

CISC 327 - Software Quality Assurance

Lecture 20

Inspection

Inspections

- Today we begin our look at **inspection** as a quality assurance technique
 - Statistical Process Control
 - What is **inspection**?
 - **Informal** vs. **formal** inspection
 - Inspection in the software **process**
 - Inspection **roles**
 - **Effectiveness** of inspections vs. testing

Statistical Process Control

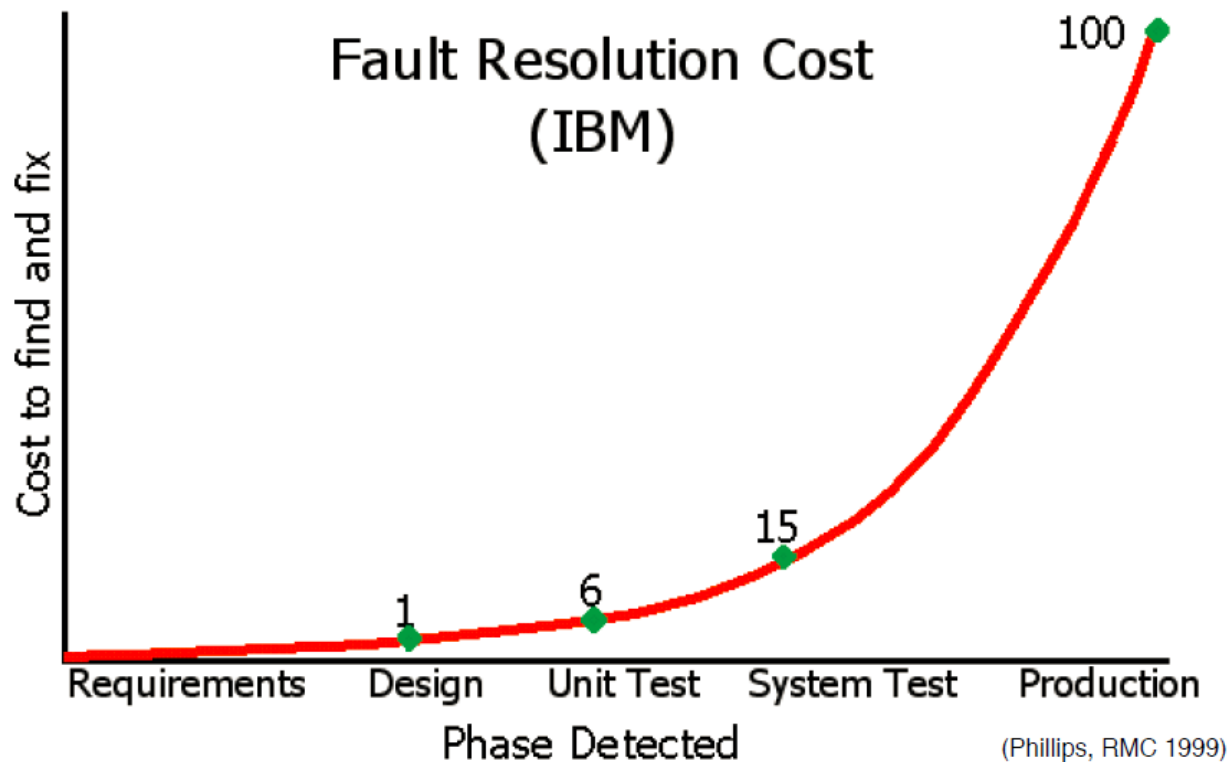
- Historical background: W.A. Shewhart
 - 1924: Bell wanted more reliable phone equipment
⇒ *control chart*
 - 1939: *Statistical Method from the Viewpoint of Quality Control*
 - Influential in US during the war → postwar Japan

First Law of Software Development

- **Earlier is Cheaper**

- The later in the development cycle a fault is detected, the more **expensive** it is to fix

- Methods that find faults **earlier** deliver more bang for the buck



Software Development Products

- What do we **produce** when making software?
 - Plans, procedures, requirements specifications, design specifications, source code, comments, test cases, test reports, user documentation, technical documentation
- Of all these, we can only **test** one of them (code), and only when we are already far along (at least partially runnable)
- So how can we discover and address quality and detect faults **earlier**?

Reviews, Walkthroughs, and Inspections

- **Terminology**

- Unfortunately, there is no good agreement on **precise** definitions for these terms, but...

- **Reviews**

- ...are the management practice of **meetings** to informally consider state of the project at certain stages, to gain confidence in project direction
 - e.g., preliminary **design review**, critical **design review**
- Used to provide **confidence** that the design is sound
- Often attended by **management** and customers

Reviews, Walkthroughs, and Inspections

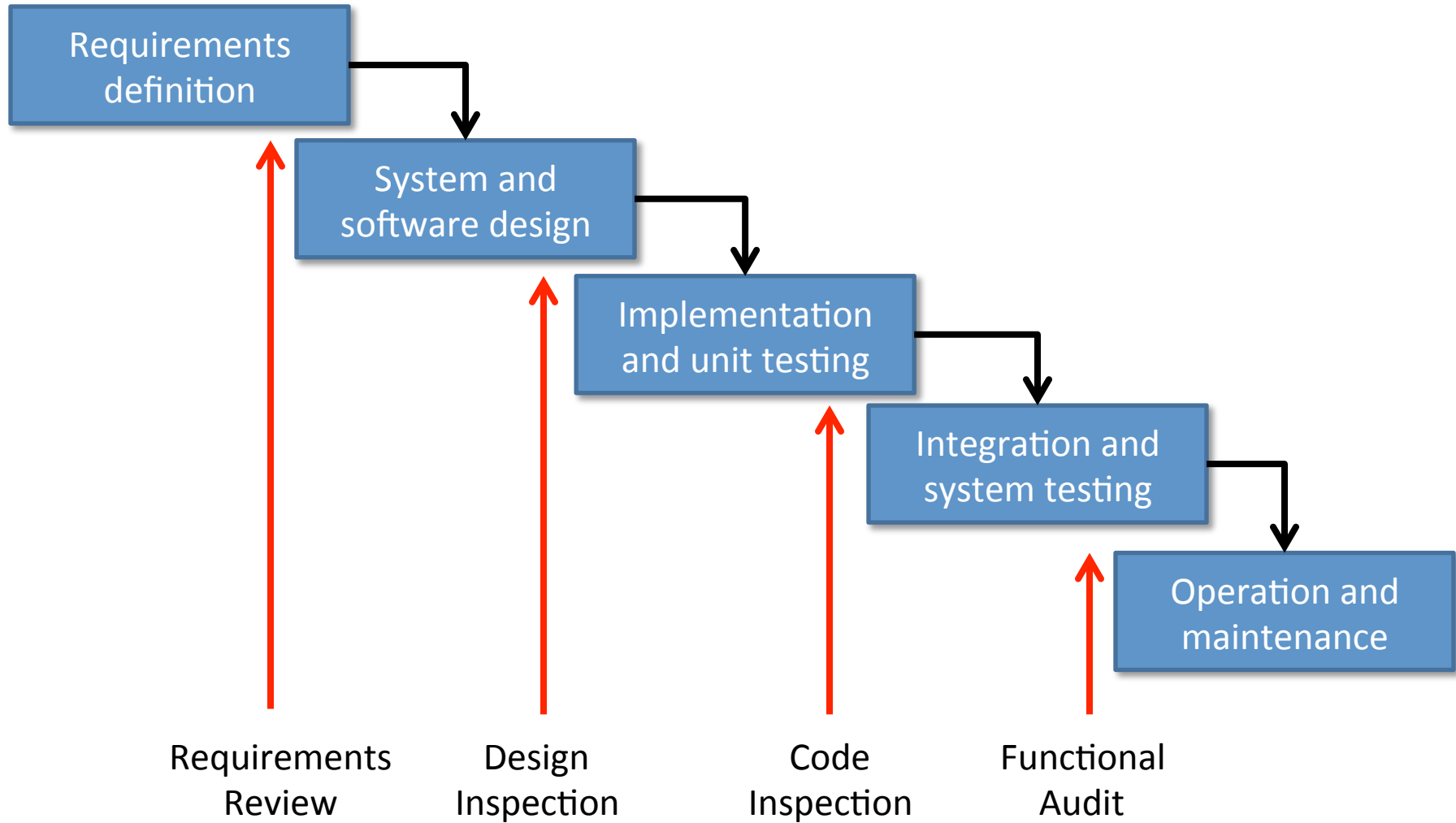
- **Walkthrough**

- ...refers to an **informal technical** review, normally carried out by developers
- Used by development teams to improve product quality by involving whole team in **quality assurance** at each stage
- Focus is on **critical analysis** of artifacts, in an attempt to find or predict defects

Reviews, Walkthroughs, and Inspections

- **Inspection**
 - ...refers to a completely **formal** process of review, also known as **formal technical reviews**
 - A formal system used to identify and remove **defects**, and improve the overall **quality** of the development process
 - **Involves**: Formal written **reports**, defect data collection and **analysis**, required standards and **measures**
 - Emphasis on **documenting** process progress and defects
 - First introduced by **Fagan** (IBM) about **1976**, now **required** by some customers (e.g., U.S. military)

Inspections in the Software Process



Kinds of Inspections

- **A Generic Technique**

- Inspections can assist at **every stage**, the earlier the better
- E.g., U.S. Mil-Std-1521B, "**Technical Reviews and Audits for ... Computer Software**" identifies 10 separate kinds to be carried out

System requirements review (SRR)	Test readiness review (TRR)
System design review (SDR)	Functional configuration audit (FCA)
Software specification review (SSR)	Physical configuration audit (PCA)
Preliminary design review (PDR)	Formal qualification review (FQR)
Critical design review (CDR)	Production readiness review (PRR)

Example: PDR

- 3.4 **Preliminary Design Review (PDR)**.
 - This review shall be conducted for each configuration item or aggregate of configuration items to
 - (1) **evaluate** the progress, technical adequacy, and risk resolution (on a technical, cost, and schedule basis) of the selected design approach,
 - (2) **determine** its compatibility with performance and engineering specialty requirements of the Hardware Configuration Item (HWCI) development specification,
 - (3) **evaluate** the degree of definition and assess the technical risk associated with the selected manufacturing methods/ processes, and
 - (4) **establish** the existence and compatibility of the physical and functional interfaces among the configuration item and other items of equipment, facilities, computer software, and personnel.

Example: PDR

- 3.4 **Preliminary Design Review (PDR)**.
 - ...
 - For CSCIs, this review will focus on:
 - (1) the evaluation of the progress, consistency, and technical adequacy of the selected **top-level design** and **test approach**,
 - (2) compatibility between **software requirements** and **preliminary design**, and
 - (3) on the preliminary version of the **operation** and **support documents**.

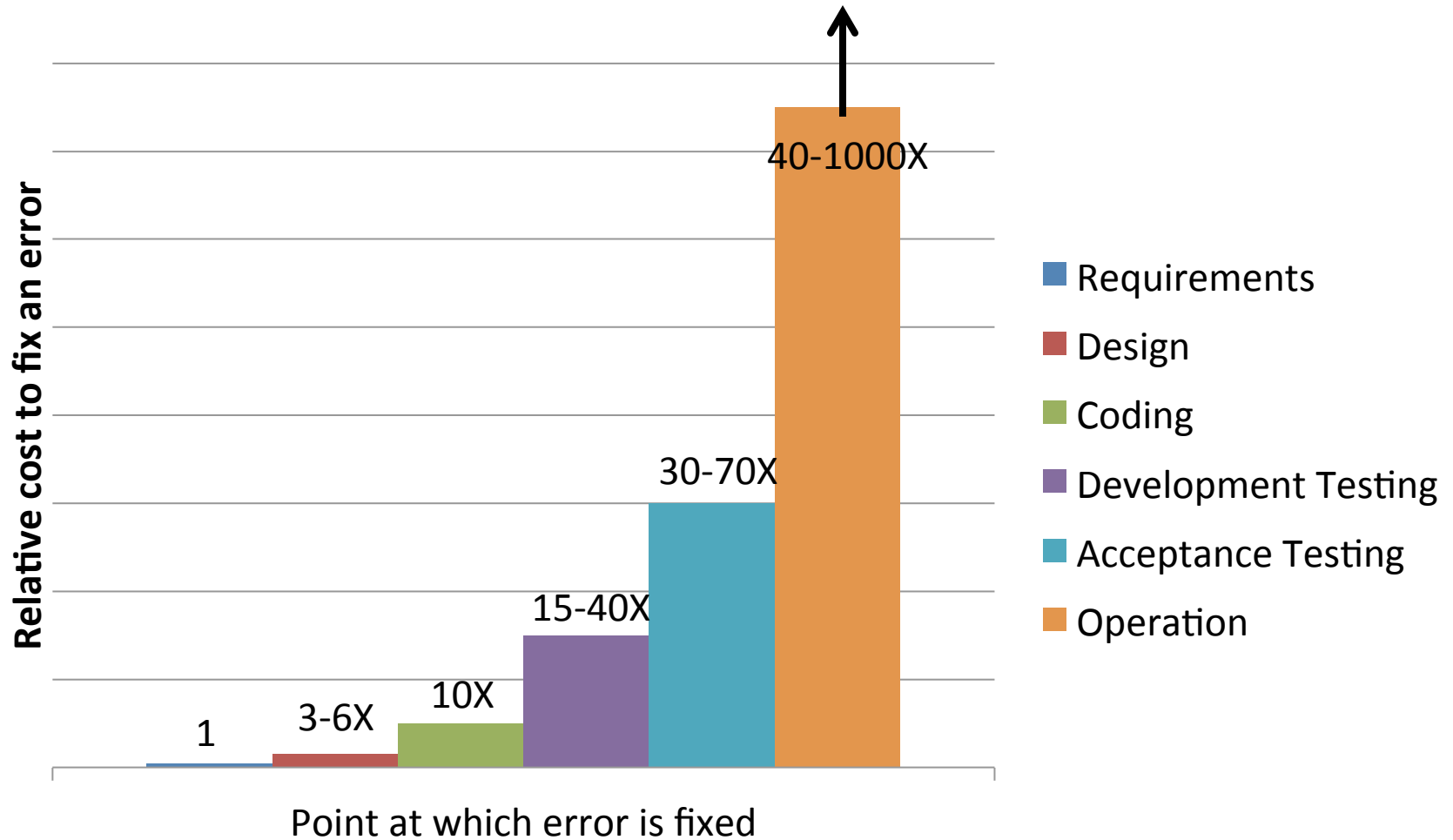
The Prevention Principle

Prevention is better than cure.

OR

An ounce of prevention is worth a
pound of cure.

Cost of Fixing Errors



Inspection

- IEEE **Definition** of Inspection
 - "... a **formal** evaluation technique in which software requirements, design, or code are examined in detail by a person or group other than the author to detect faults, violations of development standards, and other problems..."
- IEEE **Objective** of Inspection
 - "... to detect and identify software element defects. This is a rigorous, formal peer examination..."

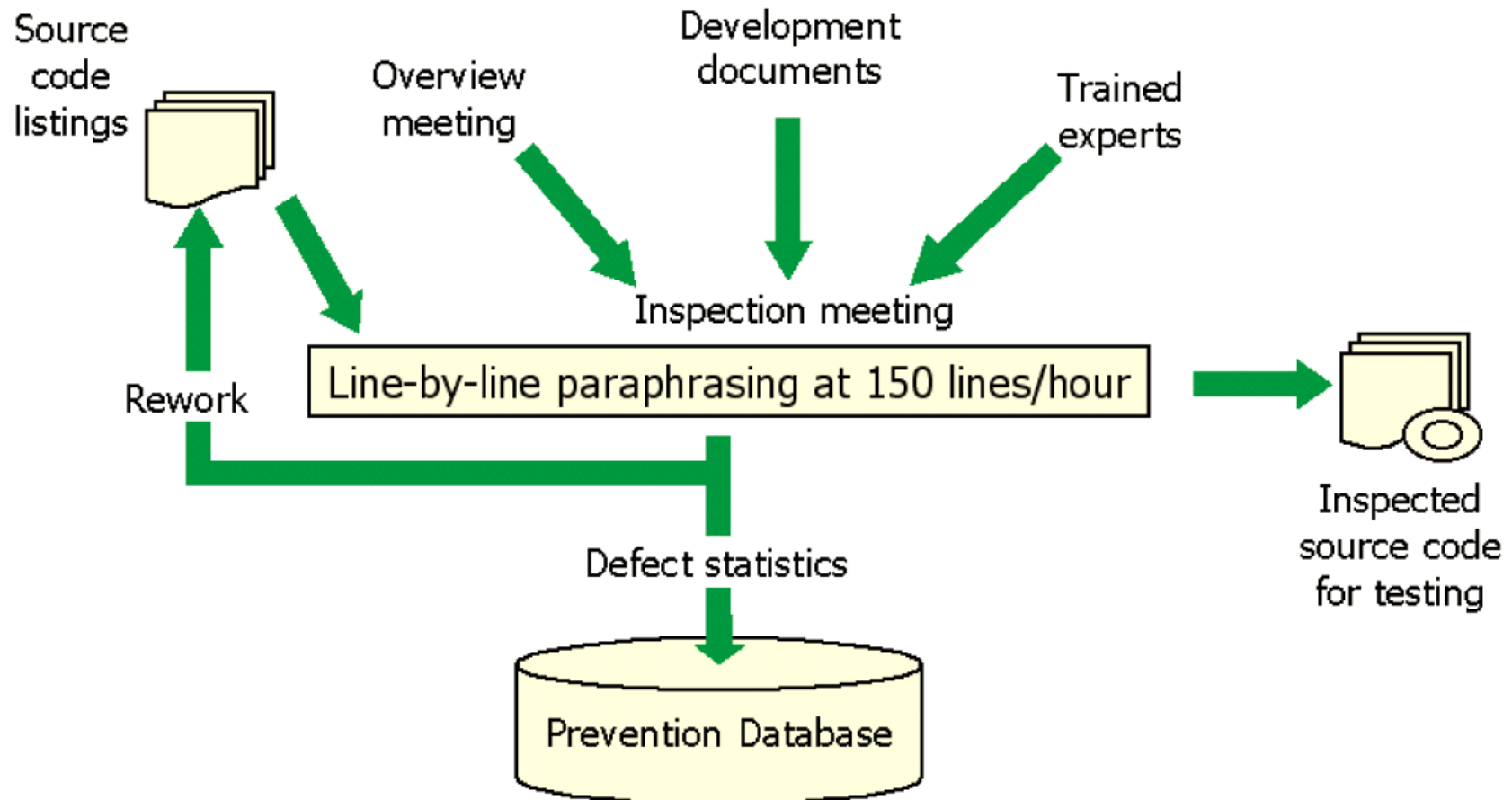
Inspection

- Verifies that the software elements **satisfy** its specifications
- Verifies that the software elements **conform** to applicable standards
- Identifies **deviations** from standards and specifications
- Collects software engineering **data** (for example, defect and effort data)
- Does not examine **alternatives** or **stylistic issues**

Inspection

- But **Inspection** (capital i) is a **formal process**!
 - One study found that **84%** of surveyed organizations performed reviews or inspections, but **0%** performed inspections entirely correctly
 - Even a walkthrough or a poorly done Inspection can be **effective** at improving software quality
 - Inspection is not only about **defect correction**, but also importantly about **defect prevention**

Fagan Inspections (e.g., for Code)



(Phillips, RMC 1999)

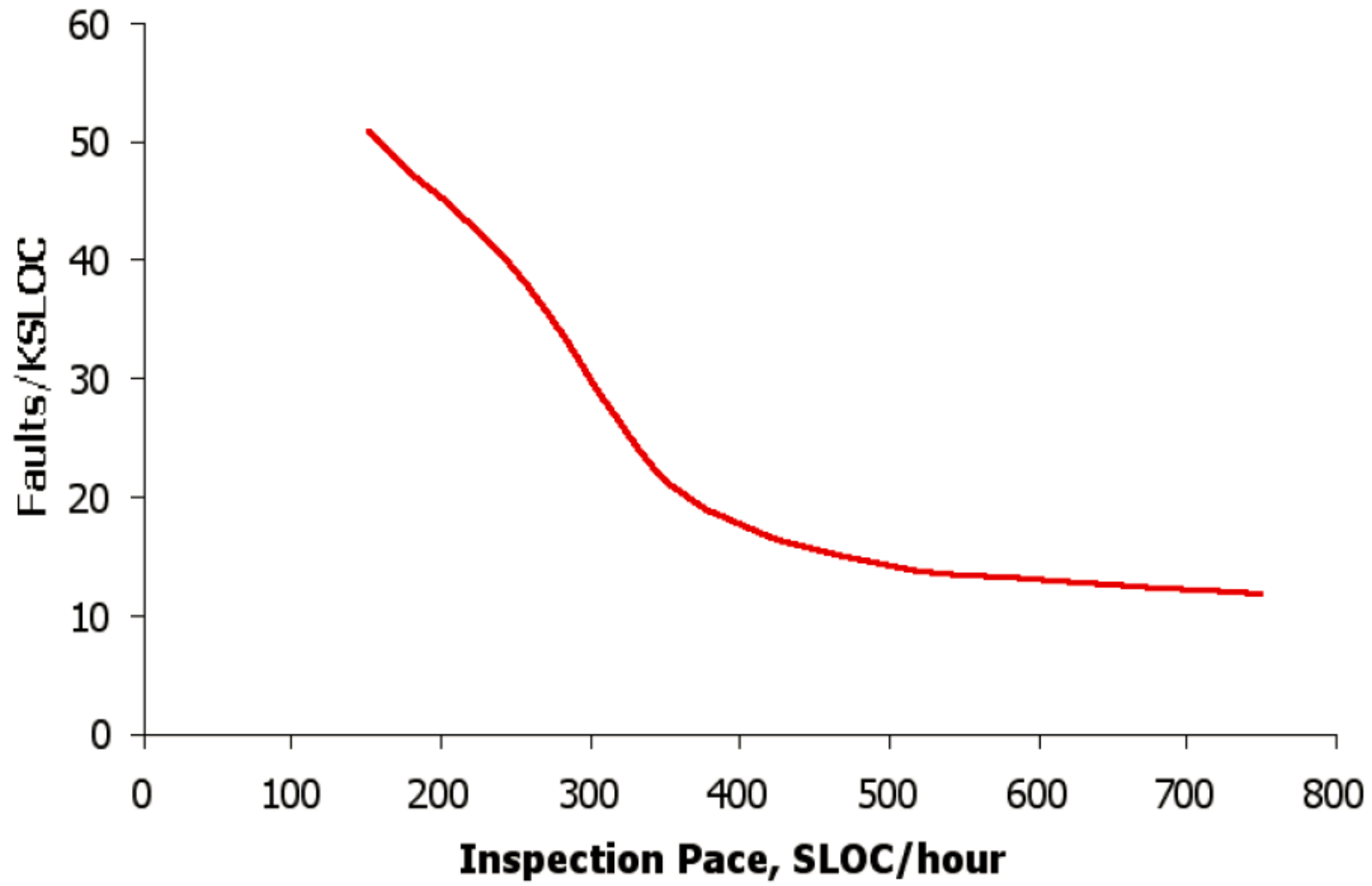
Inspection Roles (Fagan, Code Inspection)

- **Moderator**
 - Chairs the meeting, **records** faults found
 - Helps others stick to paraphrasing code, at the right **pace**
 - Keeps proceedings **objective**, professional, friendly
- **Inspectors (2 or 3)**
 - Knowledgeable **peers** who paraphrase the code, line by line
 - Must have attended **overview** meeting, reviewed **requirements** and **design** documents, must understand **context** of code
- **Author**
 - Silent **observer** who assists or clarifies only when asked

Choosing Inspectors (Fagan)

- **Good Choices**
 - Review specialists (e.g., **QA analysts**)
 - Technical people from the **same team** as author
 - Technical people with **special expertise** in subject matter of code
 - People with a special interest in the **product**
 - People from **other parts** of the org. or outside it
- **Bad Choices (exclude!)**
 - Managers, **supervisors**, or appraisers of the author
 - Anyone with a **personality clash** with the author or other reviewers
 - All **management**
 - Anyone with a **conflict of interest**

Inspection Efficiency

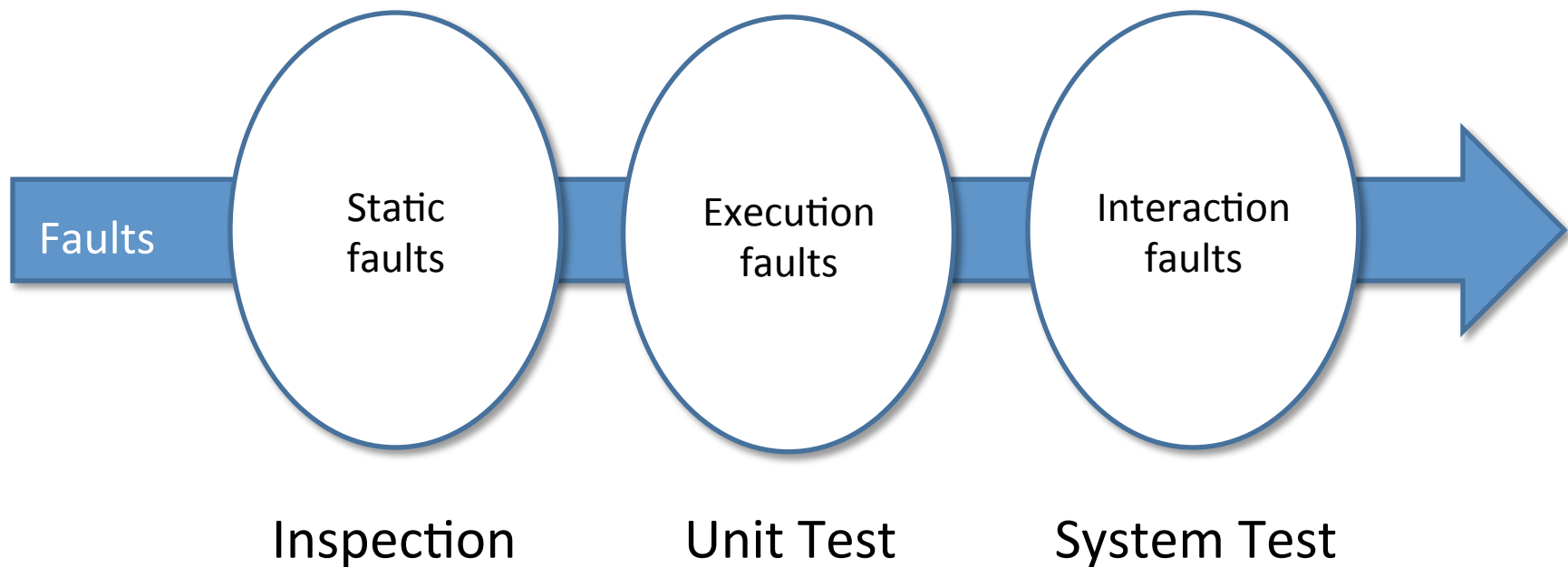


(Phillips, RMC 1999)

Side Benefits of Inspection

- **Cultural**
 - Team members gain a **broader perspective** on the software system as they review each other's work
 - Promotes a shared "**quality culture**", joint responsibility
- **Organizational**
 - Coding **standards** and practices are learned and **enforced**
 - **Consistency** improves
- **Educational**
 - Quality **improves** over time, as authors become more aware of the kinds of faults they are **prone to make**

Inspection in Context



Summary

- **Inspections, Walkthroughs, and Reviews**
 - Designed to catch faults **earlier** than possible using testing, to reduce costs and increase quality
 - Informal or formal meetings in which **reviewers** examine work of authors in detail
 - Very effective in practice
- **References**
 - Gilb & Graham, Ch. 3, "Overview of Software Inspection"
- **Next time**
 - Inspection processes