



Software Quality Assurance and Testing

Static Testing

Outline

- Static techniques and the test process
- Review process
- Static analysis by tools

Static Techniques

- Know the difference between static and dynamic

Static testing



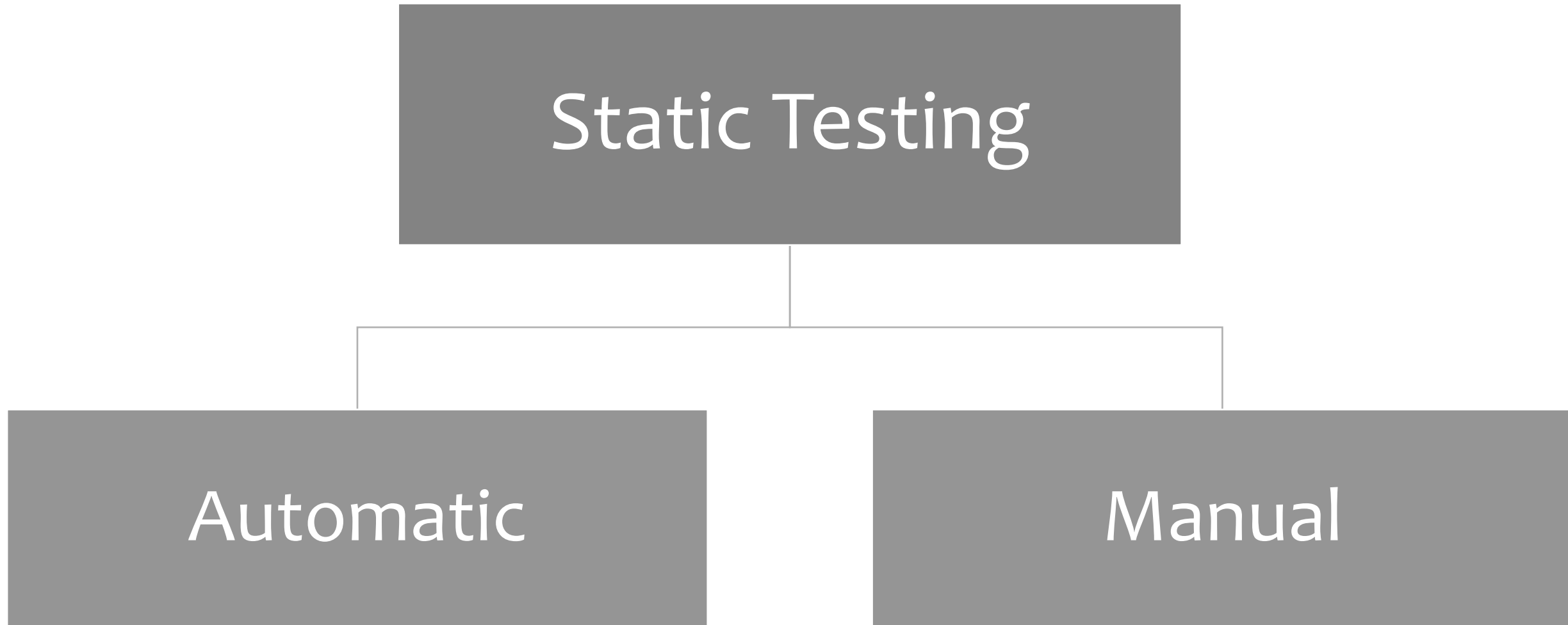
- Examination of code without executing it
- Can be applied to other work products

Dynamic testing

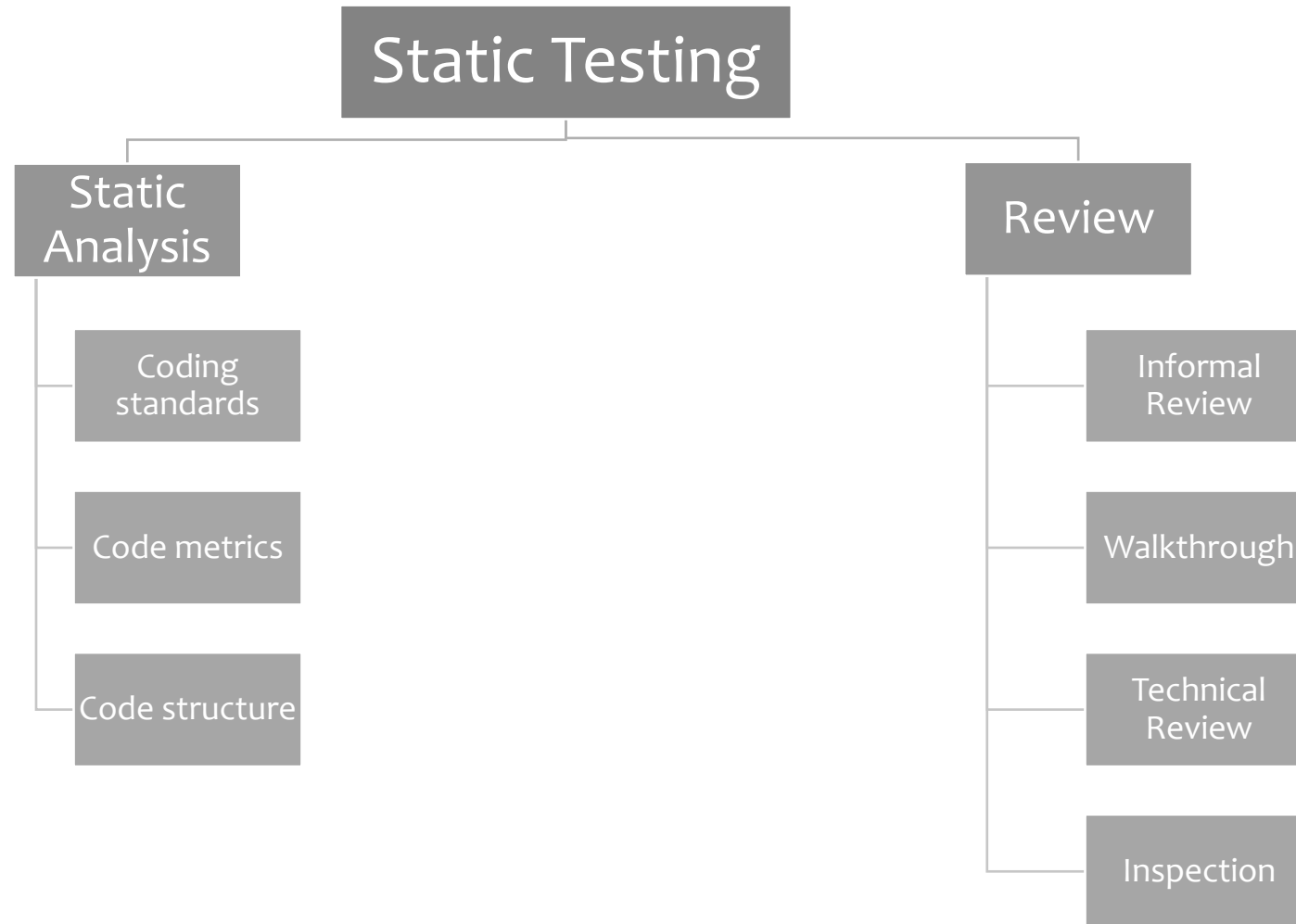


- Requires source code to be executed

Static Testing



Static Testing



Static Techniques

- Static and dynamic testing have the same objective: identifying defects.
- They are complementary.
- Compared to dynamic testing,
 - static techniques find causes of failures (defects) rather than the failures themselves.

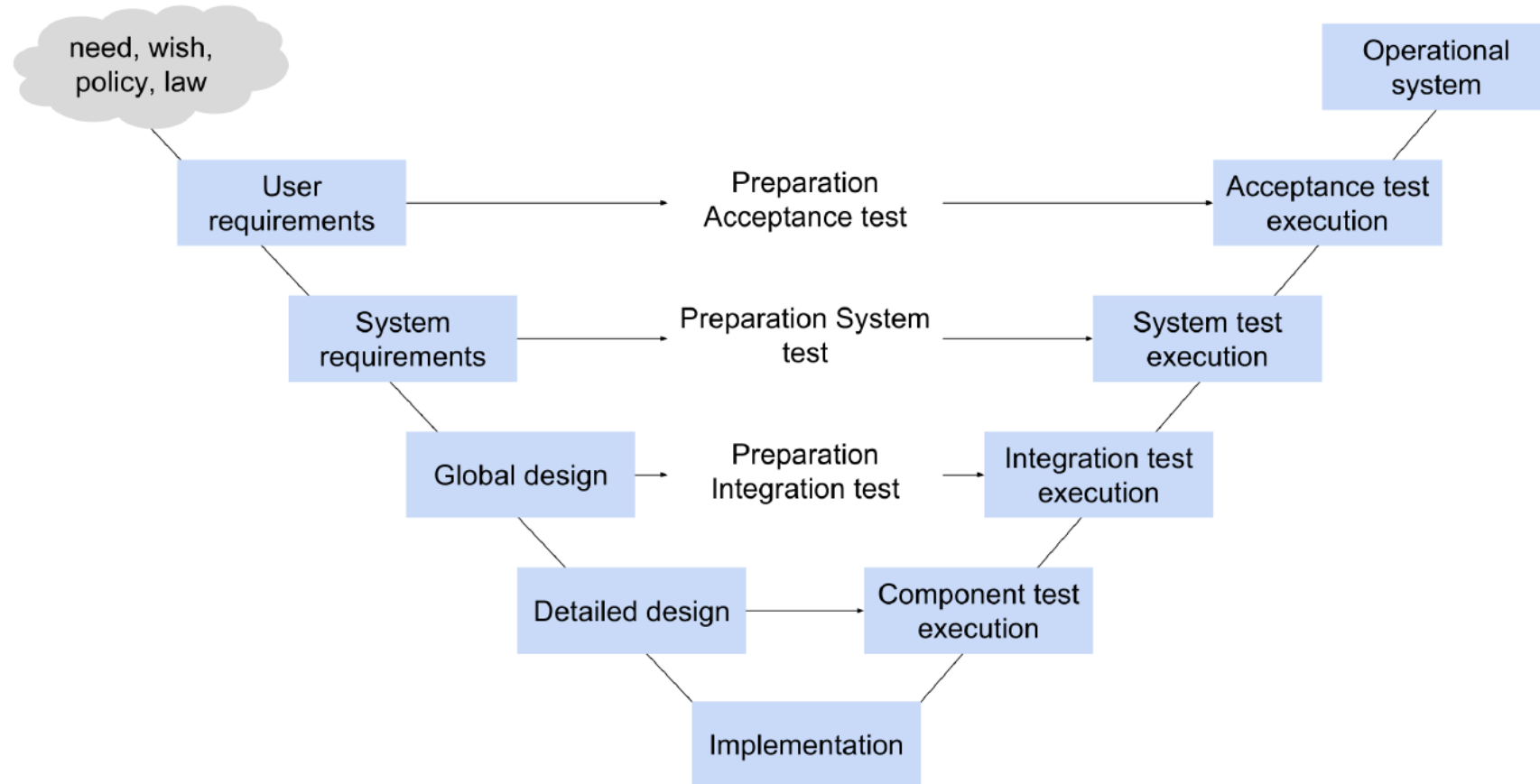
Static Techniques and the Test Process

- **Static testing** -manual and automated examination of software artefacts without execution of the software under test.
 - **Reviews** -a way of testing software products (including code) and can be performed well before dynamic test execution.
 - **Static analysis** tools analyze program code

Software Artefacts (Static Testing)

- Any type of specifications: business requirements, functional requirements, security requirements
- Epics, user stories and acceptance criteria
- Design diagrams and models
- Code
- Test strategy, test plan, test conditions, test cases
- User guides, help text, wizards
- Web pages
- Contracts

The V-model



Static Analysis by Tools

- **Objectives of static analysis**
- Find defects in
 - software source code
 - software models
- **Note!** *Static analysis finds defects rather than failures*
- Static analysis is performed without actually executing the software being examined by the tool.
- Static analysis tools analyze program code, as well as generated output such as HTML and XML.

Static Analysis by Tools

- **Typical defects** discovered by static analysis tools include:
 - referencing a variable with an undefined value
 - inconsistent interface between modules and components
 - variables that are never used
 - unreachable(dead) code
 - programming standards violations
 - security vulnerabilities
 - syntax violations of code and software models

Static Analysis by Tools

- **Developers**
- Use static analysis before and during:
 - Component testing
 - Integration testing
- **Designers**
 - Use static analysis during software modeling

Static Analysis by Tools

- **Why is static analysis valuable**

- Early detection of defects prior to test execution.
- Early warning about suspicious aspects of the code or design, by the calculation of metrics, such as a high complexity measure.
- Identification of defects not easily found by dynamic testing.
- Detecting dependencies and inconsistencies in software models, such as links.
- Improved maintainability of code and design.
- Prevention of defects, if lessons are learned in development.

Static Analysis by Tools

- **Coding standards**
- Recommended that existing standards should be adopted in order to save a lot of effort
 - Set of programming rules , i .e. always check boundaries on an array when using it
 - Naming conversions , e.g. class name should start with a Capital letter
 - Access conversions , e. public/private
 - Layout specifications, e.g. indents
 - Checking tools supports code standards



Static Analysis by Tools

- **Code metrics**

- Comments frequency
- Depth of nesting
- Cyclomatic complexity /complexity metrics

Dependency Finder

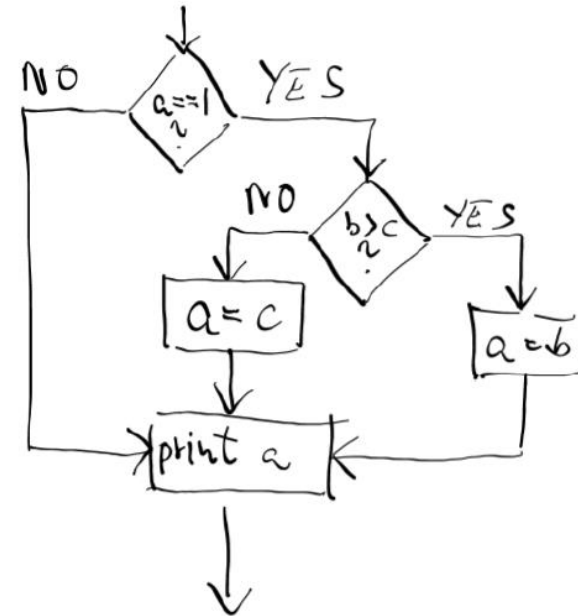
JDepend

sonarqube 

Static Analysis by Tools

- Complexity can be measured in different ways, e.g. based on the number of decisions in the program (the number of binary decisions)

```
if( a == 1 )  
{   if( b > c )  
    a = b;  
  else  
    a = c;  
}  
System.out.println( a );
```



Static Analysis by Tools

Code structure

- Control flow structure
 - The sequence in which the instructions are executed
- Data flow structure
 - follows the trail of a data item as it is accessed and modified by the code
- Data structure
 - The organization of the data itself, independent of the program (Array, list, stack, queue, tree, graph, ...)

Dependometer

Macker

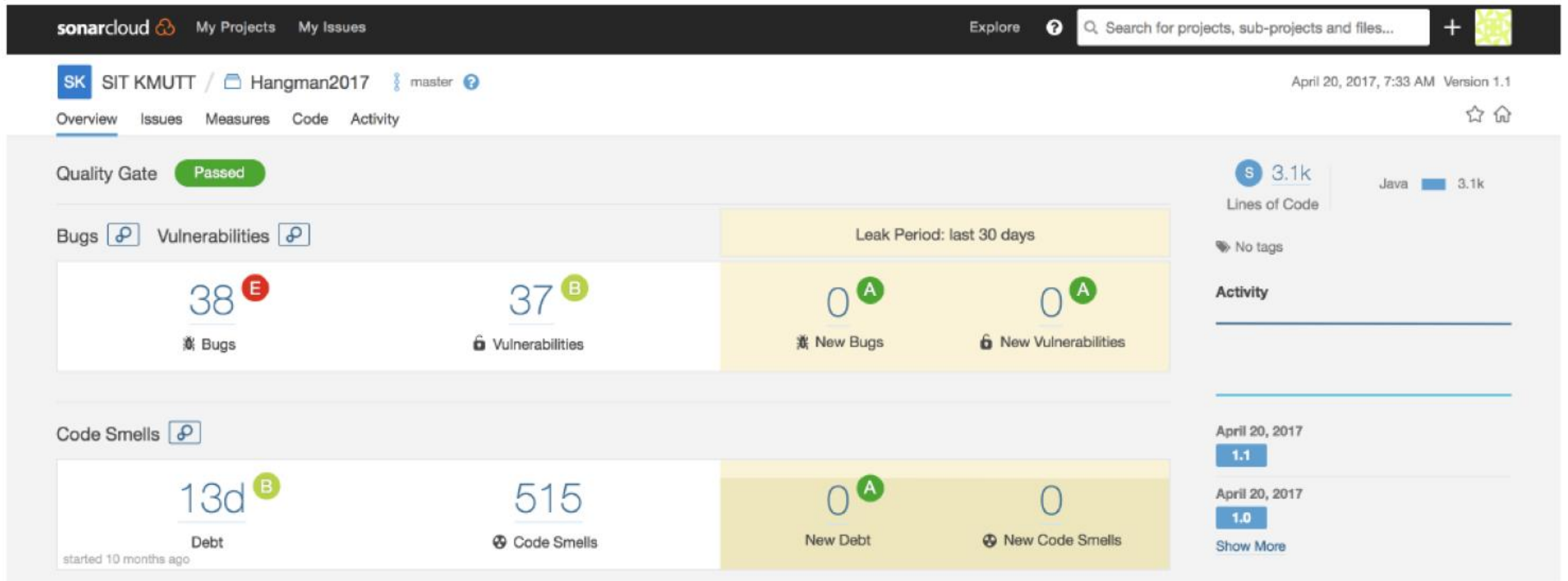


Static Analysis by Tools

- **Practical side**

- Static analysis tools may produce a large number of warning messages , which need to be well managed to allow the most effective use of the tool.

Examples of Testing tools



Reviews

- **Reviews**

- is a way of testing software products (including code) and can be performed well before dynamic test execution

- **Reason to make reviews**

- Defects detected during reviews early in the life cycle are cheaper to remove than those detected while running tests.
- Reviews can find omissions, for example, in requirements, which are unlikely to be found in dynamic testing.

- **Tools (manual + tool support)**

- The main manual activity is to examine a work product and make comments about it.

The review process

Object of reviews

- Any software work product can be reviewed, e.g.
 - requirements specifications
 - design specifications
 - code
 - test plans, test specifications, test cases, test scripts
 - user guides
 - web pages

The review process

- **Benefits**

- early defect detection and correction
- development productivity improvements
- reduced development timescales
- reduced testing cost and time
- lifetime cost reductions
- fewer defects
- improved communication

The review process

- **Typical defects**

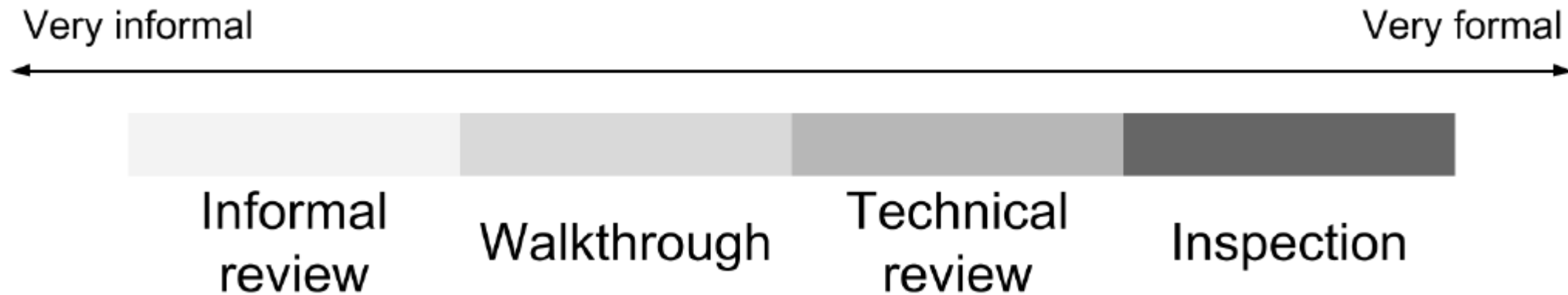
- deviations from standards
 - requirement defects
 - design defects
 - insufficient maintainability
 - incorrect interface specifications
 - inconsistencies, ambiguities, contradictions, omissions, inaccuracies, and redundancies in requirements
- These defects are easier to find in reviews than in dynamic testing

The Review Process

- Activities of a formal review
- Roles and responsibilities
- Types of reviews
- Review techniques
- Success factors for reviews

The review process

- Different types of reviews vary from:



very informal(e.g. no written instructions for reviewers) to very formal (i.e. well structured and regulated)

Review Process - Background

- The formality of a review process is related to factors like
 - Risk
 - Size of the project
 - the maturity of the development process
 - any legal or regulatory requirements
 - the need for an audit trail

Review Process - Background

- The way a review is carried out depends on the agreed objective of the review:
 - find defects and omissions
 - gain understanding
 - discussion and decision by consensus

Phases of a formal review

1. Planning
2. Initiate review – kick off
3. Individual review – individual preparation
4. Issue communication and analysis – review meeting
5. Fixing and reporting – Rework and follow-up

1. Planning

- a. Select the personnel
- b. Allocate roles
- c. Define the entry and exit criteria for more formal review types (e.g. inspection)
- d. Select which parts of documents to look at

2. Initiate review - kick off

- a. Distributing documents
- b. Explaining the objectives of the review and the review process
- c. Explaining the documents to the participants
- d. Checking and discuss entry/exit criteria

3. Individual review/preparation

- Work done by each of the participants on their own before the review meeting , noting potential defects , questions and comments
- Each participants proposes the severity of the defects
- Severity classes: critical, major or minor

4. Issue communication and analysis

Review meeting

- Logging and discussion, with documented results or minutes
- The meeting participants may simply note defects, make recommendations for handling the defects, or make decisions about the defects.
- Decisions based on the exit criteria
- Examining, evaluation and recording

5. Fixing and reporting

Rework and follow-up

- fixing defects found, typically done by the author.
- Check that defects have been addressed
- gather metrics, e.g.
 - number of defects found
 - number of defects found per page
 - time spent checking per page
 - total review effort
 - etc.

Roles and responsibilities

- *One person may take one or more roles!*
 - The author
 - Management
 - The review leader
 - The facilitator or moderator
 - The reviewers
 - The scribe (or recorder)

Author

- The writer or person with chief responsibility for the documents to be reviewed and the rework to be done.

The management

- Decides on the execution of reviews
- Assigning resources : staff, budget and time
- Determines if the review objectives have been met

The review leader

- Taking the overall responsibilities for the review
- Deciding who will be involved
- Works closely with both the management and the facilitator (moderator).

The facilitator or moderator

- leads the review of the document(s)
- planning the review
- running the meetings
- and follow up after the meeting
- If necessary, the facilitator or moderator may mediate between the various points of view and is often the person upon whom the success of the review rests

The reviewers

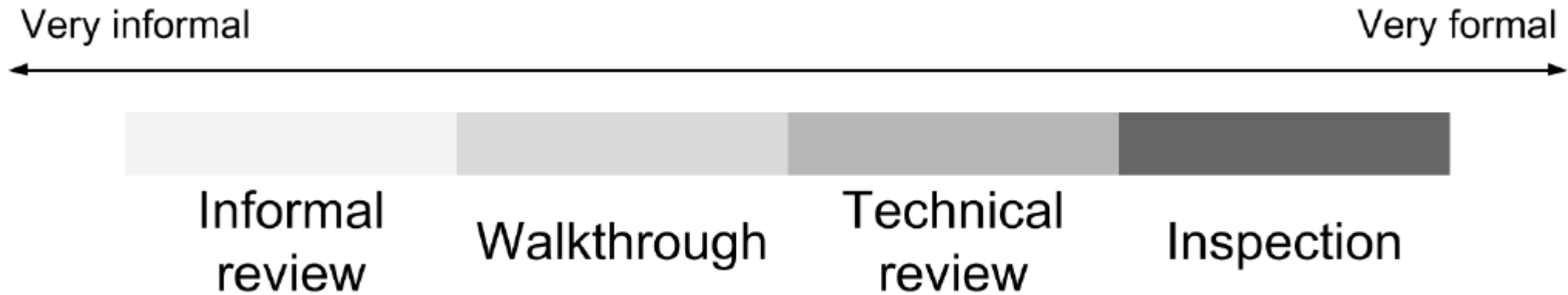
- Individuals with specific technical or business background
- Identify and describe the findings in the product under review
- Note: reviewers should be chosen to represent different perspectives and roles in the review process
- Note: reviewers should take part in the review meeting

Scribe (or recorder)

- Documents all the issues , problems and open points that were identified during the meeting.

Types of reviews

1. Informal review
2. Walkthrough
3. Technical review
4. Inspections



1. Informal review

- **Purpose**

- Inexpensive way to get some benefit

- **Form**

- Pair reviews; e.g. pair programming or a technical lead reviewing designs and code

- Note: No formal process

- Note: Optionally may be documented

2. Walkthrough

- **Purposes**

- learning
- gaining understanding
- defect finding
- feedback

- **Form**

- meeting led by author
- may vary in practice from quite informal to very formal
- stakeholders may participate

3. Technical review

- **Purposes**

- discuss
- make decisions
- evaluate alternatives
- find defects
- solve technical problems
- check conformance to specifications and standards

- **Form**

- May vary from very formal to informal peer review without management participation.

3. Technical review

- ideally led by trained facilitator or moderator
- documented, defined defect-detection process; includes peers and technical experts
- pre-meeting preparation
- optionally the use of checklists, review report, list of findings and management

4. Inspection

- **Purpose**

- Find defects

- **Form**

- Usually peer examination led by trained facilitator or moderator (not the author)
- Formal process based on rules and checklists with entry and exit criteria
- pre-meeting preparation
- defined roles
- includes metrics
- inspection report, list of findings

Review techniques

- Ad hoc reviewing
- Checklist-based reviewing
- Scenario-based reviewing and dry runs
- Role-based reviewing
- Perspective-based reviewing

Success factors for reviews

- Organizational success factors
 - Have a clear objective
 - Pick the right review type and technique
 - Review material need to be kept up to date
 - Limit the scope of review
 - Enough time!
 - Management support is critical

Success factors for reviews

- People related success factors
 - Pick the right reviewers (testers mindsets)
 - Each reviewers does their review work well
 - Limit the scope of the review and pick things that really count
 - Defects found should be welcomed
 - Review meeting are well managed
 - Trust is critical
 - Communication is important
 - Follow the rules, but keep it simple
 - Train participants
 - Continuously improve process and tools

Success factors - approach

- Defects found are welcome and expressed objectively
- Apply suitable review techniques for the type and level of software products.
- Use checklists or roles if appropriate to increase effectiveness of defect identification.
- Management supports a good review process (e.g. by incorporating adequate time for review activities).

Success factors - training and learning

- Training is given in review techniques, especially the more formal techniques, such as inspection.
- There is an emphasis on learning and process improvement