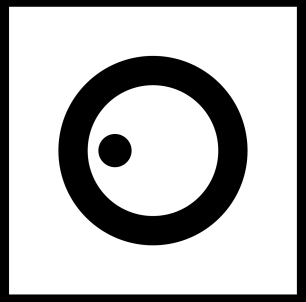


POLITÉCNICO
DE LEIRIA

ESCOLA SUPERIOR
DE TECNOLOGIA
E GESTÃO

Static Testing

José Carlos Bregieiro Ribeiro
jose.ribeiro@ipleiria.pt

Static Testing

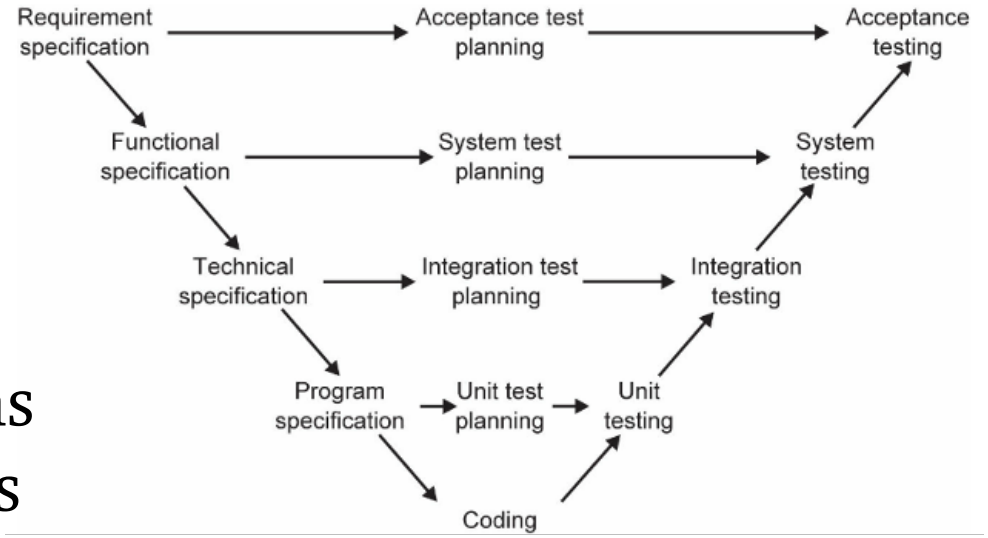
- **Definition of Static Testing**
  - Software testing method where the software is not executed
  - Evaluates work products (e.g., code, requirements) through manual or automated techniques

Static Testing

- **Key Characteristics:**
  - Can be done through manual examination (eg, reviews) or tools (eg, static analysis).
  - Test objectives include improving quality, detecting defects, and assessing characteristics like readability, completeness, correctness, testability and consistency.
  - Applied for both verification and validation.
  - Enables early defect detection.

Work Products Examinable by Static Testing

- **Examples:**
  - Requirement documents
  - Source code
  - Test plans and cases
  - Product backlog items
  - Contracts and models



- **Limitations:**
  - Work products must be interpretable
  - Some products (e.g., third-party executables) may not be analyzable

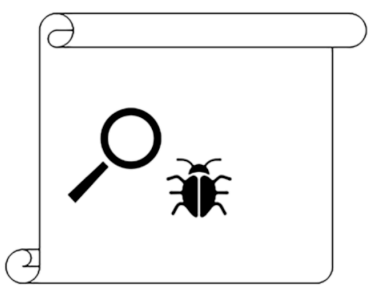
Value of Static Testing

- **Early defect detection:**
  - Reduces rework and cost
  - Improves quality
- **Detects issues missed by dynamic testing:**
  - Unreachable code
  - Design flaws
  - Defects in non-executable artifacts
- **Enhances communication:**
  - Stakeholder alignment
  - Clearer requirements

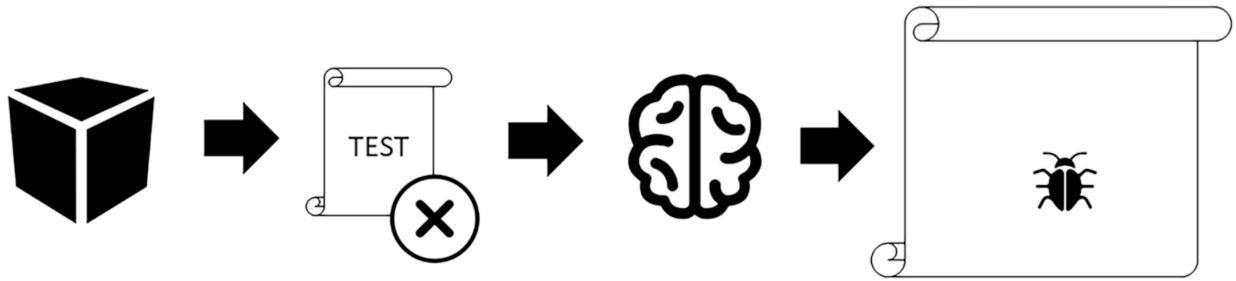
Differences Between Static and Dynamic Testing

POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

- **Static Testing:**
  - Finds defects directly
    (whereas dynamic testing finds failures and traces them to defects)
  - Identifies issues in non-executable products
    (whereas dynamic testing measures execution-dependent attributes)



Static testing: defect detection

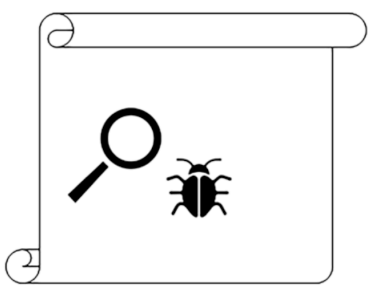


Dynamic testing: causing failure, analysis, defect localization

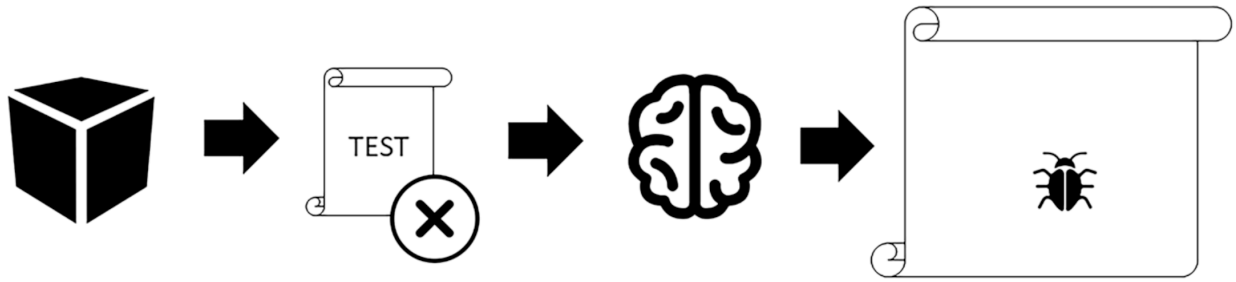
Differences Between Static and Dynamic Testing

POLITÉCNICO DE LEIRIA
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

- **Defects that are easier/cheaper to find through static testing:**
- Defects in requirements (e.g., inconsistencies, ambiguities, contradictions, omissions, inaccuracies, duplications)
- Design defects (e.g., inefficient database structures, poor modularization)
- Certain types of coding defects (e.g., variables with undefined values, undeclared variables, unreachable or duplicated code, excessive code complexity)
- Deviations from standards (e.g., lack of adherence to naming conventions in coding standards)
- Incorrect interface specifications (e.g., mismatched number, type or order of parameters)
- Specific types of security vulnerabilities (e.g., buffer overflows)
- Gaps or inaccuracies in test basis coverage (e.g., missing tests for an acceptance criterion)



Static testing: defect detection



Dynamic testing: causing failure, analysis, defect localization

Differences Between Static and Dynamic Testing

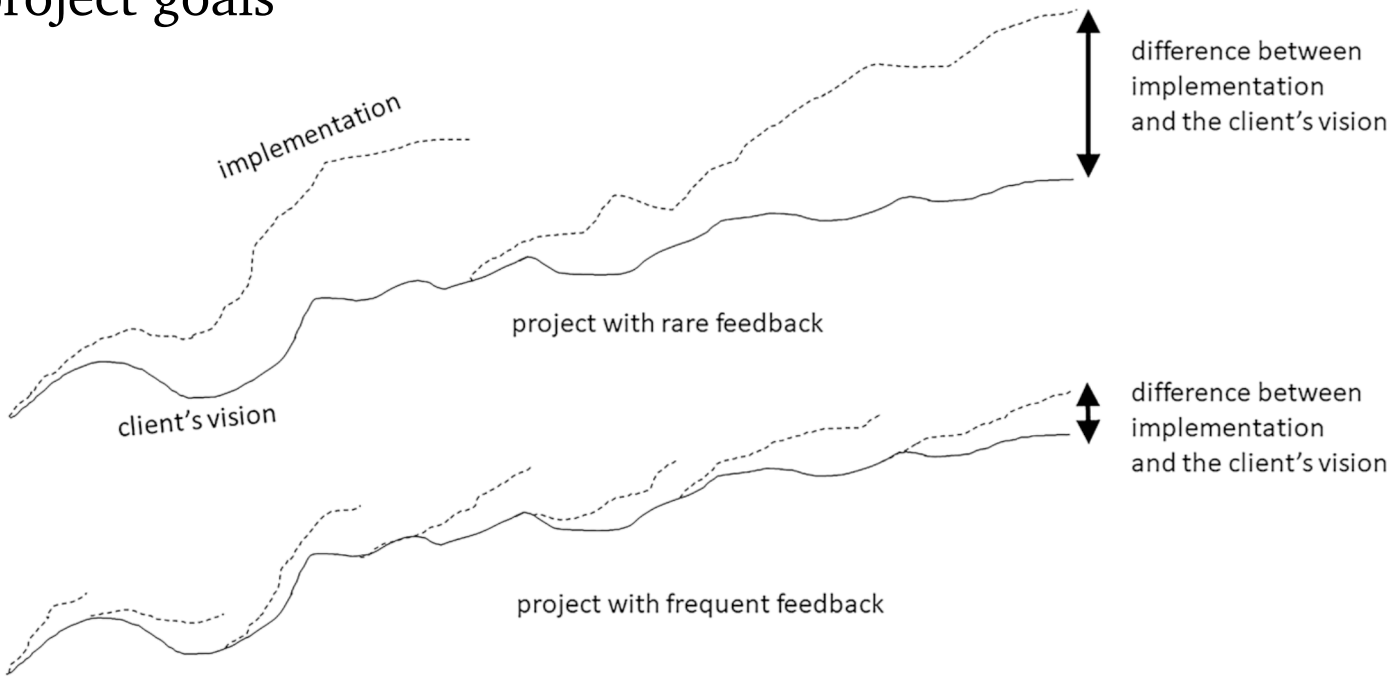
- **Example**: The testing team wants to evaluate the ease and cost of maintaining the software after its release to the customer. To do so, it intends to use static analysis techniques.
- Three metrics have been identified that will be used to measure the source code:
  - LOC -- the number of executable lines of code in a component
  - COMMENT -- the percentage of annotated lines of code in a component
  - CC -- cyclomatic complexity of a component

- The static analysis led to the following conclusions:
  - The components are rather small.
  - The code is not commented enough.
  - Some components have high cyclomatic complexity.

Component	LOC	COMMENT	CC
`main`	129	0%	7
`split`	206	1%	12
`to_lower`	62	1%	6
`to_upper`	63	3%	6
`merge`	70	0%	15
`config`	42	15%	8
`stdio`	243	2%	21

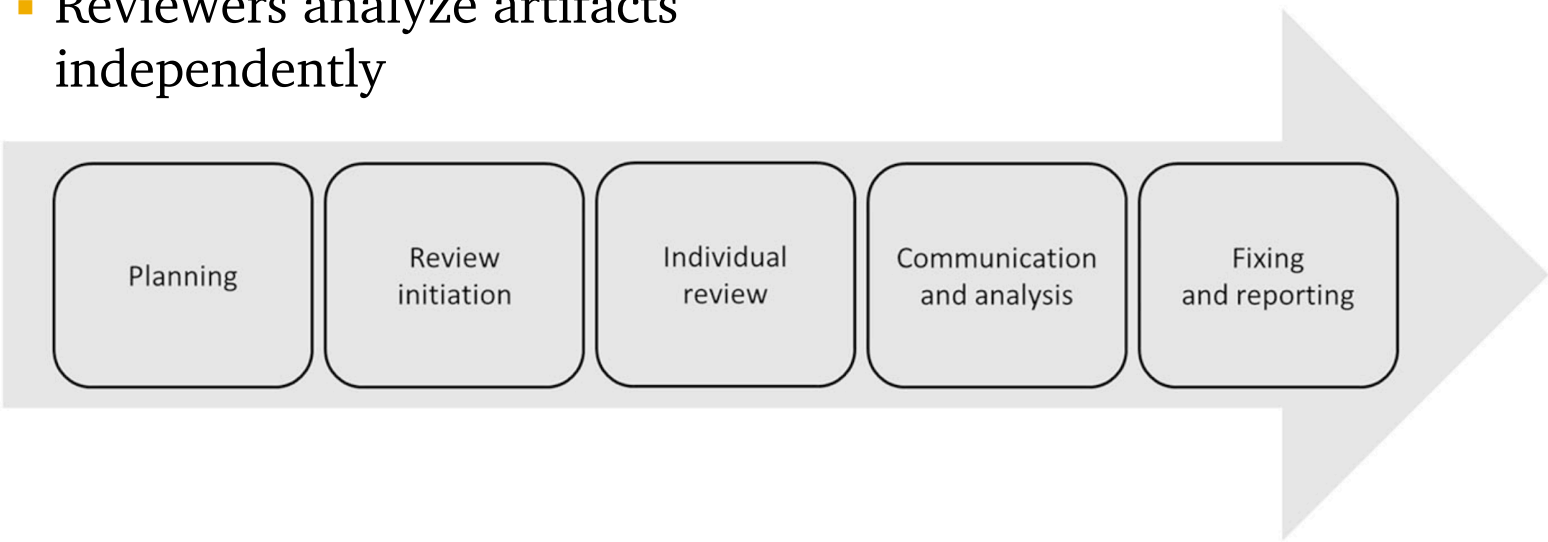
Feedback and Review Process

- **Importance of Early and Frequent Feedback:**
  - Prevents misunderstandings
  - Reduces costly rework
  - Aligns stakeholders with project goals
- **Benefits:**
  - Improves software quality
  - Enhances team collaboration
  - Reduces project risks



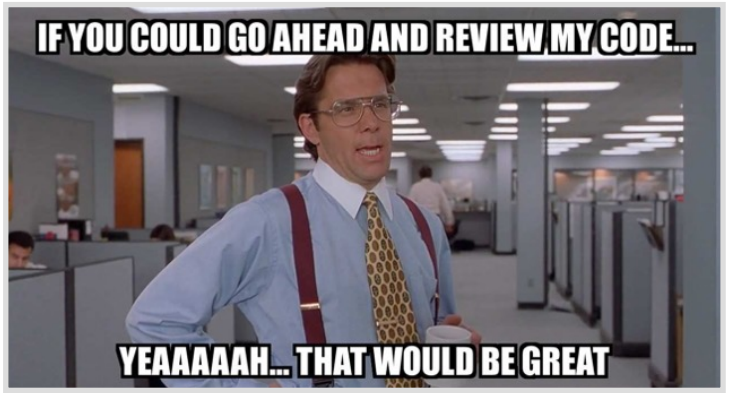
Review Process Activities

- **Planning**:
  - Define scope, objectives, participants, exit criteria
- **Review Initiation**:
  - Ensure reviewers have necessary materials
- **Individual Review**:
  - Reviewers analyze artifacts independently
- **Communication and Analysis**:
  - Discuss anomalies, assign actions
- **Fixing and Reporting**:
  - Document defects, report results



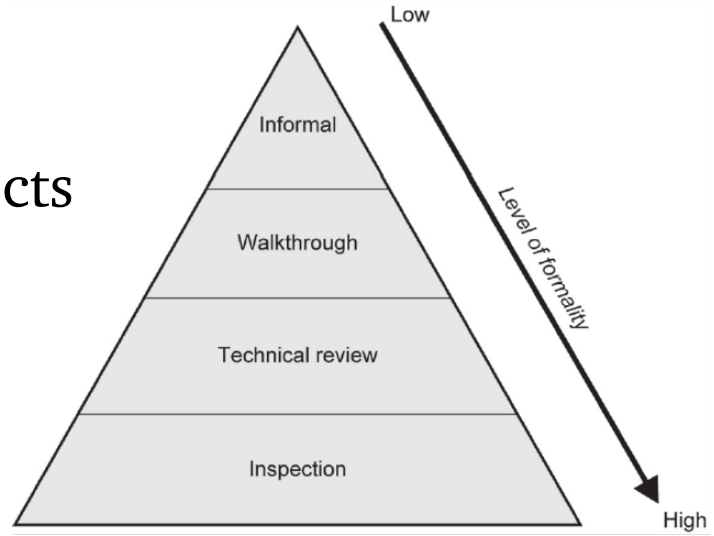
Roles in Reviews

- **Manager**: Allocates resources, defines scope
- **Author**: Creates work product, implements fixes
- **Moderator**: Facilitates review meetings
- **Scribe**: Records review findings
- **Reviewer**: Assesses work product, identifies issues
- **Review Leader**: Organizes and oversees the review



Review Types

- **Informal Review:**
  - No formal process or documentation
  - Focus on anomaly detection
- **Walkthrough:**
  - Led by the author
  - Ensures understanding, detects defects
- **Technical Review:**
  - Performed by technical experts
  - Aims to make decisions regarding a technical problem
- **Inspection:**
  - Most formal type
  - Follows structured process with metrics



Success Factors for Reviews

- Clear objectives and criteria
- Selecting the right review type
- Conducting reviews on small work products
- Providing adequate preparation time
- Management support
- Embedding reviews in company culture
- Providing training for reviewers