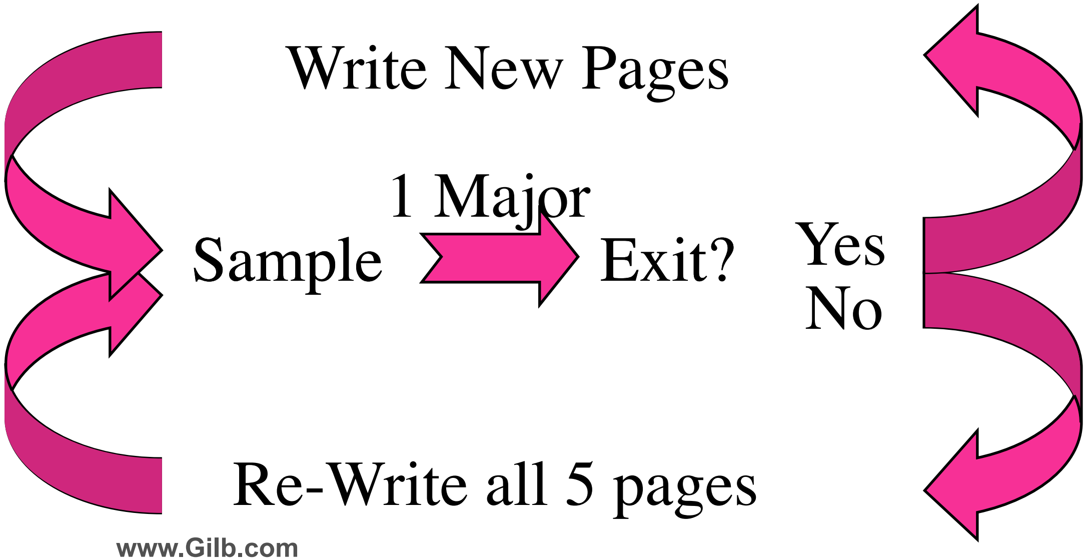
Then sample **one** page with Inspection

Too many Threats

Good Enough

Now the Author can write 5 more pages.

The Author is expected to write about 45 pages



Write New Pages

1 Major

Sample

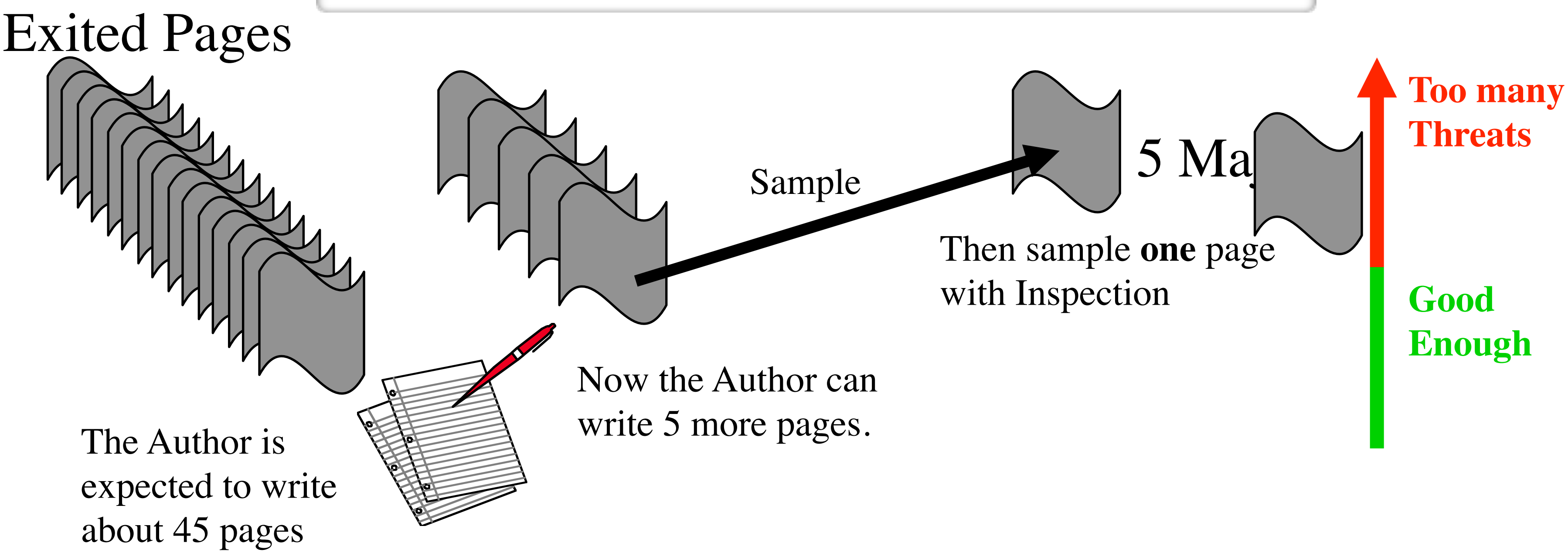
Exit?

Yes

No

Re-Write all 5 pages

Sample “During” Authoring 5



Exited Pages

Sample

5 Ma

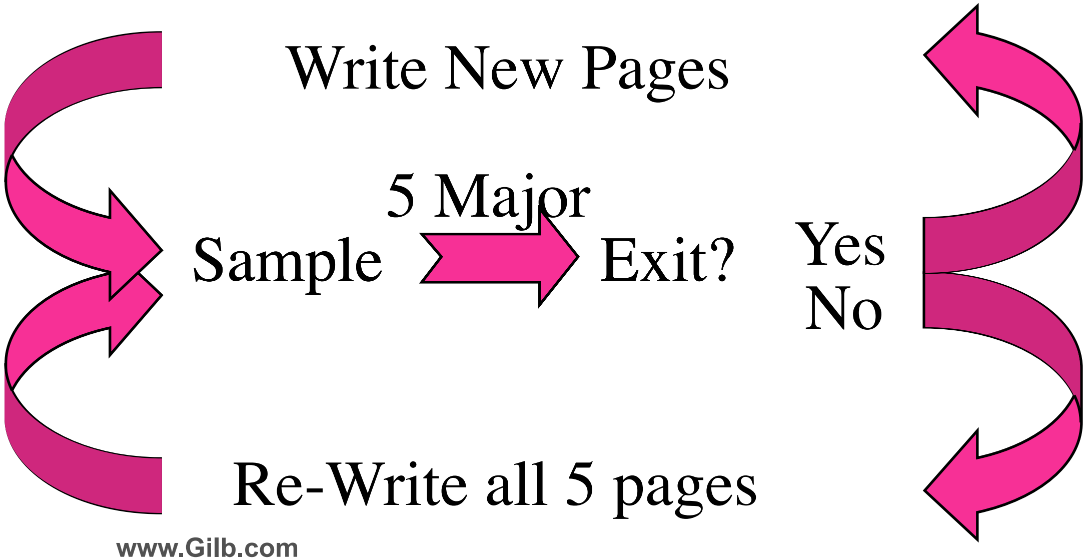
Then sample **one** page with Inspection

Too many Threats

Good Enough

Now the Author can write 5 more pages.

The Author is expected to write about 45 pages



Write New Pages

5 Major

Sample

Exit?

Yes

No

Re-Write all 5 pages

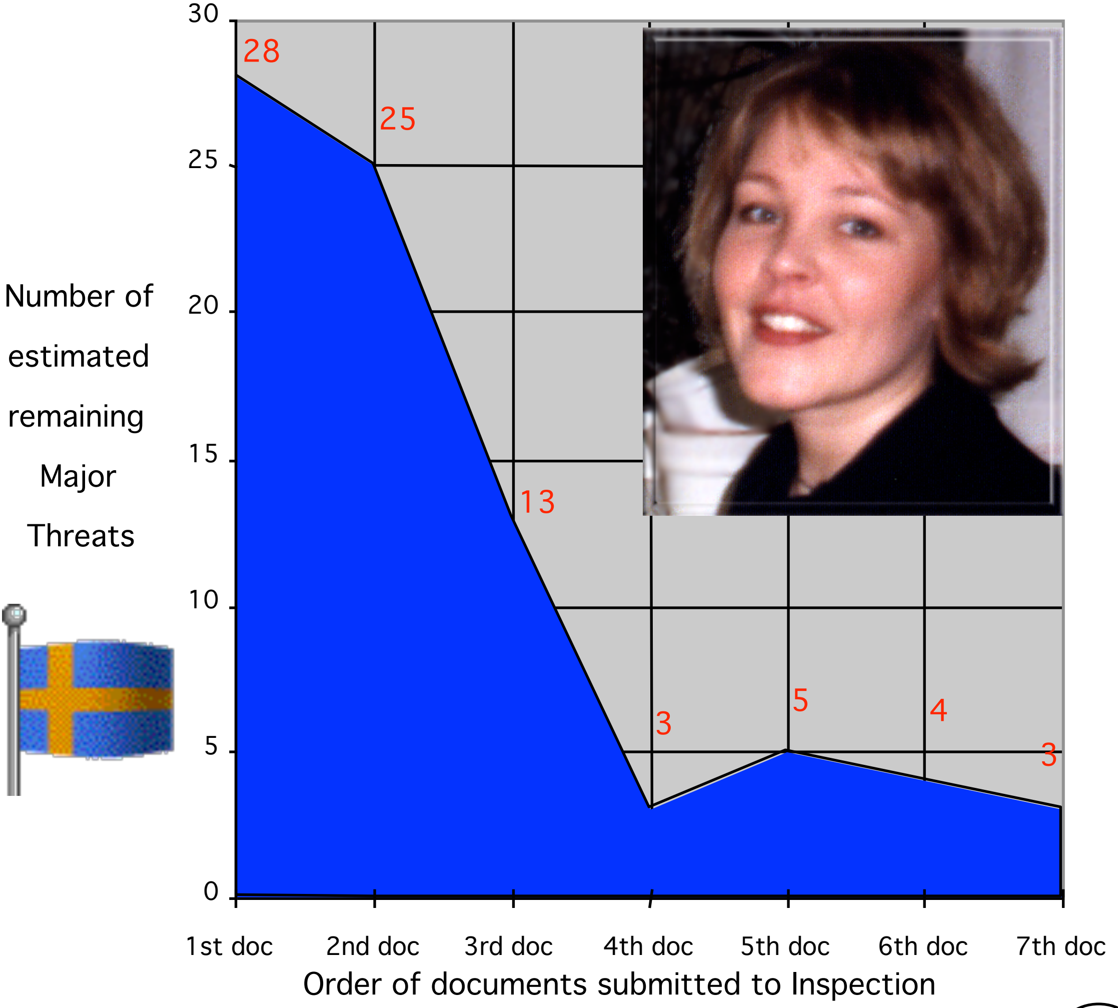
Individual learning Curve

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



Workshop

Hold Lean Inspection

- Team up 2 to 3 people on each team.
- Plan
  - Product Doc - Mother doc. - Rules
  - Decide on a numeric Exit criteria.
  - Decide on Sample.
    - Review Plan with Tom or Kai

- Kickoff - Checking - Reporting - Exit
- Review numeric results with class
  - Estimated Major per Page
  - Actual Checking Rate
  - Total Cost of Inspection
  - Exit yes or no
  - Any other insights/comments

Dr. Juran’s Test

The One Page - Lean Inspection Process

EI Procedure

EI.P1: Team. The specification writer ('writer') finds one other person (called a Checker) to (help) carry out the QC (Quality Control) of their specification.

EI.P2: Time. a meeting time, with maximum duration 1.0 hour is agreed, no matter the size of the document. (if the Checker is experienced, they can in fact do their checking at any time, alone, and report their results to the writer.)

EI.P3: Process-Brief. The writer makes sure the checker is knowledgeable about the following:

Readership: the spec's intended readership and their uses of the spec.

Rules: the specification Rules that apply (and their practical interpretation)

Major: The definition of Major defect, and how to spot them

Purpose: the purpose of the Spec QC process (to help the writer get to real exit-able level of defect density).

EI.P4: Page. The writer and the checker will each select the same one logical page 'at random' (300 Non-commentary words) sample to check. The writer is now performing the role of a 'checker' on their own work. They should agree that the page selected is representative of the quality of the rest of the document.

EI.P5: Checking-Style. Checking will be done individually (but maybe in same room)

EI.P6: Checking-Time. The initial checking time will be 10 minutes. If NO Major defects are found by either checker. The checking process will continue for another 30 minutes. Even if no further Majors are found.

EI.P7: Early-Exit. If any Major defect is found (and acknowledged by the writer as a real Major defect) in the first 10 minutes of checking, then this will be considered a sign that the spec contains many more major defects. The writer will consider whether they want to stop the QC process and improve the spec, or whether they want to continue for another 30 minutes to gather more Major defect cases (to better signal what they need to rewrite).

EI.P8: Estimation. At the end of the checking time, the writer (or the checker if they decide to take reporting responsibility) will calculate the estimated Majors/Page in the current document (using formulas or tools supplied) and will report (on a form or to a database) all time used and results (Majors found, Majors/page estimated, decision to

Workshop

Hold Lean Inspection

- Team up 2 to 3 people on each team.
- Plan
  - Product Doc - Mother doc. - Rules
  - Decide on a numeric Exit criteria.
  - Decide on Sample.
    - Review Plan with Tom or Kai

- Kickoff - Checking - Reporting - Exit
- Review numeric results with class
  - Estimated Major per Page
  - Actual Checking Rate
  - Total Cost of Inspection
  - Exit yes or no
  - Any other insights/comments

Workshop

Hold Lean Inspection

- Team up 2 to 3 people on each team.
- Plan
  - Product Doc - Mother doc. - Rules
  - Decide on a numeric Exit criteria.
  - Decide on Sample.
    - Review Plan with Tom or Kai

- Kickoff - Checking - Reporting - Exit
- Review numeric results with class
  - Estimated Major per Page
  - Actual Checking Rate
  - Total Cost of Inspection
  - Exit yes or no
  - Any other insights/comments

Inspection Results

Major	minor	Design	time	words	pages	Major/300 words	Major/ min.
7	3		20	341	1,1	6,2	0,4
2			20	155	0,5	3,9	0,1
5			20	155	0,5	9,7	0,3
2	2		20	150	0,5	4,0	0,1
12	11		20	150	0,5	24,0	0,6
16	7		20	150	0,5	32,0	0,8
					0,0		
					0,0		

Extrapolations

	Team 1	Team 2	Team 3	Team 4	Team 5	Team 6
Number of Checkers	3					
Number of Unique Majors found per 300 words.	45					
% effectiveness	33 %	33 %	33 %	33 %	33 %	33 %
Actual Majors per 300 words	136	0	0	0	0	0
Pages (300 words) in document	5					
Total Majors in Document	682	0	0	0	0	0
1/3 actually occur	227	0	0	0	0	0
Average Cost in work hours / Major if let through	9	9	9	9	9	9
Estimated delay in work hours caused by Majors	2 045	0	0	0	0	0
Majors Remaining per page after edit	97	0	0	0	0	0

Inspection Results

Major	minor	Design	time	words	pages	Major/300 words	Major/ min.
40			30	100	0,3	120,0	1,3
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		

Inspection Results

Major	minor	Design	time	words	pages	Major/300 words	Major/ min.
40			30	100	0,3	120,0	1,3
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		

Inspection Results

Major	minor	Design	time	words	pages	Major/300 words	Major/ min.
40			30	100	0,3	120,0	1,3
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		

Inspection Results

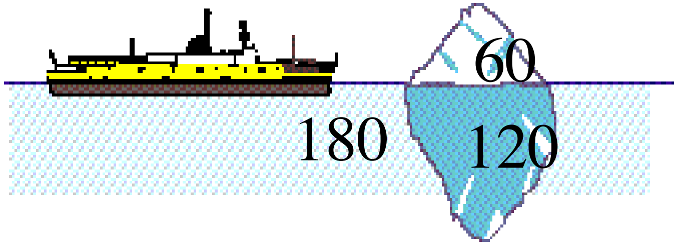
Major	minor	Design	time	words	pages	Major/300 words	Major/ min.
40			30	100	0,3	120,0	1,3
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		
					0,0		

Time used

Logical pages checked

Majors found

Defect Density Estimation



- Total for group (page ??)
  ?? x 2 = ?? Majors
  assume are unique.
- If 33.333% effective,
  total in page = 3x ?? = ???
- Of which 2/3 or ??? were not yet found.
- If we fix all we found (??), then the estimated remainder of Majors would be ??? (not found) +?? not fixed for real = ??? Majors remaining.

Conclusions

- **Human defect removal by Inspections/reviews/SQC is**
  - **a hopeless cause: not worth it.**
- **Spec QC can be used, in spite of imperfect effectiveness,**
  - **to accurately estimate major defect level per page.**
- **This measurement** can be used to **motivate engineers to**
  - dramatically **(100x! Over about 7 learning cycles)**
  - reduce their defect insertion **(rule violation)**
  - **to a practical exit level (like < 1.0 Major/page)**

Extrapolation to Whole Document

- **Average: ??? Majors/page**
  - Page ??: ??? Majors/page
  - Page ??: ??? Majors/page
- **Total in whole document: ??,??? Majors**
  - ??? Majors/page x ?? pages.

Estimated Project Loss

- If a Major has
  - 1/3 chance of causing loss
- And each loss caused by a Major is
  - avg. 10 hours
- then total project Rework cost is
  - about ??,000 hours loss.
  - 1 year = 2,000 hour 10 people

Assumptions

- Small teams will find double that of a single person.
  - So, double the Majors found by the best checker to get a good estimate of total unique Majors found by the team
- Team is 30% Effective (unexperienced team checking for 30 min.)
  - So, multiply what the team found by 3.
  - ?? x 3 = ??? Majors/page

Dr. Juran's Test
"The Game Rules"

Close your slide copies now!

No questions! No discussion. Make your best interpretation of these rules.

You will get 30 seconds to check

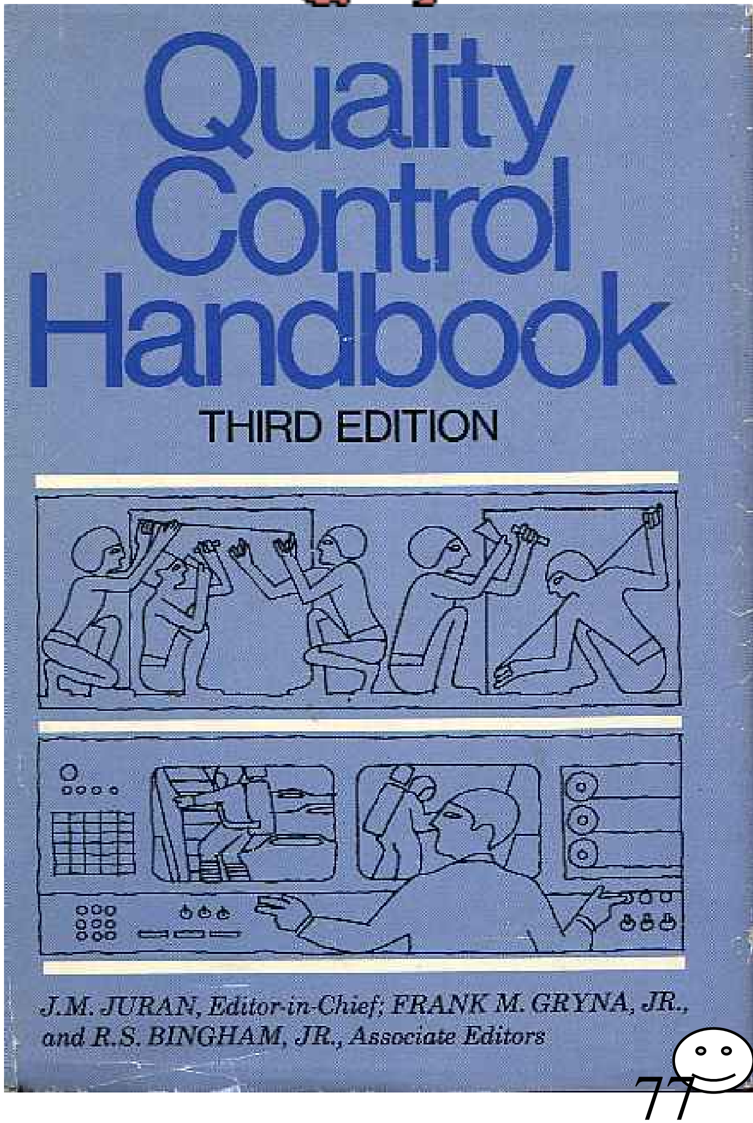
Count all defects for

Rule F: no instances of "F" (any type) allowed
on the screen.

Advice: count even "**remote cousins**" (example "f" and "*F* ")

Write down your count of "defects" on paper when the time is up (one answer only).

You may move to any position in the room to see better. Do not interfere with the view of others.


Quality Control Handbook
THIRD EDITION
J.M. JURAN, Editor-in-Chief; FRANK M. GRYNA, JR., and R.S. BINGHAM, JR., Associate Editors

Juran's “80%” Test

"FEDERAL FUSES ARE THE RESULTS OF YEARS OF SCIENTIFIC STUDY COMBINED WITH THE EXPERIENCE OF YEARS."

How many letter F's can you find on this page?

Write the number down in this box

Checklist for "F" Searching
All questions support “Rule F”

F1. Do you find the word "of"?

F2. Do you find large graphic patterns resembling F ?

F3. Did you look outside borders?

F4. Did you turn images backwards and all angles?

F5. Did you find all "F" shapes within other symbols?
for example in letter "E"?

F6. Did you find all numbers and shapes pronounced "F",
for example 14, 75 and "frames"?

F7. Did you examine things under a microscope?

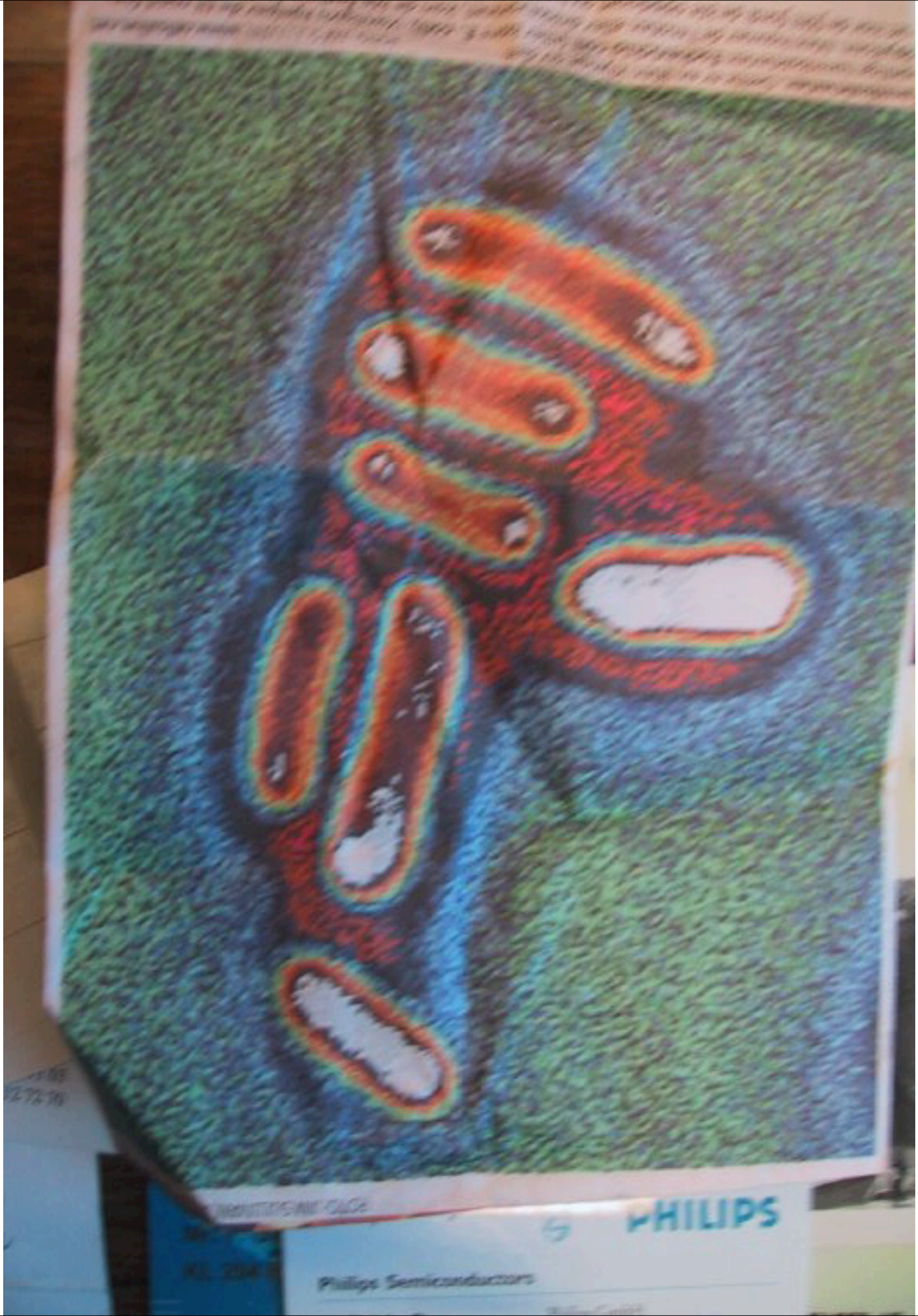
F8. Did you check the back of the screen?

F9. Did you look for lettering on the screen casing?

F 10. Check “f” sound in apostrophe and hyphen

F11. did you see the upside-down, backwards letter “t” (= “f”)?

Rule F: no instances of “F” (any type) allowed on the screen.



PHILIPS
Philips Semiconductors

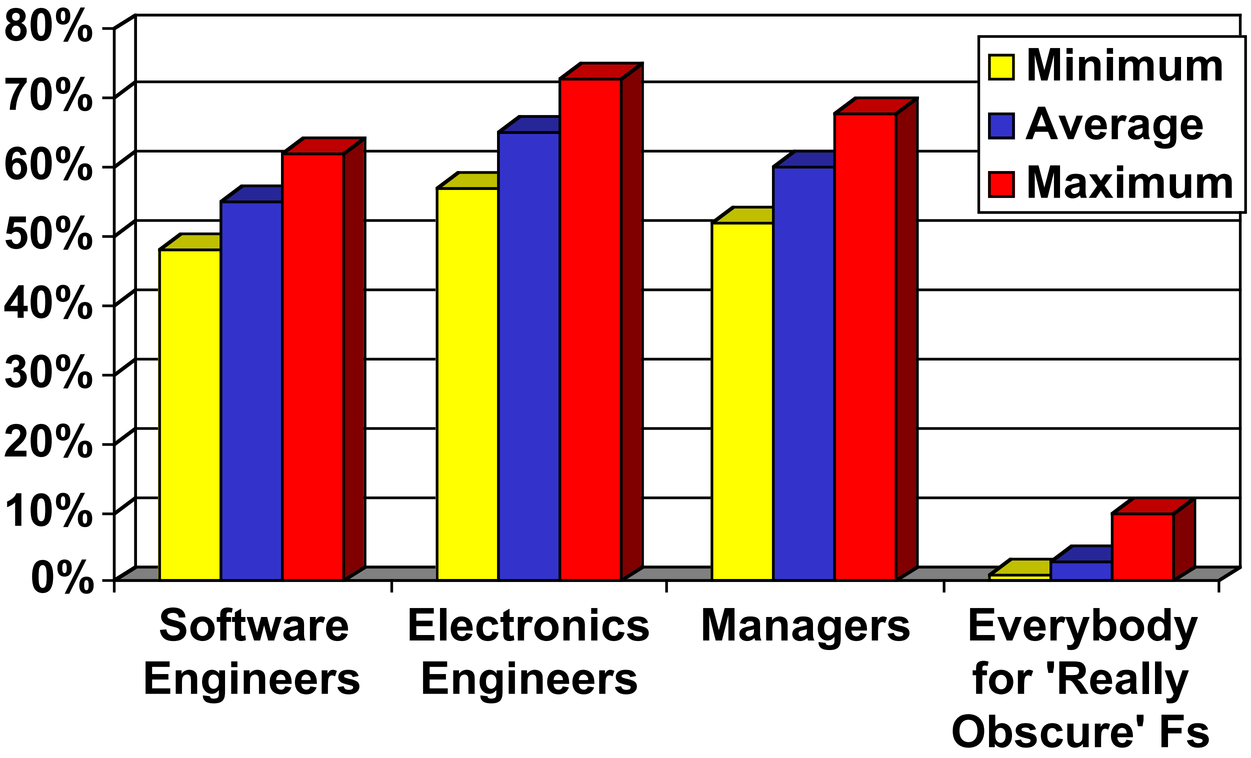
F Test Predictions

Prediction (made in advance, we will check it out later):

The group average for "obvious" F's will be

- 55%±7% for software people
- 65%±8% for engineers

The group average for 'really obscure' F's will be less than 10%.



The F-test ‘Lessons’

Defect finding is difficult

Even when a simple clear Rule defines a defect

Tools (Checklists) might be necessary to understand a simple Rule.

Checklists supplement but do not change the Rule

Time might be necessary to use the Tools.

We can predict number of defects NOT found!