ILLINOIS TECH | College of Computing

ITMD 536 Software Testing & Maintenance

Nazneen Hashmi

Call or Text: 312-498-8387

IIT email: nhashmi@iit.edu

Time 10:00 AM to 12:40 PM

ILLINOIS TECH | College of Computing

ITMD 536 Software Testing & Maintenance

03 Static Techniques

ILLINOIS INSTITUTE OF TECHNOLOGY 1890

Objective

- What is Static Testing?
- What is Dynamic Testing?
- What is Formal Review Process?
- What is Software Technical Reviews?
- What are different types of review?
- What is Informal Review Process?
- What are the tools required for Static Analysis?

3. Static Techniques

- Helps improve the quality and productivity of software development
- Reviews, Requirements, Design, Implementation, Testing and Maintenance
- Detects earlier in SDLC
- Implements automated checks on code

3.1 Static Techniques and the Test Process

- The definition of testing outlines objectives that relate to evaluation, revealing defects and quality.

There are two different approaches to achieve it:

1. Static Testing
2. Dynamic Testing

3.1 Static Techniques and the Test Process

Static Testing:

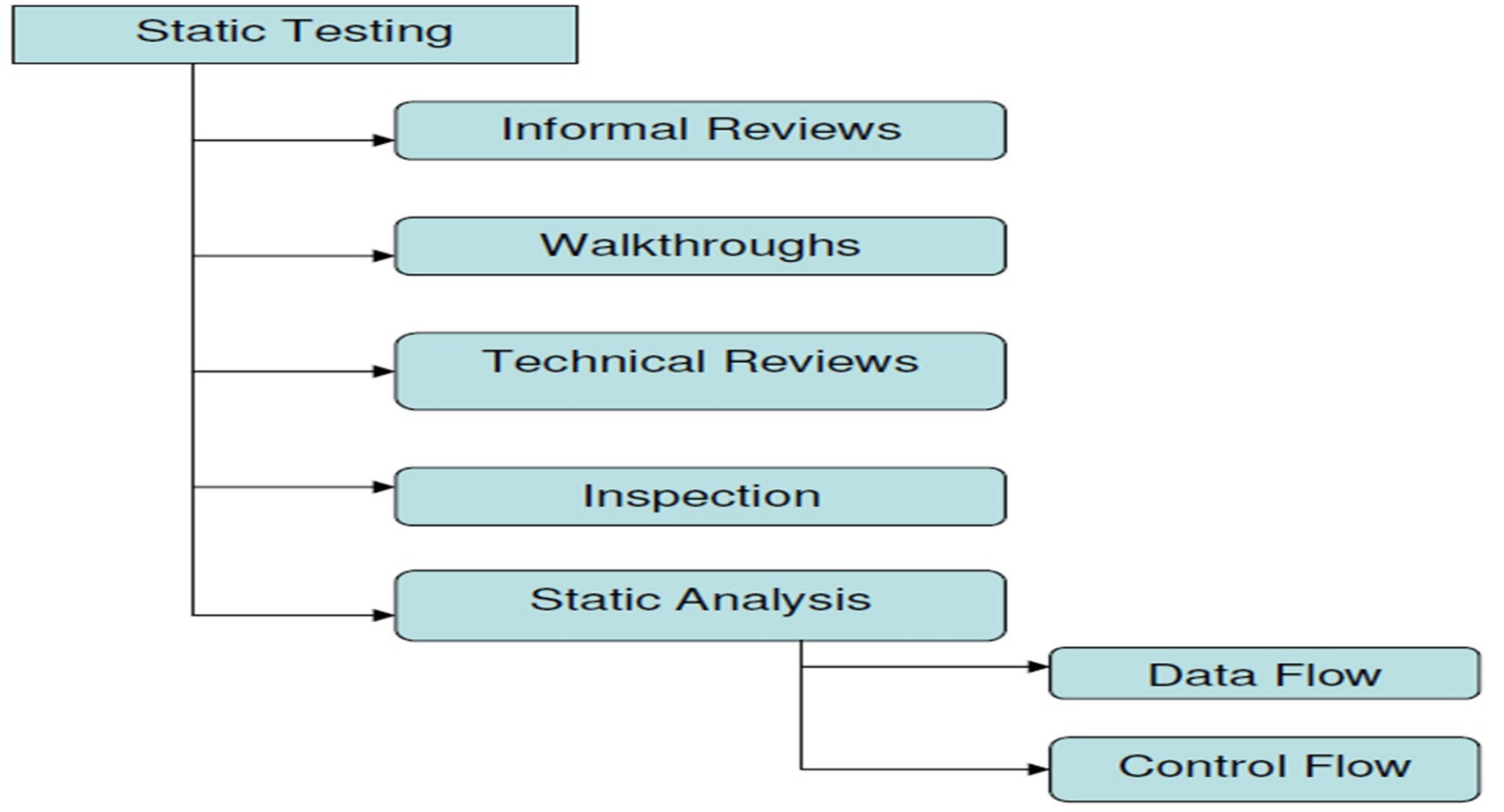
- Testing of a component or system at specification or implementation level without execution of that software, e.g. reviews or static analysis
- During static testing, software work products are examined manually, or with a set of tools, but not executed

Advantages of static testing:

- Can start earlier in life cycle, feedback on quality can be established early validation of requirements

3.1 Static Techniques and the Test Process

- Early detection of defects – will cost low to fix the defects
- Development is likely to increase
- Evaluation by the team with exchange of information helps between participants
- Increase awareness of quality issues


Static Testing
Informal Reviews
Walkthroughs
Technical Reviews
Inspection
Static Analysis
Data Flow
Control Flow

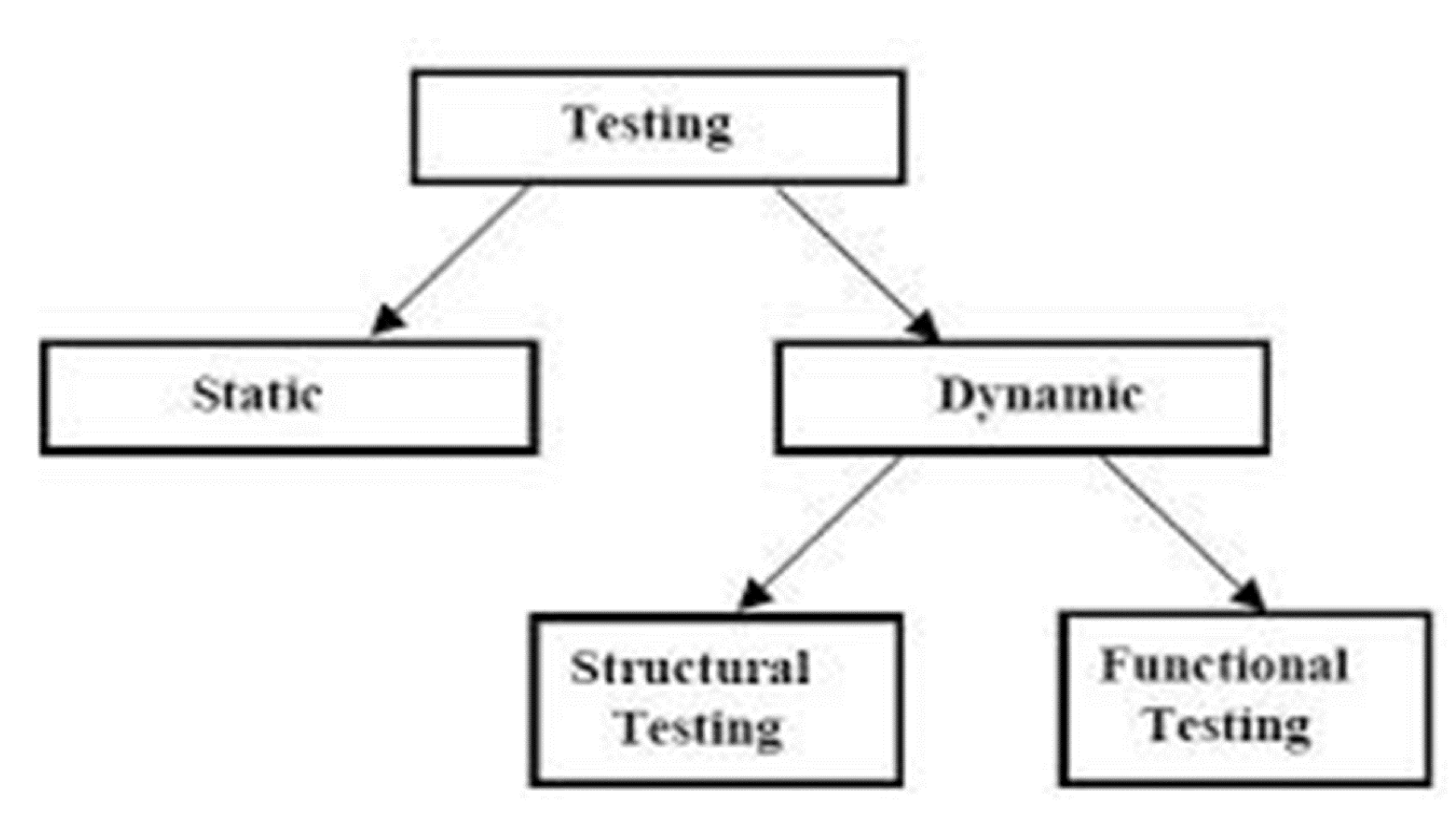
3.1 Static Techniques and the Test Process

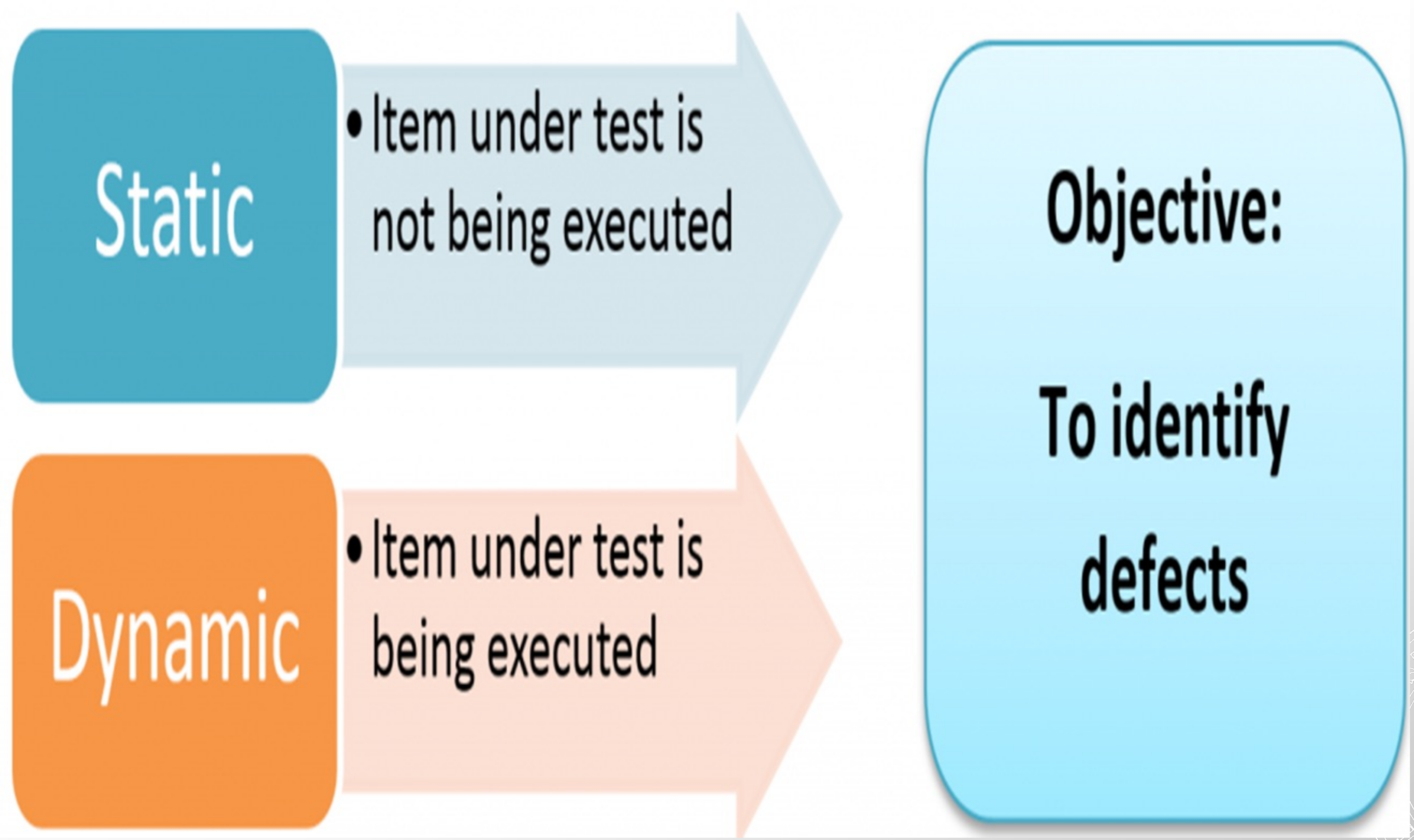
Dynamic Analysis: The process of evaluating behavior, e.g. memory performance, CPU usage, of a system or component during execution

Dynamic Analysis Tool: A tool that provides run-time information on the state of software code. These tools are most commonly used to identify unassigned pointers, check pointer arithmetic and to monitor the allocation, use and de-allocation of memory and to flag memory leaks.

3.1 Static Techniques and the Test Process

Dynamic Testing: Testing that involves the execution of the software of a component or system


Testing
Static
Dynamic
Structural Testing
Functional Testing



Static

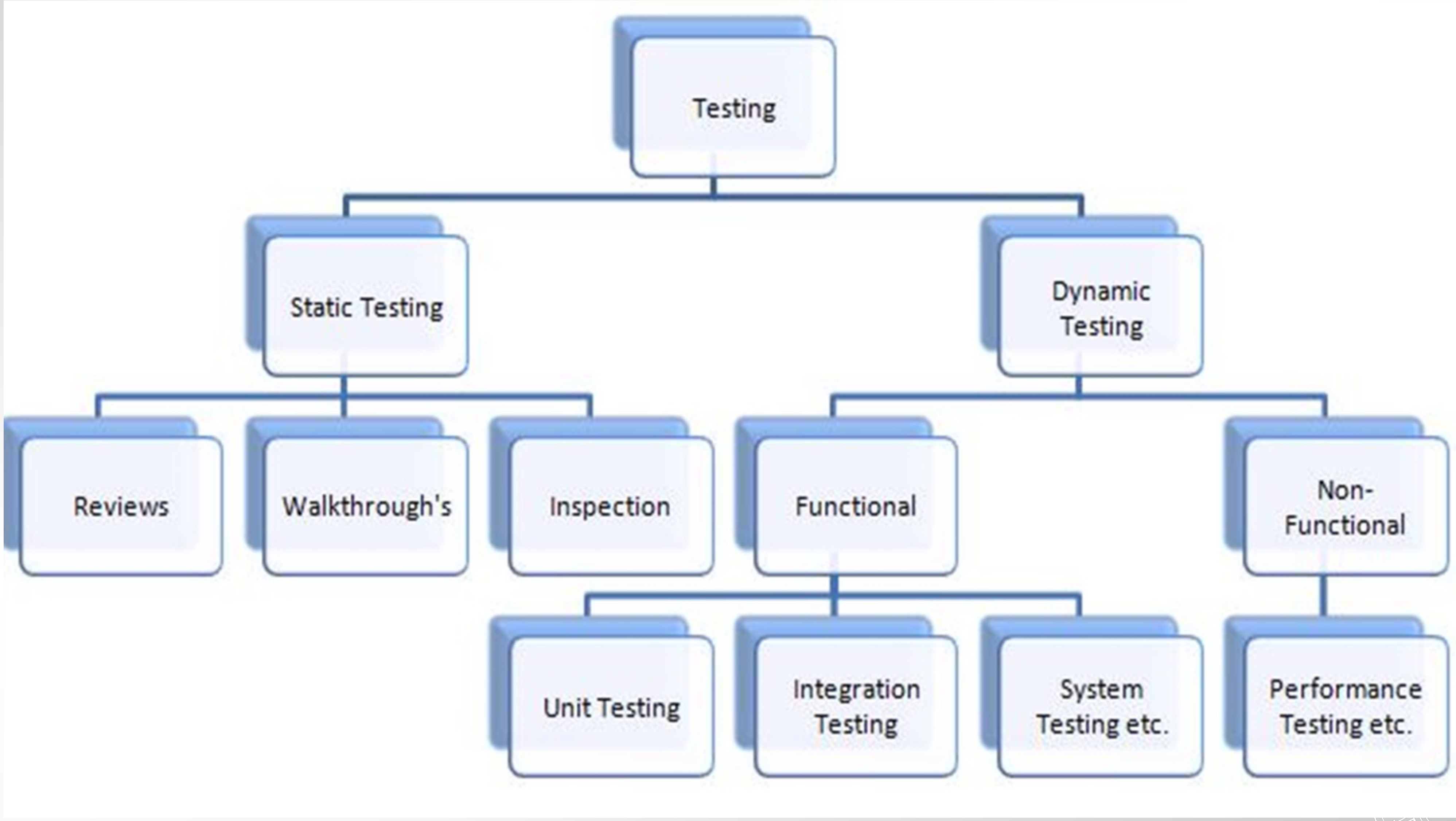
- Item under test is not being executed

Dynamic

- Item under test is being executed

Objective:

To identify defects



Testing

Static Testing

Dynamic Testing

Reviews

Walkthrough's

Inspection

Functional

Non-Functional

Unit Testing

Integration Testing

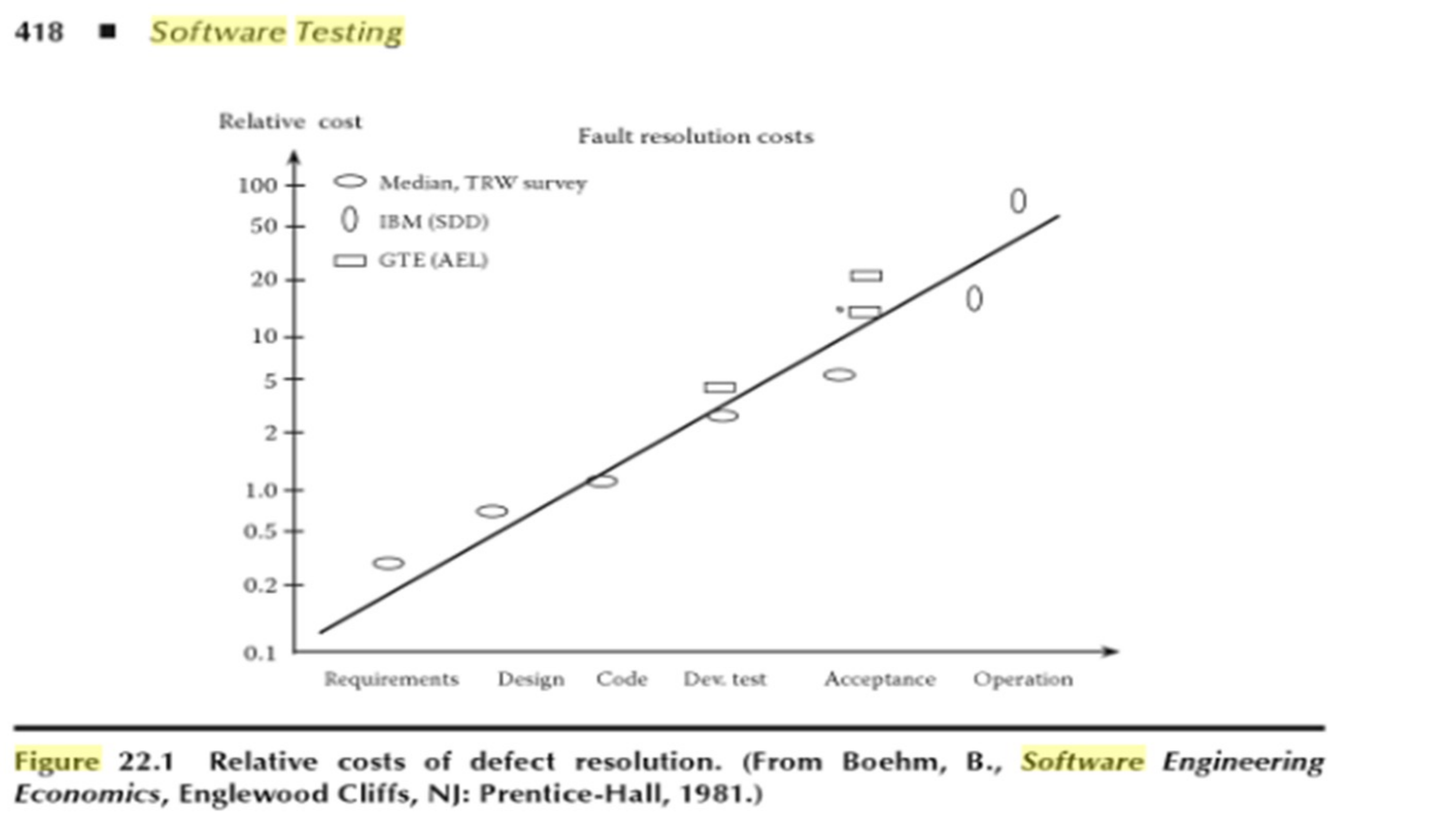
System Testing etc.

Performance Testing etc.

3.2 Review Process

Reviews vary from informal to formal

1. **Informal Review:** A review not based on a formal (documented) procedure
2. **Formal Review:** A review characterized by documented procedures and requirements , e.g. inspection



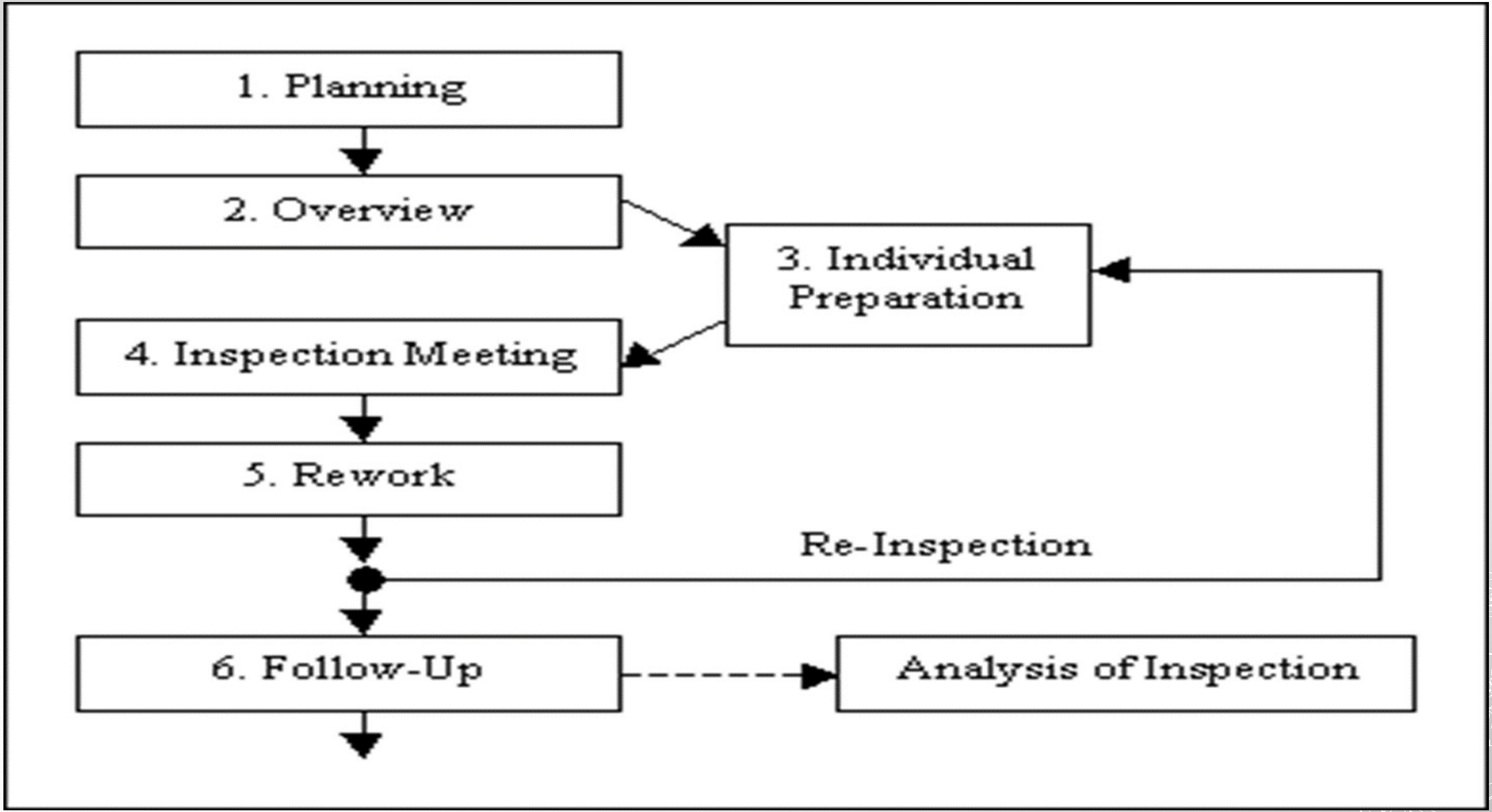
418 ■ *Software Testing*

Figure 22.1 Relative costs of defect resolution. (From Boehm, B., *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice-Hall, 1981.)

3.2.1 Phases of a formal review

A typical formal review process consist of six steps:

1. Planning
2. Kick-off
3. Preparation
4. Review meeting
5. Rework
6. Follow-up


1. Planning
2. Overview
3. Individual Preparation
4. Inspection Meeting
5. Rework
Re-Inspection
6. Follow-Up
Analysis of Inspection

3.2.1 Phases of a formal review

1. Planning:

- Define the review criteria
- Selecting the personnel
- Allocating roles
- Define the entry and exit
- Selecting which parts of the document to review
- Checking entry criteria

1. Planning

- **Review:** A review is requested by the author from the moderator (inspection leaders)
- **Moderator (inspection leader)**
- The leader and main person responsible for an inspection or other review process

Entry Criteria (Entry Check)

- The set of generic and specific conditions for permitting a process to go forward with a defined task, e.g. test phase. The purpose of entry criteria is to prevent a task from starting which would entail more (wanted) effort compared to the effort needed to remove the failed entry criteria

Entry Check

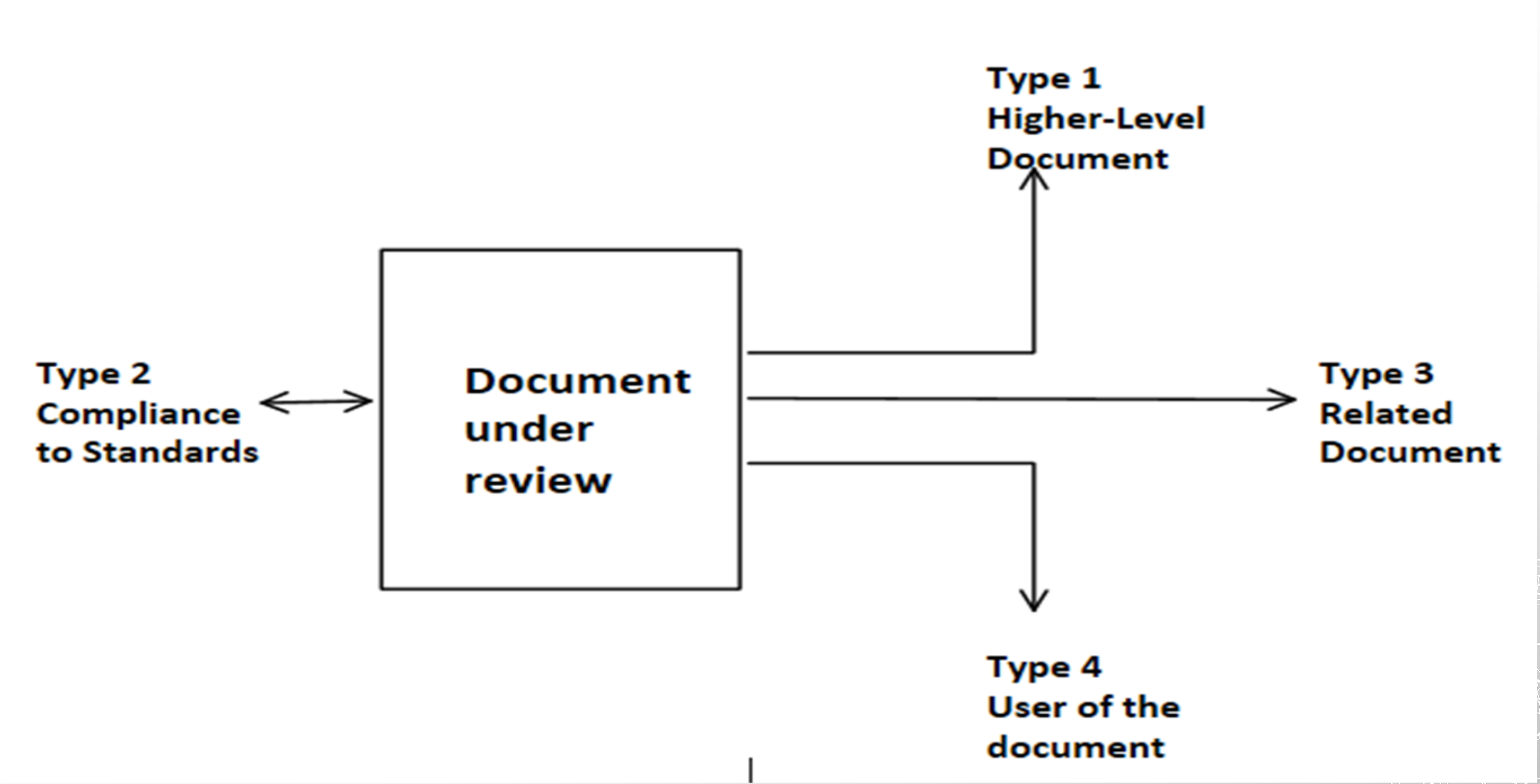
- Document to be reviewed is available with line numbers
- Document has been cleaned up by automated checks that apply
- References for inspection are stable and available
- Author is confident of the quality and is able to join the review team

Reviewer (inspector)

- The person involved in the review that identifies and describes anomalies in the product or project under review. Reviewers can be chosen to represent different viewpoints and roles in the review process

Different Roles within a Reviewer

- Focus on higher level – does the design comply to the requirement
- Focus on standards – consistency, clarity, naming conventions, templates
- Focus on related documents – interfaces between software functions
- Focus on usage of document – testability or maintainability


Type 1
Higher-Level
Document
Type 2
Compliance
to Standards
Document
under
review
Type 3
Related
Document
Type 4
User of the
document

2. Kick-off

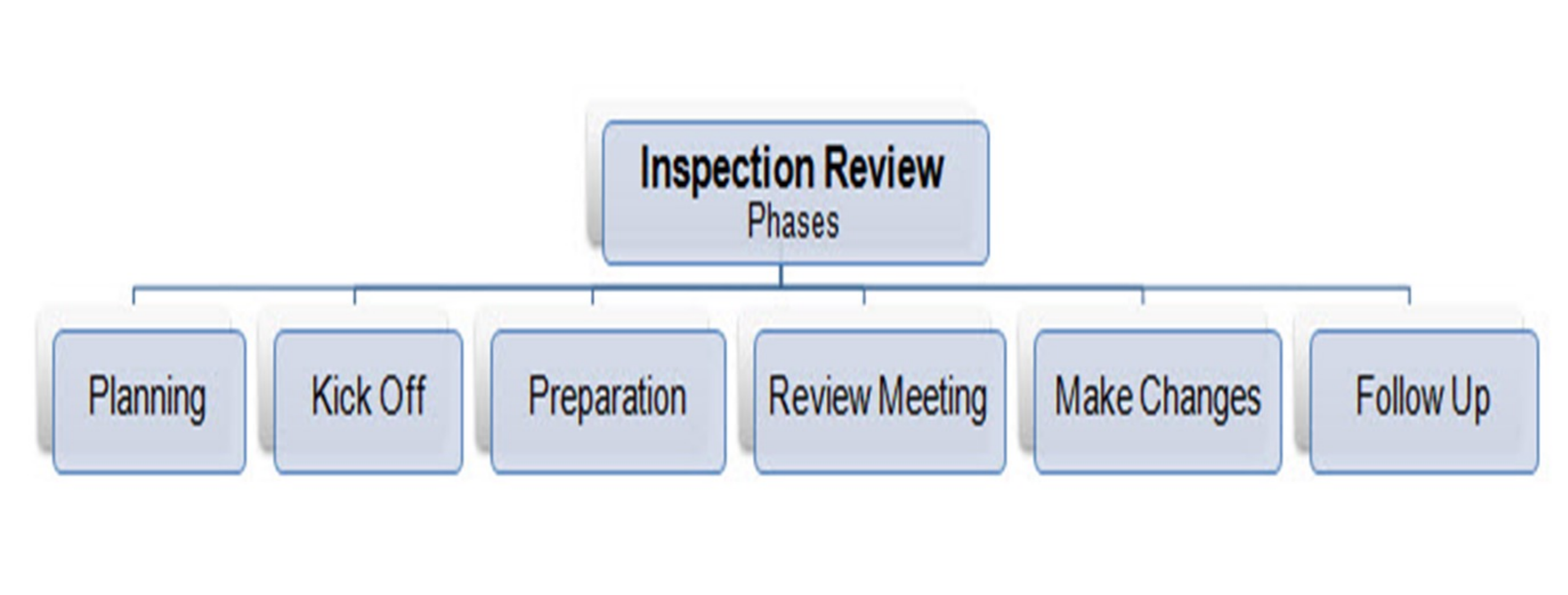
- Distributing documents
- Explaining the objectives, process and documents to the participants
- Kick-off meeting is to get everyone on the same page
- Reviewers receive a short introduction
- Role assignments, checking rate, the pages to be checked, process changes and other questions are discussed

kick-off meeting

Kick-off

- Prepare for the meeting by reviewing the document or documents
- Note the defects, questions and comments
- Review related documents, procedures, rules and checklists
- Use a checklist do document issues
- Annotated document will be given back

Formal Review Steps



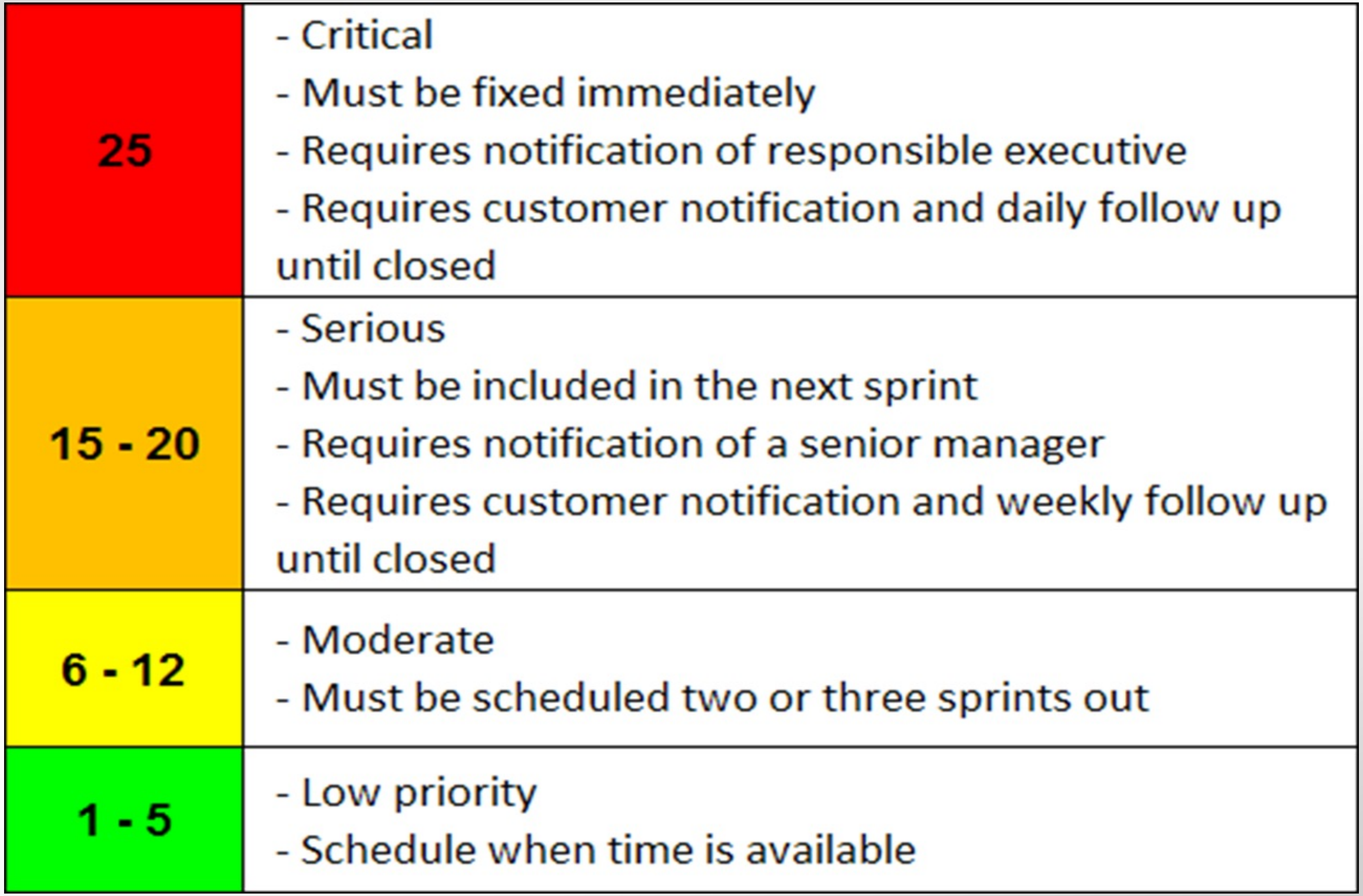
Recording Results from Formal Review

- Document results or minutes from formal review meeting
- Noting defects, making recommendations and decisions
- Examining, evaluating and recording issues during physical meetings or tracking any group electronic communications

Recording Results from Formal Review

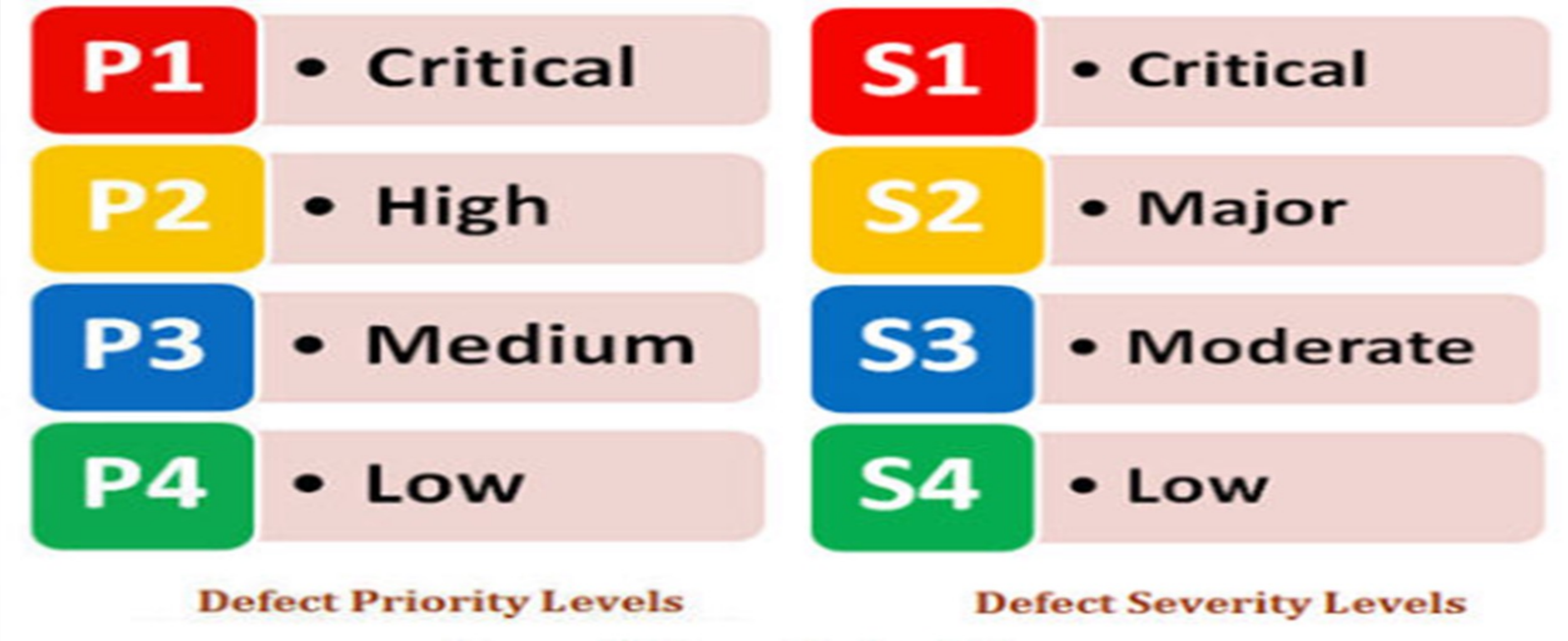
- Every defect and its severity should be logged

1. **Critical** – the scope and impact of the defect is beyond the document under inspection
2. **Major** – fault in design can result in an error in the implementation
3. **Medium** – must be fixed in couple sprints
4. **Minor** – non-compliance with standards



25	- Critical - Must be fixed immediately - Requires notification of responsible executive - Requires customer notification and daily follow up until closed
15 - 20	- Serious - Must be included in the next sprint - Requires notification of a senior manager - Requires customer notification and weekly follow up until closed
6 - 12	- Moderate - Must be scheduled two or three sprints out
1 - 5	- Low priority - Schedule when time is available

Current Defects Priority and Severity



P1	• Critical
P2	• High
P3	• Medium
P4	• Low

Defect Priority Levels

S1	• Critical
S2	• Major
S3	• Moderate
S4	• Low

Defect Severity Levels

Rework and Follow-up

Rework:

- Fixing defect found
- Recording updated status of defects

Follow-up:

- Checking the defects have been addressed
- Gathering metrics
- Checking exit criteria

Metrics

- A measurement scale and the method used for measurement
- Moderator tracks number of defects found, defects found per page, total review efforts
- This information is correct and stored for future analysis

Metrics

3.2.2 Roles and Responsibilities

There are four types of participants in review process:

1. Moderator
2. Author
3. Scribe
4. Reviewer

3.2.2 Roles and Responsibilities

Moderator: The leader and main person responsible for an inspection or other review process

- Performs the entry check and follow-up on the rework.
- Schedules meeting, disseminates documents before the meeting leads the discussion and writes the report

3.2.2 Roles and Responsibilities

The Author (Producer): The person who created the document

- The author's task is to illuminate unclear areas and to understand the defects found
- The author resolves the action items identified during the review meeting

3.2.2 Roles and Responsibilities

- **The Scribe (Recorder):** The person who records each defect mentioned and any suggestions for process improvement during a review meeting, on a logging form. The scribe should ensure that the logging form is readable and understandable

3.2.2 Roles and Responsibilities

The Scribe (Recorder): Takes notes during the review meeting. The recorded notes will be the basis for the formal review report. Last few minutes go over the notes to see if anything was missed

3.2.2 Roles and Responsibilities

- **Reviewer (Checkers or Inspectors):** The person involved in the review that identifies and describes anomalies in the product or project under review. Reviewers can be chosen to represent different viewpoints and roles in the review process

3.2.2 Roles and Responsibilities

The tasks of reviewers is to check prior to the meeting:

- Reviewer Name
- Reviewer Preparation Time
- A list of issues with severity
- An overall review recommendations, standards and checklists (manager decides on execution)

3.2.3 Types of Review

There are four types of review:

1. Walkthrough
2. Technical Review
3. Peer Review
4. Inspection

3.2.3 Types of Review

1. **Walkthrough:** A step-by-step presentation by the author of a document in order to gather information and to establish a common understanding of its content.

- The content of the document is explained step by step by the author, to reach consensus or to gather information

Walkthrough

- To present the document to the stakeholders
- To explain and evaluate the contents
- To establish the common understanding
- To examine and discuss validity and use the proposed alternatives
- Scenarios and dry run used to validate

3.2.3 Types of Review

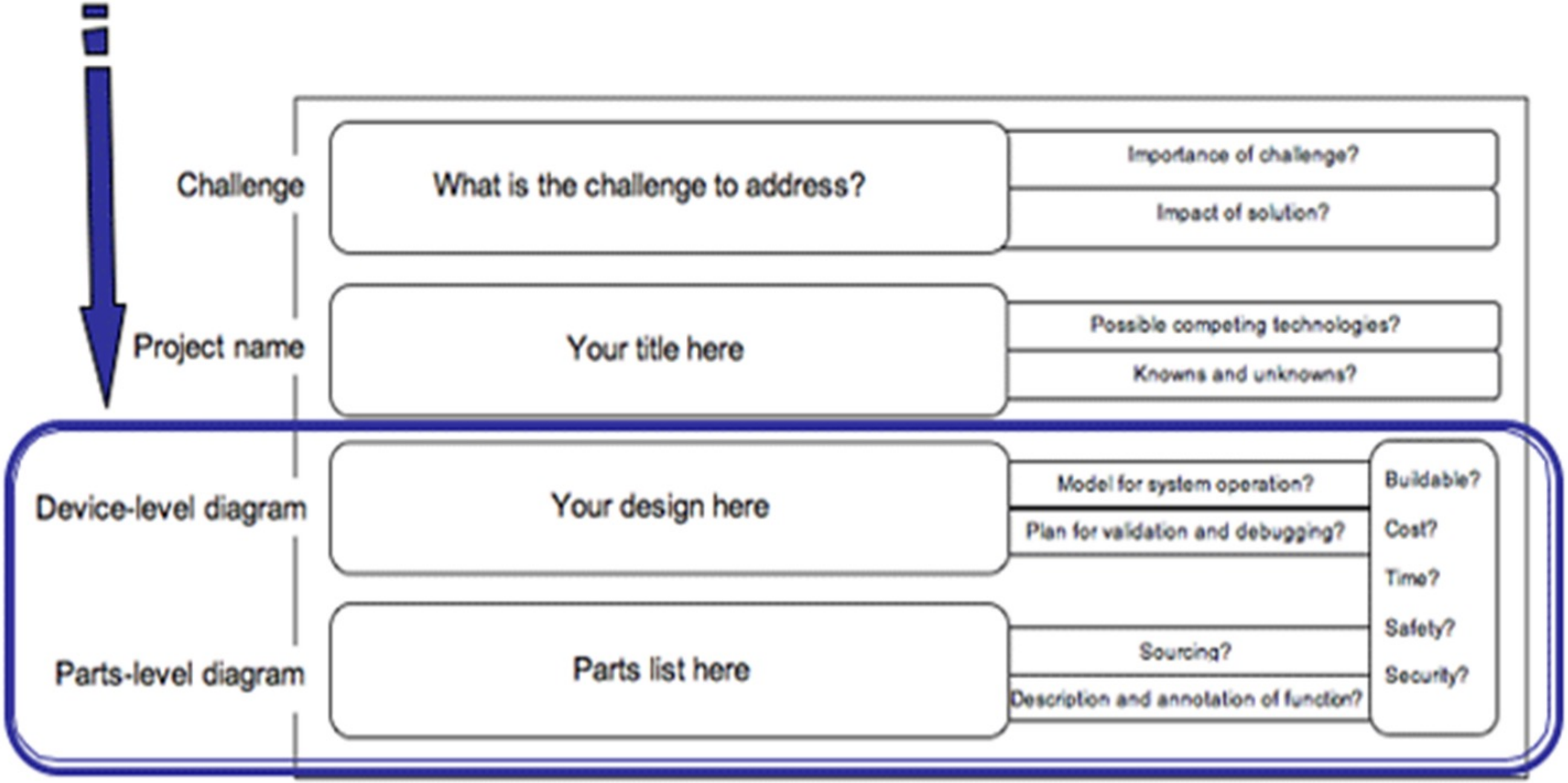
2. Technical Review: A peer group discussion activity that focuses on the achieving consensus on the technical approach to be taken

- During technical reviews defects are found by experts, who focus on the content of the document
- Architects, chief designers are included

Technical Review

- A documented inspection process
- Formal review training
- Establish consistency
- Budget review and preparation
- Sufficient lead time
- Checklist
- “Buy in” by everyone

Focus of your Technical Specification Review



Challenge

What is the challenge to address?

Importance of challenge?

Impact of solution?

Project name

Your title here

Possible competing technologies?

Knowns and unknowns?

Device-level diagram

Your design here

Model for system operation?

Plan for validation and debugging?

Parts-level diagram

Parts list here

Sourcing?

Description and annotation of function?

Buildable?

Cost?

Time?

Safety?

Security?

3.2.3 Types of Review

- **3. Peer Review:** A review of a software work product by colleagues of the producer of the product for the purpose of identifying defects and improvements. Examples are inspection, technical review and walkthrough

Peer Review

- Key characteristics of technical review are:
- Documented defect-detection process that involves peers and technical experts
- Lead by trained moderator
- Formal checklist are optional

3.2.3 Types of Review

- **4. Inspection (Audits):** A type of peer review that relies on visual examination of documents to detect defects, e.g. violations of development standards and non-conformance to higher level documentation. The most formal review technique and therefore always based on a documented procedure

Inspection (Audits)

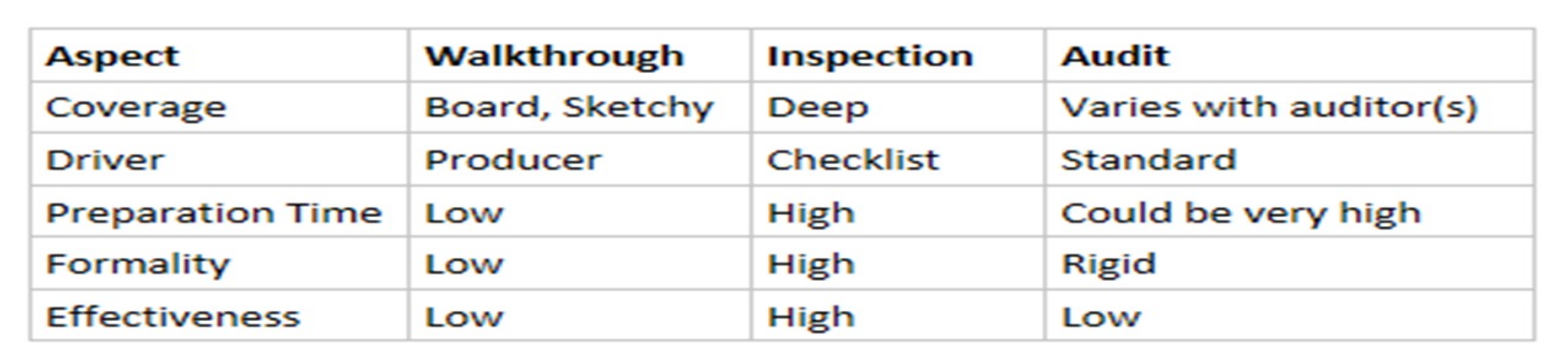
- Inspection (Audit) is the most formal review type
- Usually conducted by the software quality assurance group
- Help the author to improve
- Remove defects efficiently as early as possible

Inspection (Audits)

- Improve product quality
- Create common understanding
- Train new employees
- Learn from defects
- Rules and checklists are used
- Metrics are gathered and analyzed to optimize the process

Comparison of Review Types

- Walkthrough
- Inspection
- Audit



Aspect	Walkthrough	Inspection	Audit
Coverage	Board, Sketchy	Deep	Varies with auditor(s)
Driver	Producer	Checklist	Standard
Preparation Time	Low	High	Could be very high
Formality	Low	High	Rigid
Effectiveness	Low	High	Low

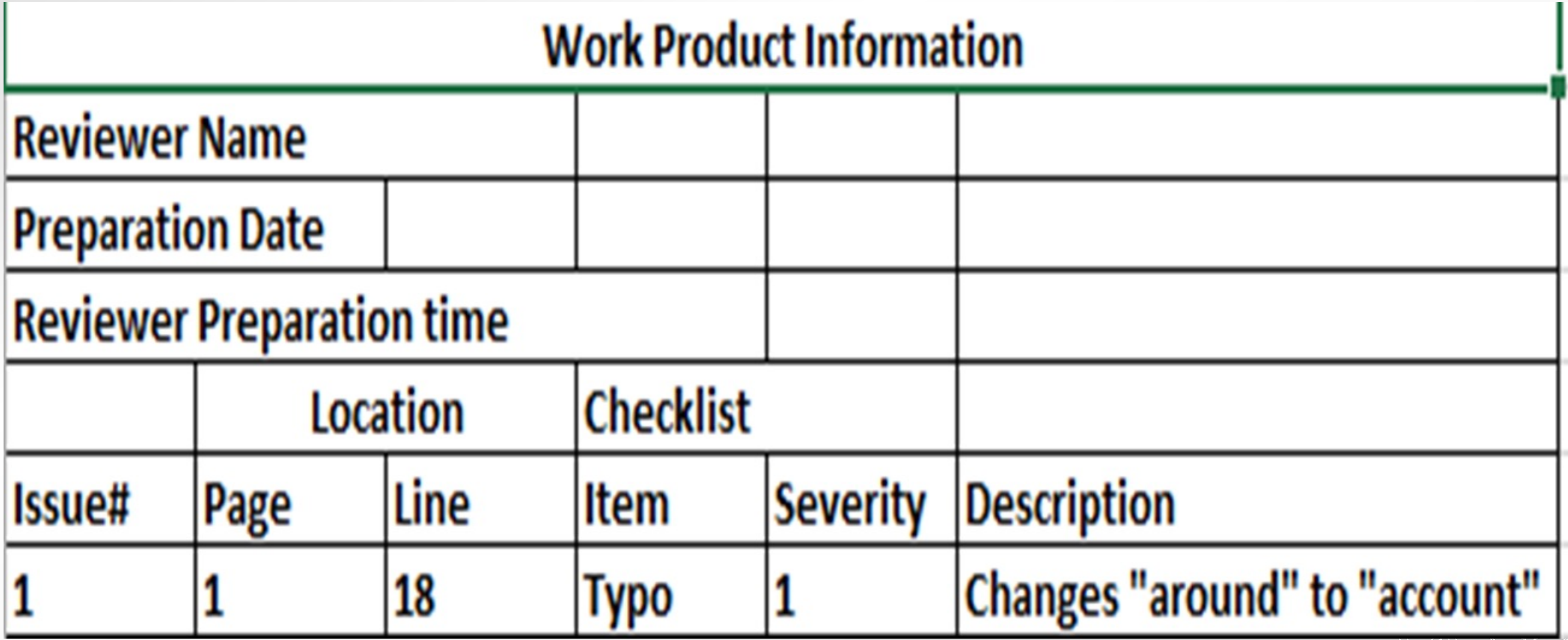
Contents of an Inspection Packet

- **Work Product Requirements** – what/how cycles are set for technical inspection – verify how part is accomplished.
- **Frozen Work Product** – each member receives full inspection packet.

Contents of an Inspection Packet

- **Standards and Checklists** – identifies the kinds of problems that a reviewer should look for.
- **Review Issues Spreadsheet** – Individual reviewers identifies issues and submit them to the review leader.

Review Issues Spreadsheet

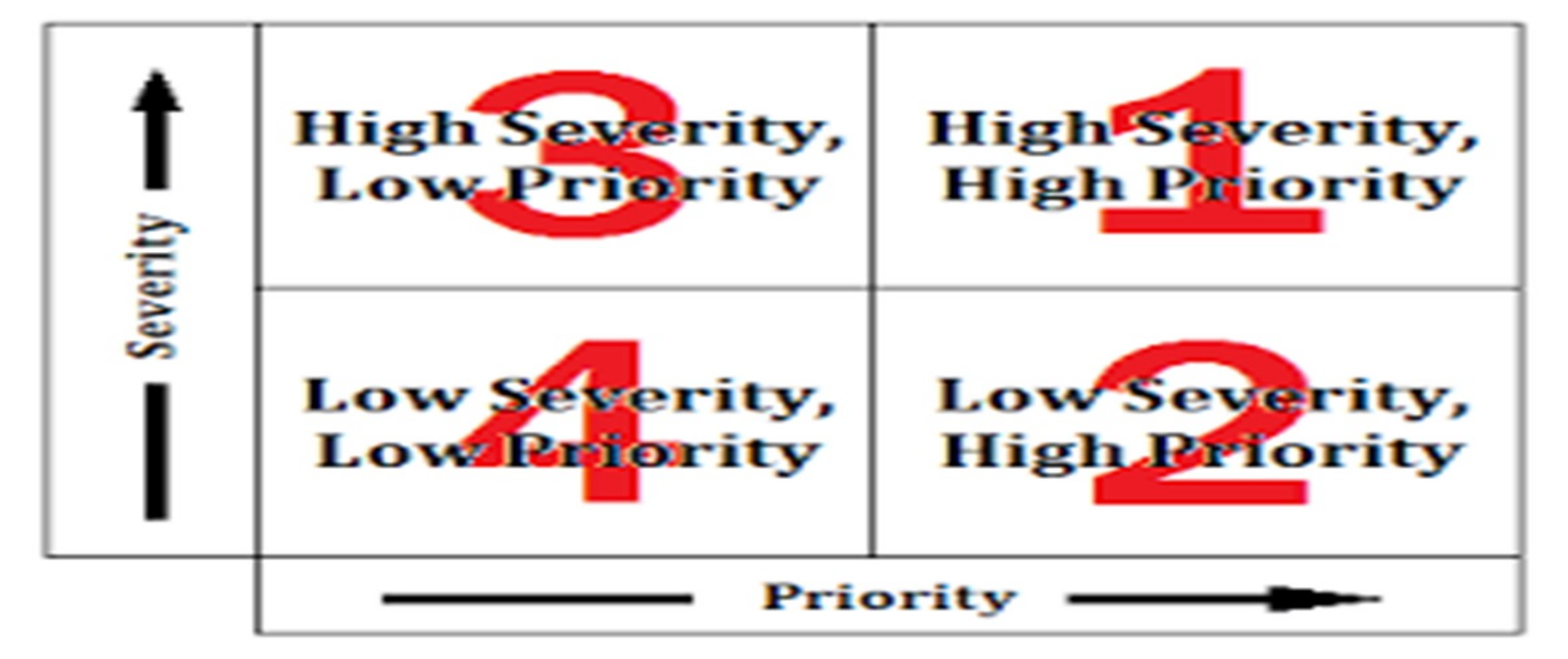


Work Product Information					
Reviewer Name					
Preparation Date					
Reviewer Preparation time					
	Location		Checklist		
Issue#	Page	Line	Item	Severity	Description
1	1	18	Typo	1	Changes "around" to "account"

Review Reporting Forms

- Individual reviewers complete and submit their individual reports to the review leader.
- Reviewer leader combines all the reviews and lists:
- Showstopper issues
- OK as is, minor or major rework needed

Fault Severity Levels Priority/Severity



3 High Severity, Low Priority	1 High Severity, High Priority
4 Low Severity, Low Priority	2 Low Severity, High Priority

Review Report Outline

1. Introduction
   - Work product identification
   - Review team members and roles
2. Preliminary issue list
   - Potential fault
   - Severity
3. Prioritized action item list
   - Identified faults
   - Severity

Review Report Outline

4. Summary of individual reports
5. Review statics
   - Total hours spent
   - Faults sorted by severity
   - Faults sorted by location

- Review recommendations
- Appendix with the full review packet

3.2.4 Success Factors for Reviews

- Here are the critical success factors:
- Find a “Champion”
- Pick things that really count
- Pick the right technique
- Explicitly plan and track review activities
- Train participants

3.2.4 Success Factors for Reviews

- Manage people issues
- Follow the rules but keep it simple
- Continuously improve process and tools
- Report results
- User testers
- Just do it

Find a 'Champion'

Champion has expertise, enthusiasm and a practical mindset in order to guide moderators and participants.

- Authority should be clear and management support is essential for success.
- Schedule adequate time for review.

Pick things that really count

- Select documents that are highly important.
- Review highly critical requirements.
- The invested hours will have a clear and high return on investment.

Pick the Right Techniques

- Consider the type, importance, and risk level of the work product to be reviewed, and the reviewers who will participate.
- Make sure each review has a clear objective and the correct type of review is selected that matches the defined objective.

Explicitly plan and tack review activities

- Review hours should be tracked and made visible.
- Tracking hours will improve planning for the next review.
- Management plays and important part in planning of review activities.

Train Participants

- Training participants is crucial for the success of the project.
- Special training should be provided to the moderators to prepare them for their critical role in the review process.

Manage People Issues

- Reviews are about evaluating someone's document.
- People issues and psychological aspects should be handled tactfully by the moderator.
- Defects should be welcomed and expressed objectively.
- Work and operate in trusted atmosphere

Follow the Rules but keep it Simple

- Make the process only as formal as the project culture or maturity level allows.
- Do not become too theoretical or too detailed.
- Checklists and roles are recommended to increase effectiveness of defect identification.

Continuously Improve Process and Tools

- Checklist based upon the ideas of participants, ensures the motivation of the developers/engineers.
- Motivation is the key to a successful change process.
- Emphasis on continuous learning and process improvement

Report Results

- Report quantified results and benefits to all those who are involved.
- Discuss the consequences of defects if they had not been found this early.
- Costs should be tracked.
- Quantify the benefits of the costs.

User Testers

- Users provide valuable input to the review.
- Testers who participate in reviews learn about the product.

Just do it!

- The process is simple but not easy.
- Every step of the process is clear, but experience is needed to execute them correctly.
- Get the experience people on board to help where possible.
- Start doing reviews and start learning from every review.

3.3 Static Analysis by Tools

Static analysis – Analysis of software artifacts, e.g. requirements or code, carried out without execution of these software development artifacts. Static analysis is usually carried out by means of a supporting tool.

Static Analysis

- Performed on requirements, design or code without actually executing the software.
- Ideally performed before the types of formal review.
- The goal is to find defects

Static Analysis Tools

- Static analysis tools are typically used by the developers before, and sometimes during component and integration testing and by designers during software modeling.
- Compiler is a static analysis tool check for non-compliance to code language convention (syntax).

Compiler

Compiler: A software tool that translates programs expressed in a high order language into their machine language equivalents.

Defects – during static analysis of code

- Referencing a variable with a undefined value.
- Inconsistent interfaces between modules and components, such as wrong parameters etc.
- Improper declaration of variables or the declaration of variables that are never used.

Defects – during static analysis of code

- Unreachable (‘dead’) code that can safely be removed.
- Missing or erroneous logic or infinite loop.
- Highly complex functions.
- Standards violations – risks and failures.
- Security vulnerabilities.
- Syntax violations of code and software models.

3.3.1 Coding Standards

- Coding standard consists of a set of programming rules (check boundaries when copying an array)

3.3.2 Code Metrics

- Experienced programmers know that 20% of the code will cause 80% of the problems.
- Complexity analysis helps to find all-important 20%, which relates back to the principle of defect clustering.

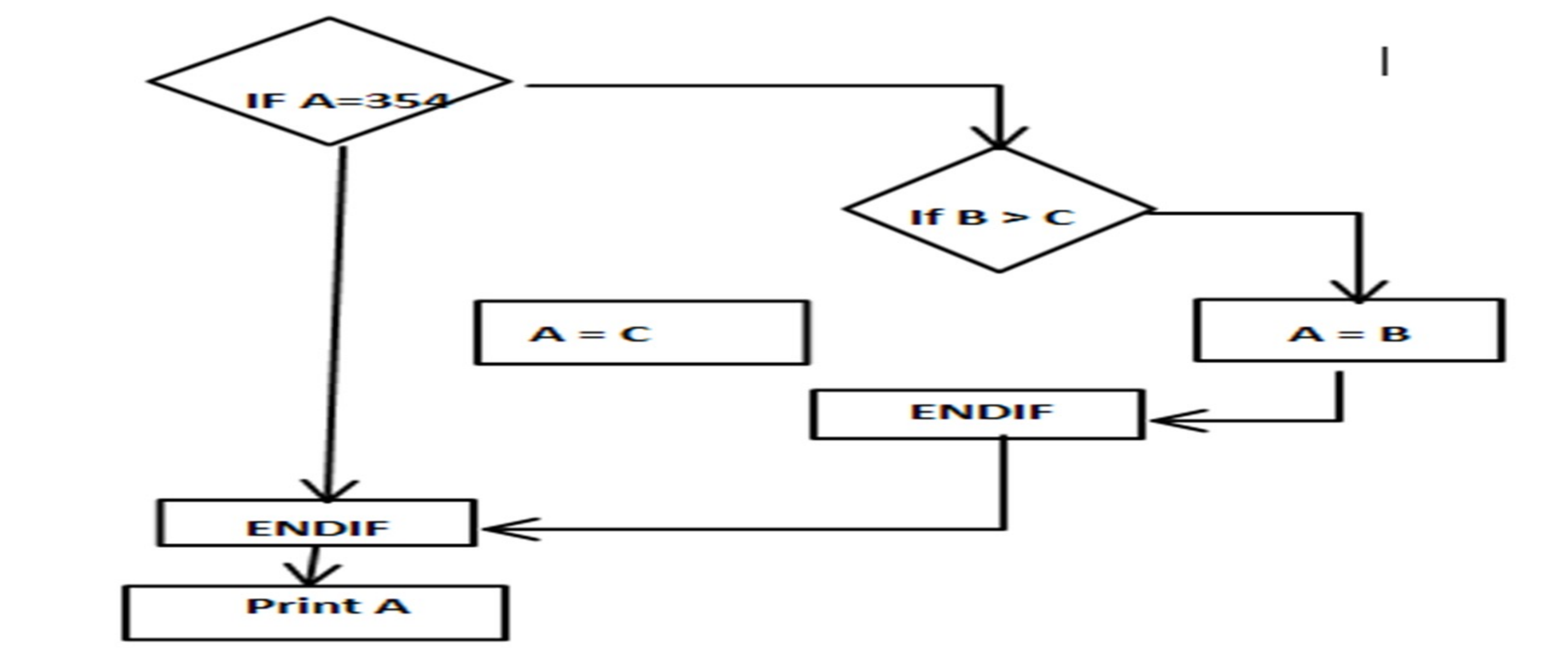
Complexity

- **Complexity –** The degree to which a component or system has a design and/or internal structure that is difficult to understand, maintain and verify.

Cyclomatic Complexity

- The number of independent paths through a program. Cyclomatic complexity is defined as:
- L=N+2P, where
- - L = the number of edges/links in a graph
- - N = the number of nodes in a graph
- + P = the number of disconnected parts of the graph(called graph or subroutine).

Control Flow of a Simple Program



3.3.3 Code Structure

- There are several aspects of code structure
  - Control Flow Structure
  - Data Flow Structure
  - Data Structure

Control Flow Structure

- The control flow structure addresses the sequence in which the instructions are executed.
- Control flow analysis can also be used to identify unreachable (dead) code. (cyclomatic complexity)

Data Flow

- **Data flow** An abstract representation of the sequence and possible changes of the state of data objects, where the state of an object is any of creation, usage or destruction.
- Data flow structure follows the trail of a data item as it is accessed and modified by the code.

Data Structure

- Data structure refers to the organization of the data itself, independent of the program.
- When data are arranged as a list, queue, stack, or other well-defined structure, the algorithms for creating, modifying or deleting them are more likely to be well-defined, too.

3. Static Techniques – Question 1

1. Which of the following artefacts can be examined by using review techniques?

a. Software code.

b. Requirements specifications.

c. Test design.

d. All of the above.

Question 2

2. Which statement about the function of a static analysis tool is true?

a. Gives quality information about the code without executing it.

b. Checks expected results against actual results.

c. Can detect memory leaks.

d. Gives information about what code has and had not been exercised.

Question 3

3. Which is not a type of review?

a. Walkthrough

b. Inspection

c. Informal review

d. Management approval

Question 4

4. What statement about reviews is true?

a. Inspections are led by a trained moderator, whereas technical reviews are not necessarily.

b. Technical reviews are led by a trained leader, inspections are not.

c. In a walkthrough, the author does not attend.

d. Participants for a walkthrough always need to be thoroughly trained.

Question 5

5. What is the main difference between a walkthrough and an inspection?

a. An inspection is led by the authors, while walkthrough is led by a trained moderator.

b. An inspection has a trained leader, while a walkthrough has no leader.

c. Authors are not present during inspections, while they are during walkthroughs

d. A walkthrough is led by the author, while an inspection is led by a trained moderator.

Question 6

6. Which of the following characteristics and types of review processes belong together?

1. Led by the author

2. Undocumented

3. No management participation

4. Led by a trained moderator or leader

5. Uses entry and exit criteria

    s. Inspection
    t. Technical review
    u. Informal review
    v. Walkthrough

a. s=4, t=3, u=2 and 5, v=1

b. s=4, and 5, t=3, u=2, v=1

c. s=1 and 5, t=3, u=2, v=4

d. s=5, t=4, u=3, v=1 and 2

Question 7

7. What statement about static analysis is true?

a. With static analysis, defects can be found that are difficult to find with dynamic testing.

b. Compiling is not a form of static analysis.

c. When properly performed, static analysis makes functional testing redundant.

d. Static analysis finds all faults.

Question 8

8. Which of the following statements about early test design are true and which are false?

1. Defects found during early test design are more expensive to fix.

2. Early test design can find defects.

3. Early test design can cause changes to the requirements.

4. Early test design take more effort.

a. 1 and 3 are true, 2 and 4 are false.

b. 2 is true, 1,3 and 4 are false.

c. 2 and 3 are true, 1 and 4 are false.

d. 2,3 and 4 are true, 1 is false

Question 9

9. Static code analysis typically identifies all but one of the following problems, which is it?

a. Unreachable code

b. Undeclared variable

c. Faults in the requirements

d. Too few comments