SE4831 Software Quality Assurance
Inspections Part 2

- Objectives
  - Explain how checklists can be used to improve the effectiveness of a review process.
  - List the problems identified by the National Software Quality Experiment.
  - Explain how checklists can be used to improve the effectiveness of a review process.
  - Explain how generic checklists can yield reduced inspection effectiveness.
  - List the problems identified by the National Software Quality Experiment.
  - Critique inspection performance based on quantifiable metrics to identify potential problems.

Important aspects of inspections

- Importance of preparation
  - Majority of defects are discovered by individuals during their reviews
    - Not in the group meeting
- Size and composition of team vary based on artifact
  - Larger artifacts require multiple inspection sessions
  - Smaller artifacts may be inspected by smaller teams

Size Recommendation

- Large teams mean a large investment in staff time
  - Preparation time
  - Familiarization with material
  - Meeting time
- Small teams can be just as effective as large teams
  - Weller (1993)
    - 3 person inspection team with a lower preparation rate does as well as a four person team with a higher rate
    - Preparation rate rather than team size determines maximum effectiveness

Why review

- Petroski:
  - "To err is human"
- As wise engineers, however
  - We can deal with our errors!
    - By learning how, when, and why we error
    - By taking action to prevent our errors
    - By scrutinizing our products to find the effects of errors we missed
- Even given the advantages of peer review, peer reviews are not necessarily part of mainstream software engineering activities (Hatton)

State of the practice

Artifact	Percent Creating Artifact	Percent Performing review activities
Requirements	72%	
Design Document	68%	
Source Code	72%	

*Software Reviews: The State of the Practice (IEEE Software, 2003)

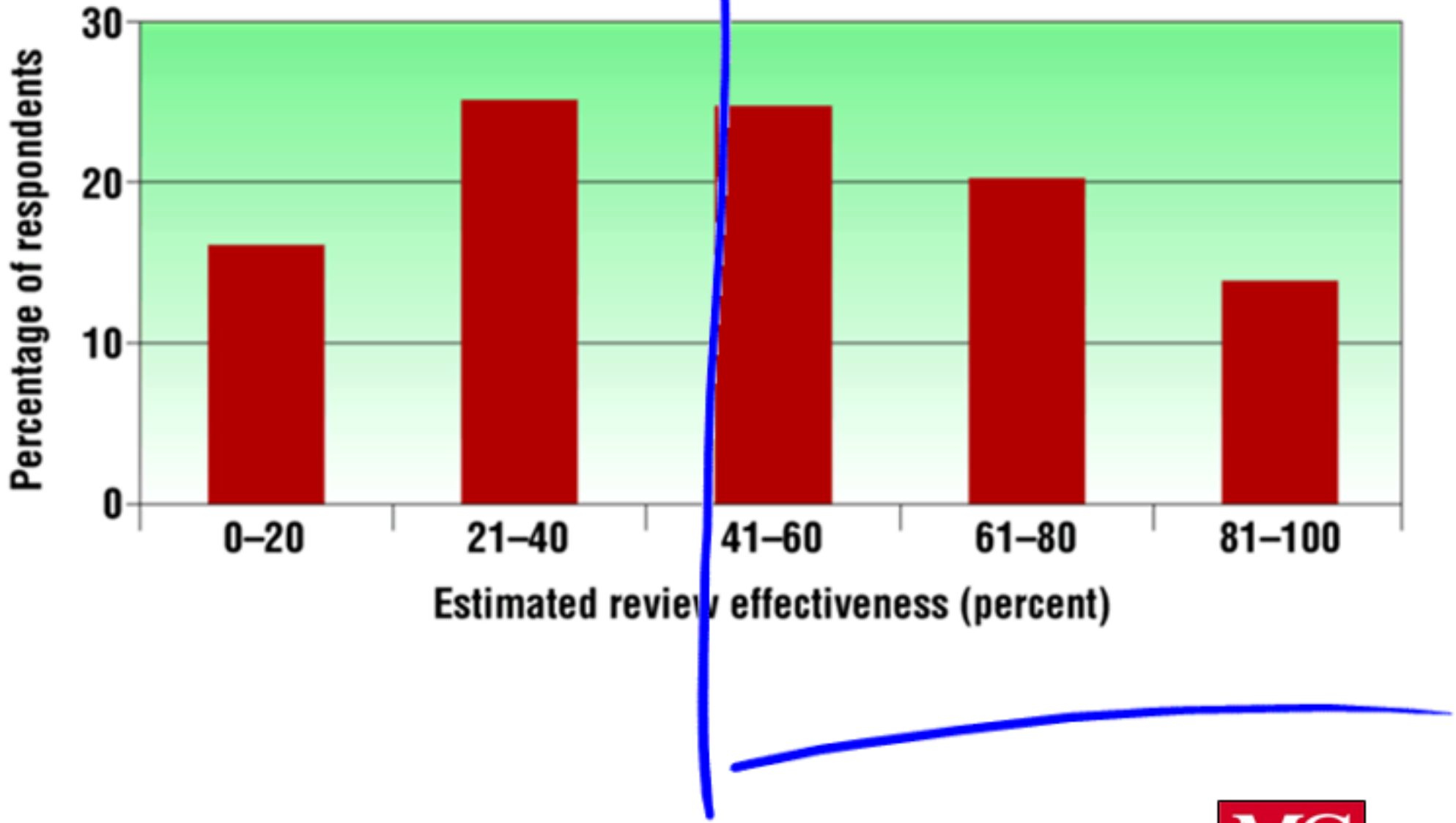
226 respondents

State of the practice

Artifact	Percent Creating Artifact	Percent Performing review activities
Requirements	72%	42%
Design Document	68%	40%
Source Code	72%	28%

*Software Reviews: The State of the Practice (IEEE Software, 2003)

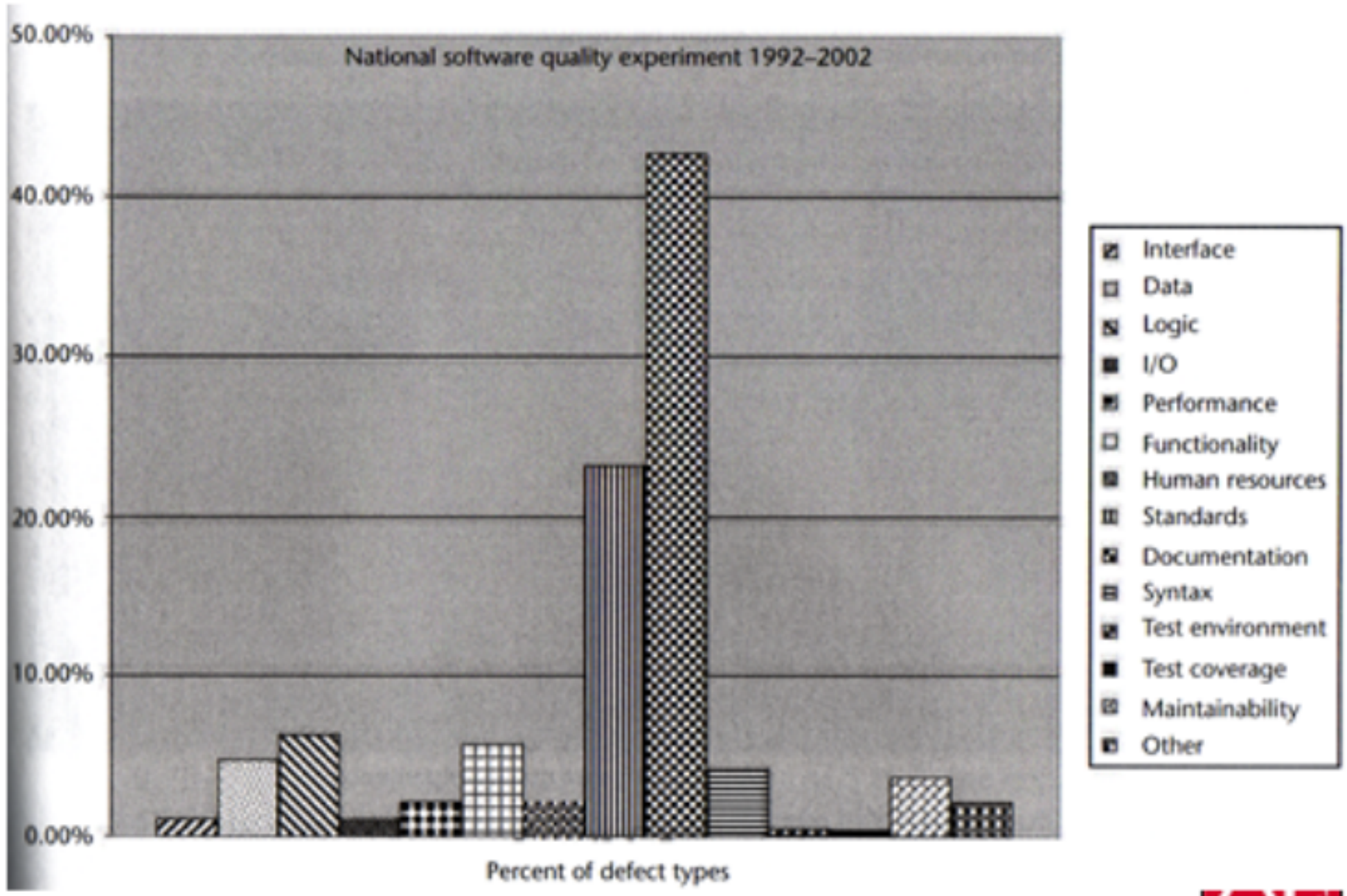
Estimated Review Effectiveness



State of the Practice

Review step	Primary survey results
Review goals	Main goals are quality improvement (73% of respondents) and enforcing standards (54%).
Development process	40% regularly review requirements and design documents, 30% regularly review code.
Review process	Only 20–40% conduct individual review steps regularly.
Planning, overview	20% usually perform planning, 30% do not use any formal entry or exit criteria for reviews.
Defect detection, preparation	40% usually conduct the preparation phase. Of those, 35% use ad hoc reading, 50% use checklists, and the rest use more advanced defect detection techniques or alternatives such as simulation.
Meeting	40% usually conduct a meeting.
Follow-up	35% have some kind of follow-up step. Independent of that, 40% verify rework, and 16% regularly give feedback.
Optimization	60% collect data, but 30% of these do no analysis, and fewer than 25% use the data to optimize reviews.

What do we find?



National Quality Experiment

Sessions	Prep Effort	Conduct Time	Major Defects	Minor Defects	Size in Lines
3,040	181,471	71,283	2,512	12,391	1 ,020,229

Metrics:

1.	12.18	Minutes of preparation effect per defect
2.	72.24	Minutes of preparation effort per major defect
3.	2.46	Major defects per thousand lines
4.	12.15	Minor defects per thousand lines
5.	858.74	Lines per conduct hour
6.	4.90	Defects per session
7.	0.64	Preparation/conduct effort
8.	335.60	Lines per session
9.	4.50	Return on investment

Common Problems

- 1. Software product source code components are not traced to requirements.
  - As a result, the software product is not under intellectual control, verification procedures are imprecise, and changes cannot be managed.
- 2. Software engineering practices for systematic design and structured programming are applied without sufficient rigor and discipline.
  - As a result, high defect rates are experienced in logic, data, interfaces, and functionality.
- 3. Software product designs and source code are recorded in an ad hoc style.
  - As a result, the understandability, adaptability, and maintainability of the software product are directly impacted.
- 4. The rules of construction for the application domain are not clearly stated, understood, and applied.
  - As a result, common patterns and templates are not exploited in preparation for later reuse.
- 5. The code and upload development paradigm is becoming predominant in emerging e-commerce applications.
  - As a result, the enterprise code base services only the short term planning horizon where code rules and heroes flourish, but it mortgages the future where traceable baseline requirements, specification, and design artifacts are necessary foundations.

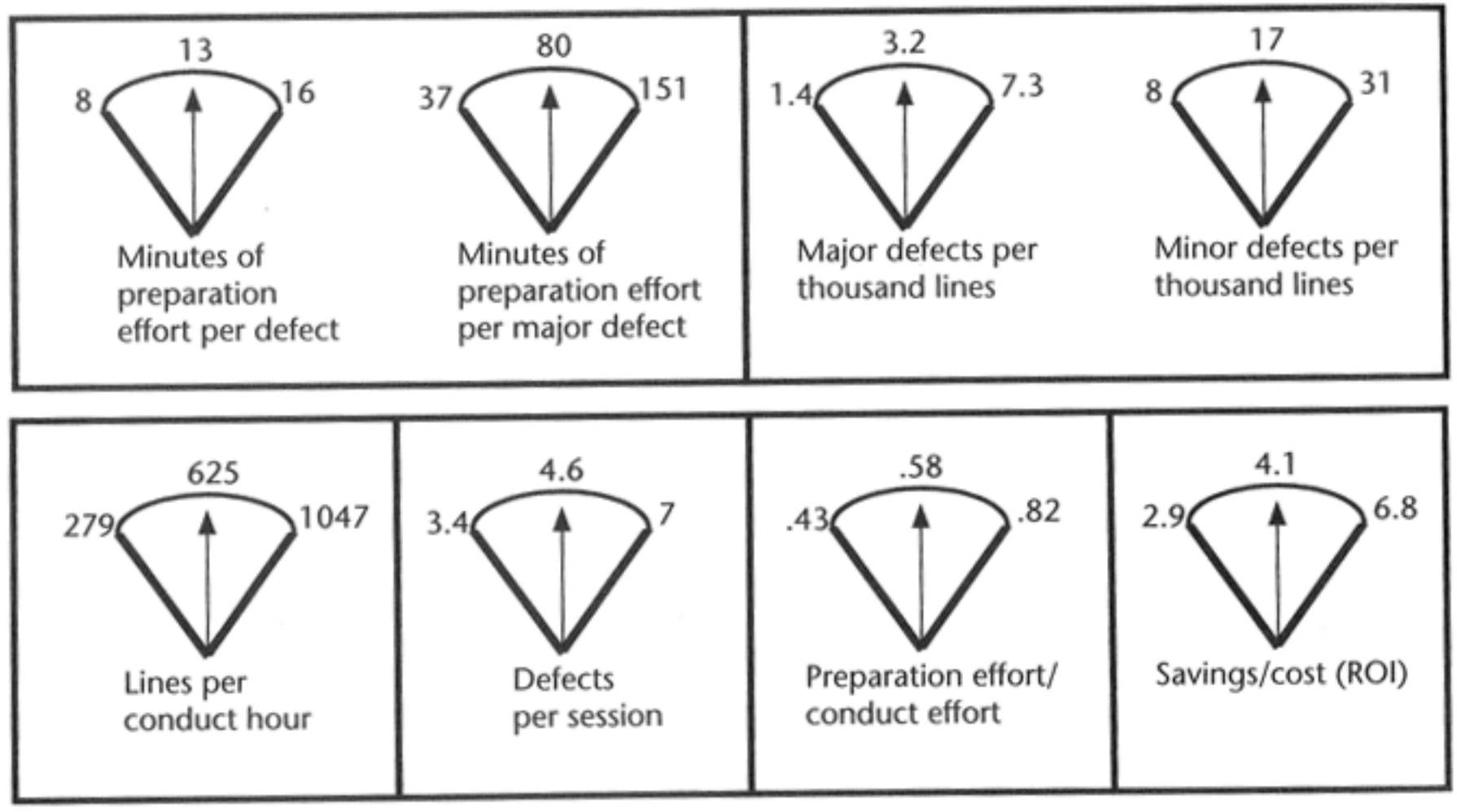
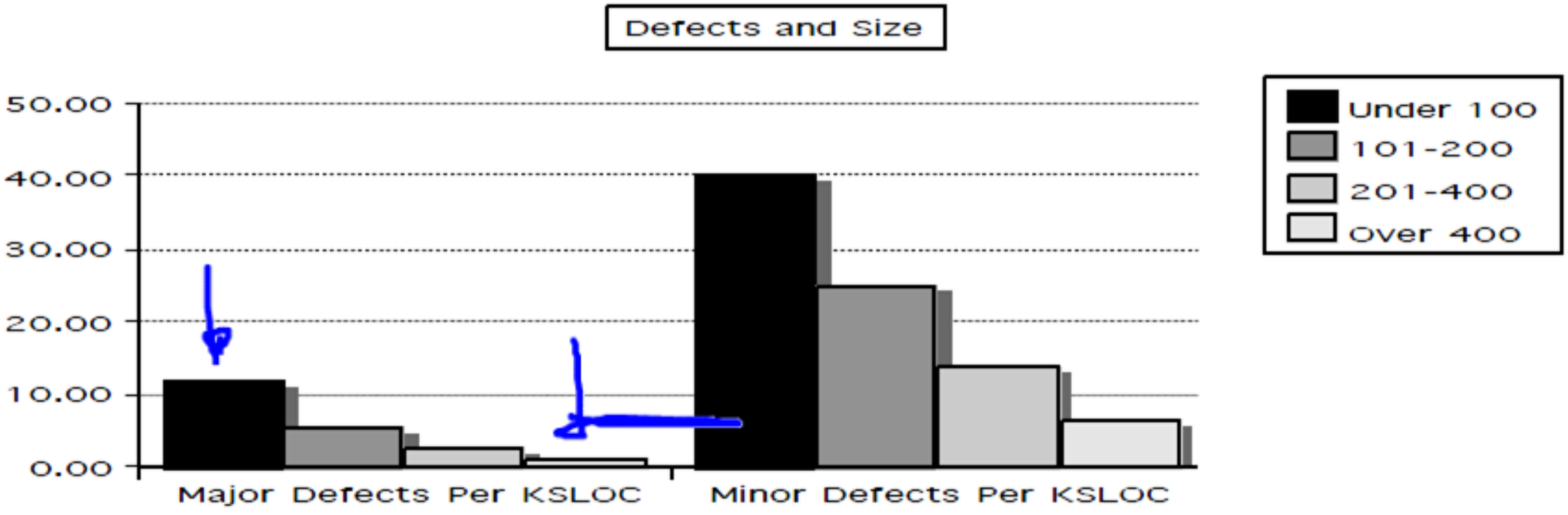


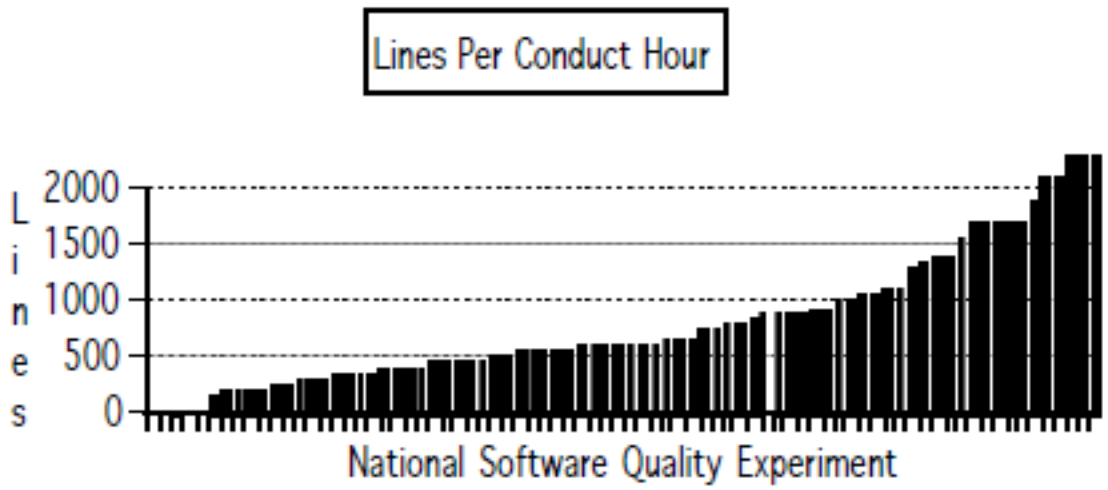
Figure 7.7 Software Inspections Control Panel.

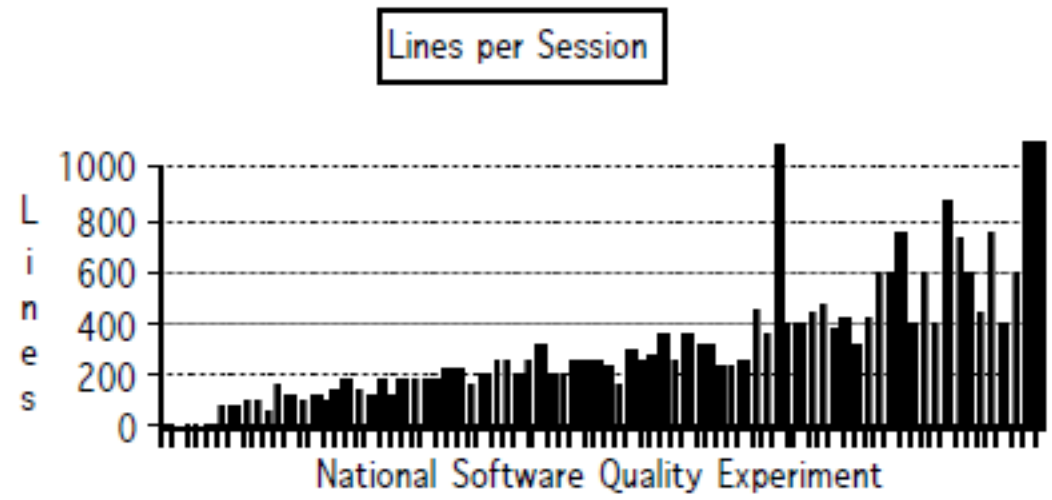
DEFECT RATES AND PRODUCT SIZE

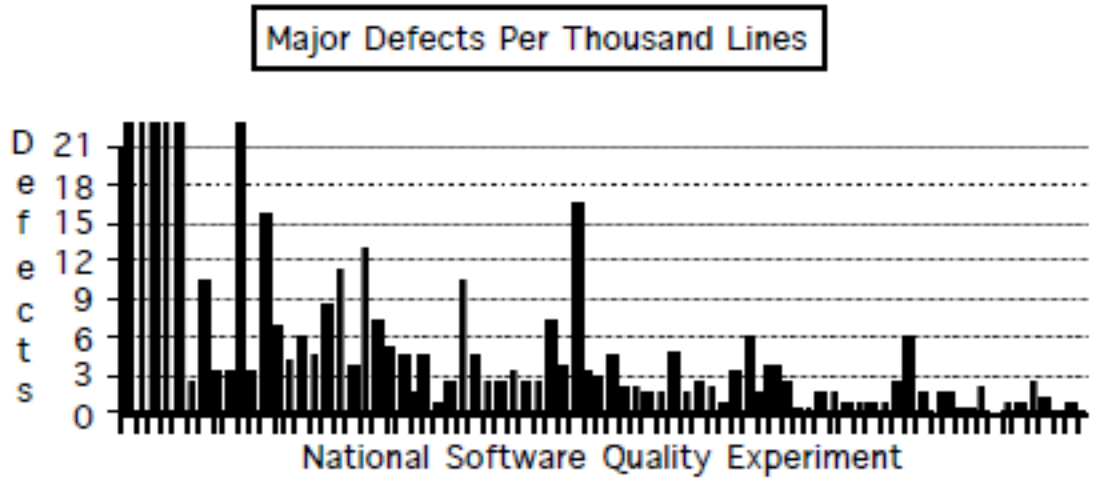
An analysis was conducted on defect rates and product size. The rate of defects detected in the Software Inspection Lab reveals an inverse relationship to product size. All programs contain a beginning, an end, and a context for operation within the larger system. Starting, finishing, and fitting in are all more error prone than the body of the program which gives it size.

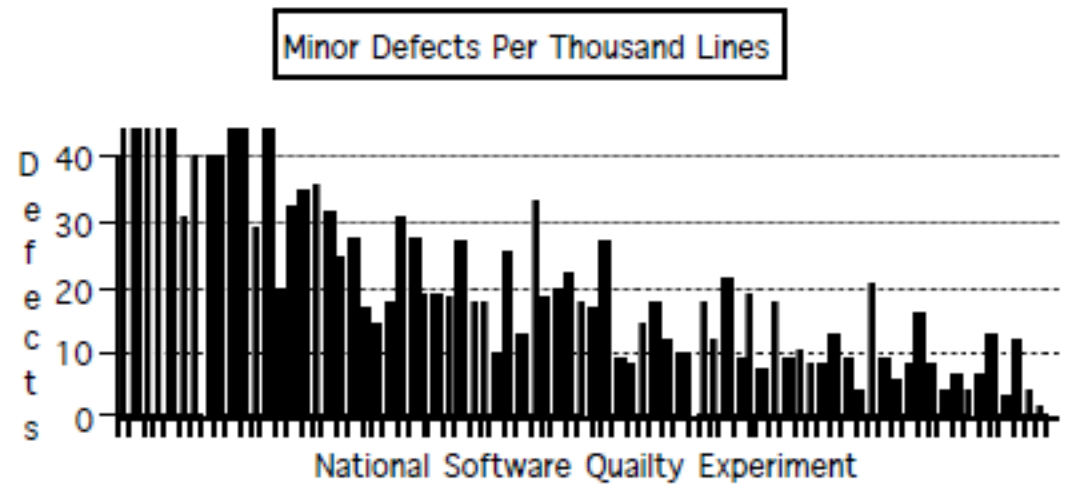


Review Sizes

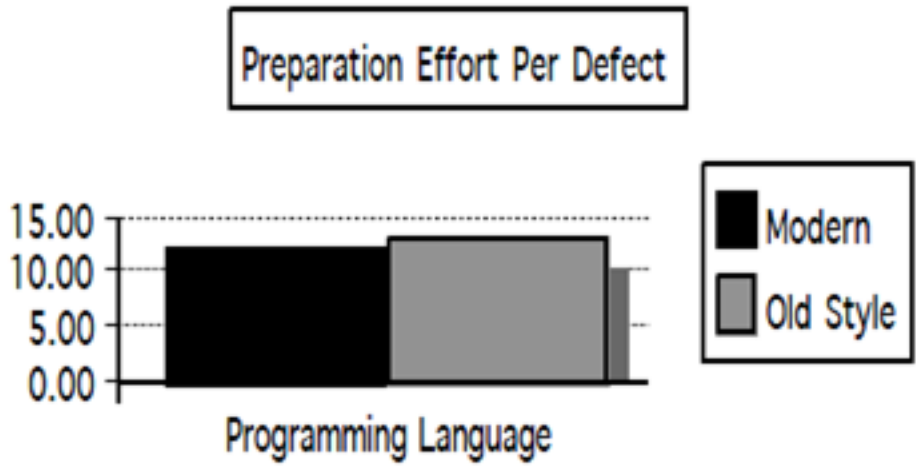


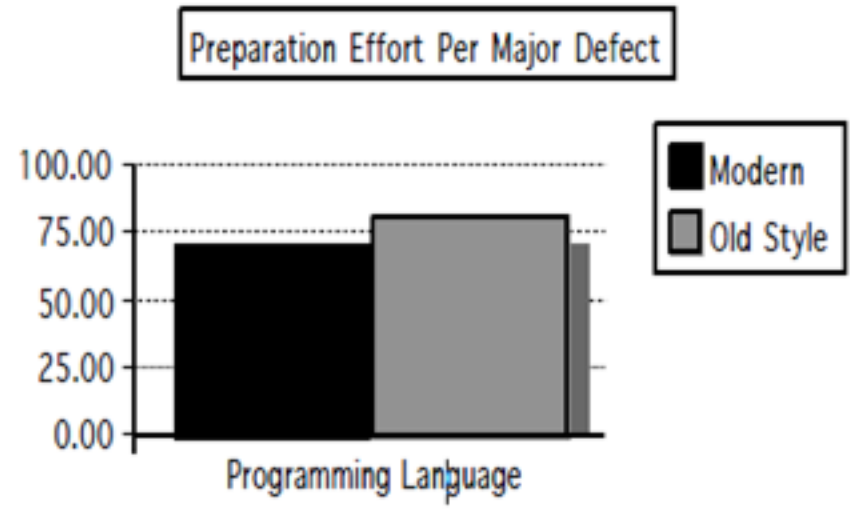


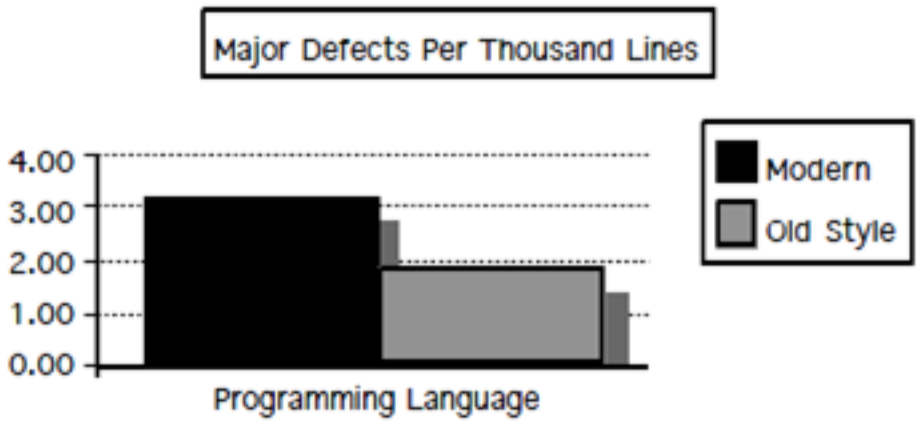


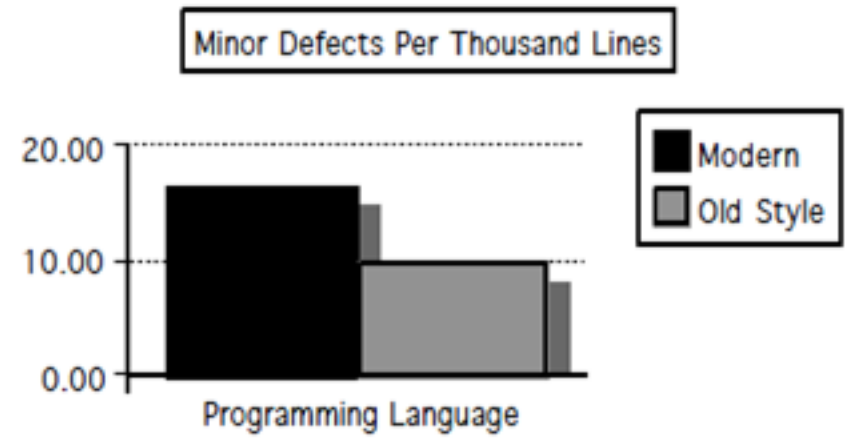


Programming Language

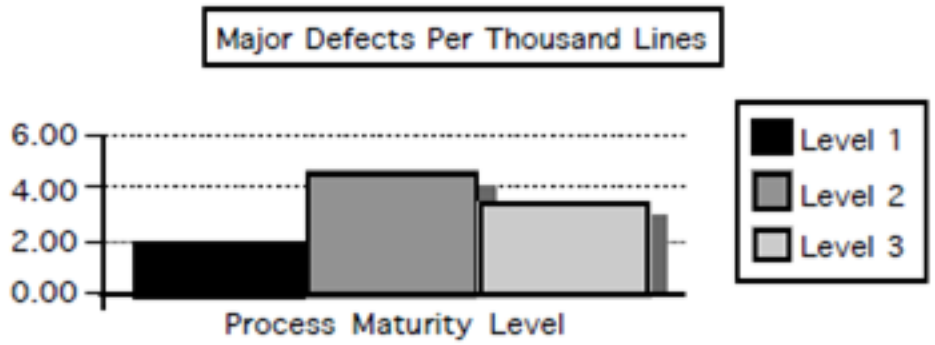


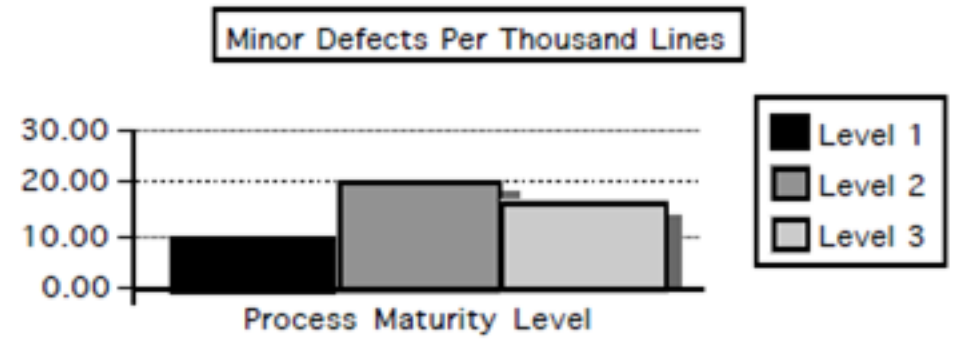




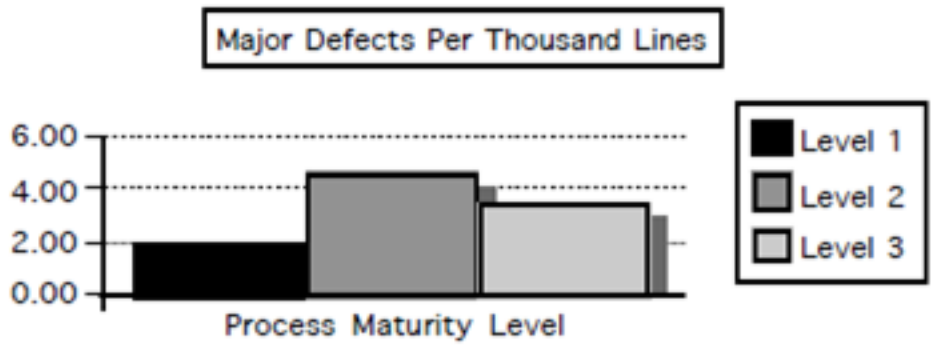


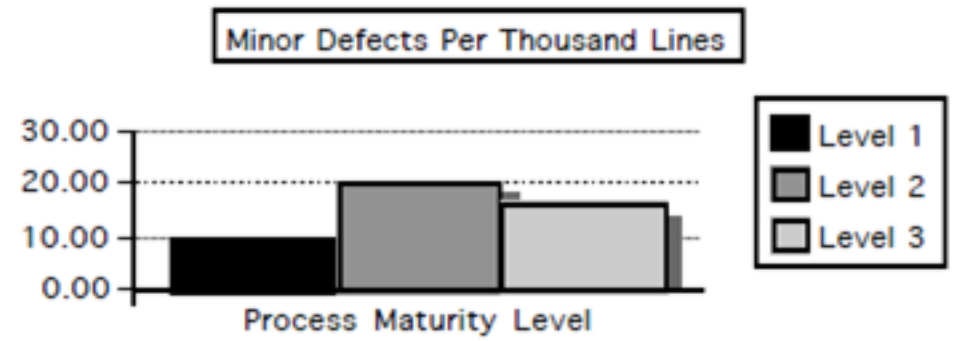
CMM Level Versus Review Effectiveness

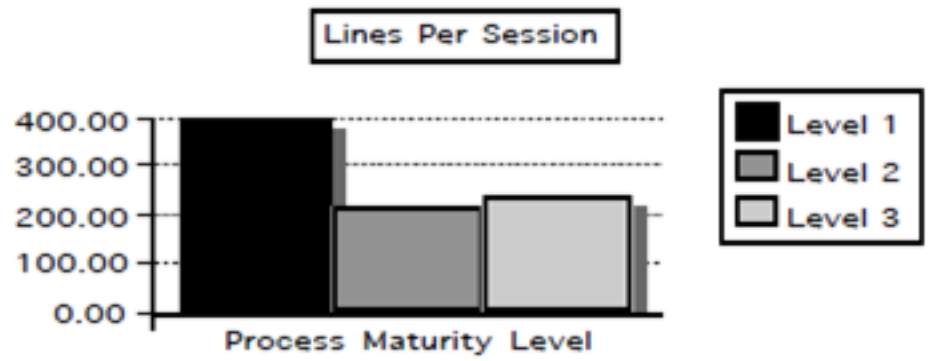


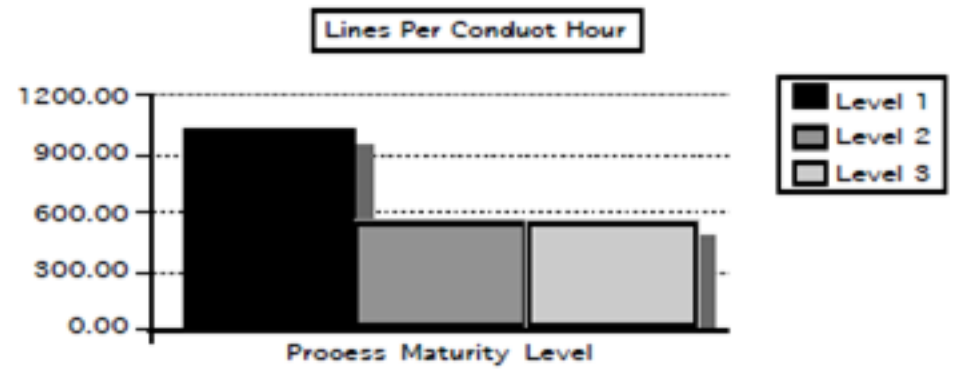


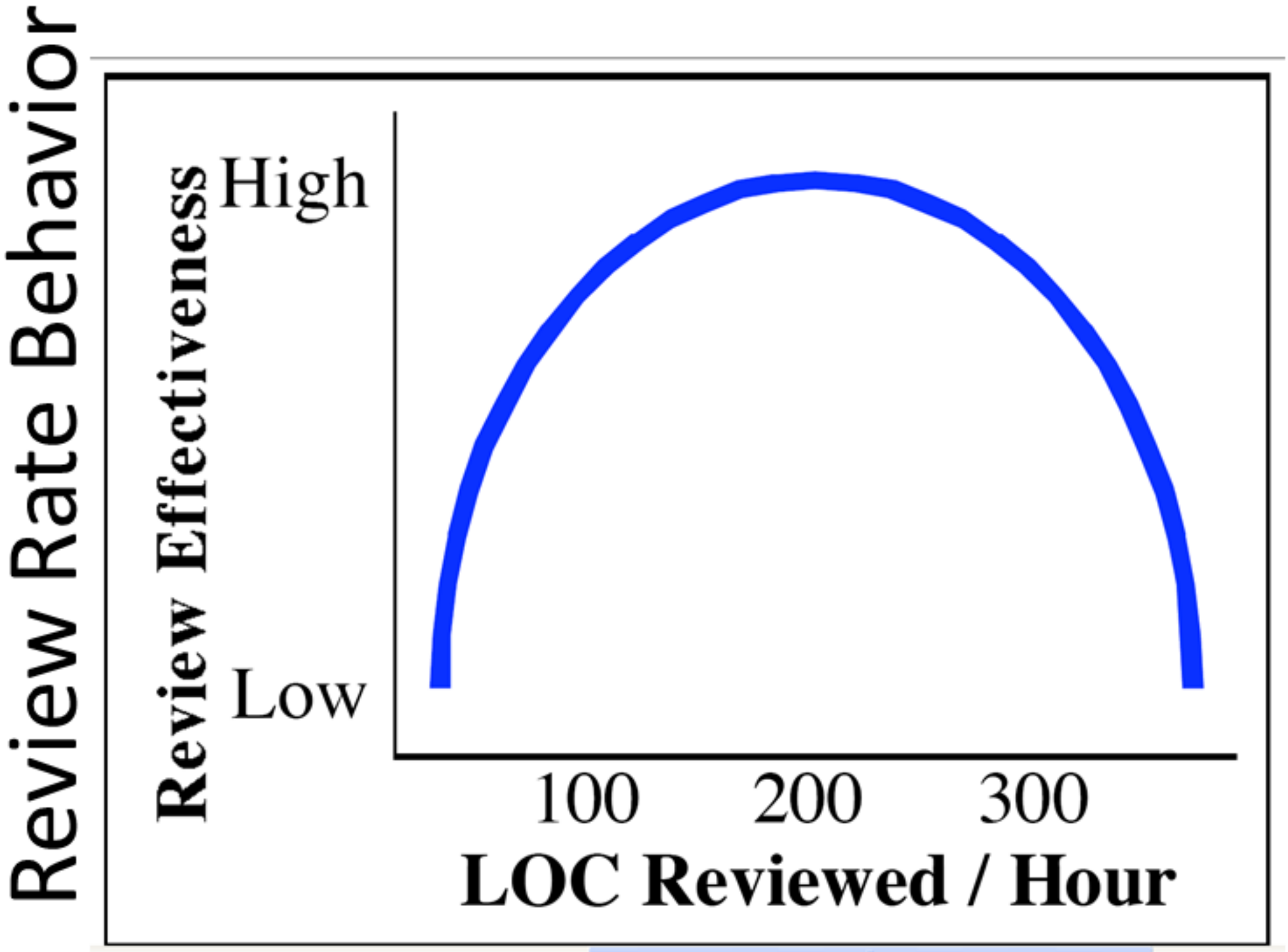
CMM Level Versus Review Effectiveness










Review Rate Behavior
Review Effectiveness
High
Low
100
200
300
LOC Reviewed / Hour

Recommended Review Rates (Humphrey, 2000)

Artifact	Preparation Time
Requirements Reviews and inspections	2 pages per hour
HLD	5 pages per hour
DD	100 pseudo-code lines per hour
Source Code	200 LOC per hour

Recommended Review Rates (Jones (1991)

Artifact	Preparation Time	Meeting Time
Requirements Specification	25 pages per hour	12 pages per hour
Functional Specification	45 pages per hour	15 pages per hour
Logical Specification	50 pages per hour	20 pages per hour
Source Code	150 LOC per hour	75 LOC per hour
User Documentation	33 pages per hour	20 pages per hour

Caution: These rates seem a bit excessive to me, but I haven't found a better set of numbers that I can share as of yet.

Checklists

Checklist Effectiveness (Hatton, 2008)

Category	Mean	Std. dev.	Number
Entire population	13.66	5.12	238
Individual faults found using checklists	13.97	5.27	106
Individual faults found not using checklists	13.40	5.00	132
Common faults found using checklists	7.87	3.69	53
Common faults found not using checklists	7.41	4.06	66

Review Effectiveness (Hatton, 2008)

Category	Mean	Std. dev.	Number
Experienced population using checklists	19.60	4.02	35
Experienced population not using checklists	19.56	3.47	35
Inexperienced population using checklists	8.22	2.24	32
Inexperienced population not using checklists	8.81	1.92	52

Reasons why?

Problems with inspections

- False Sense of security
  - Weller(1993)

Other inspection techniques

Two person inspection

- Simplification of the Fagan Process
  - Uses an author – inspector pair
- Author writes document
- Inspector reviews document
- Roles reversed later on in review
- Mutually beneficial to both team members
  - Different from Fagan inspection
- Suited for small programs or small changes to larger documents
- Significantly less cost than formal Fagan inspection
- Problems / risks
  - May be a lack of technical knowledge by reviewer
  - Lack of experience

Pair Programming

- All production software in XP is built by two programmers, sitting side by side, at the same machine.
  - Ensures that all production code is reviewed by at least one other programmer
  - Results in better design, better testing, and better code.
  - Studies tend to indicate no loss in productivity
- Code developed 15% slower, but with ≈15% fewer bugs
- Problems
  - Personality conflicts
  - Business Management justification
  - Lack of metrics
  - Can lead to pigeon-holing if not managed properly.

Meetingless Inspection

- Most defects discovered by inspectors during individual inspections
  - Only 5 – 30% found during review meeting
- "Meetingless" inspection developed
- Uses key concepts from Fagan Inspection, except no inspection meeting
  - No interaction / communication between inspection team members
- Advantages
  - Lower cost
  - Easier to schedule
- Problems
  - Higher false positive rate
  - Missed defects