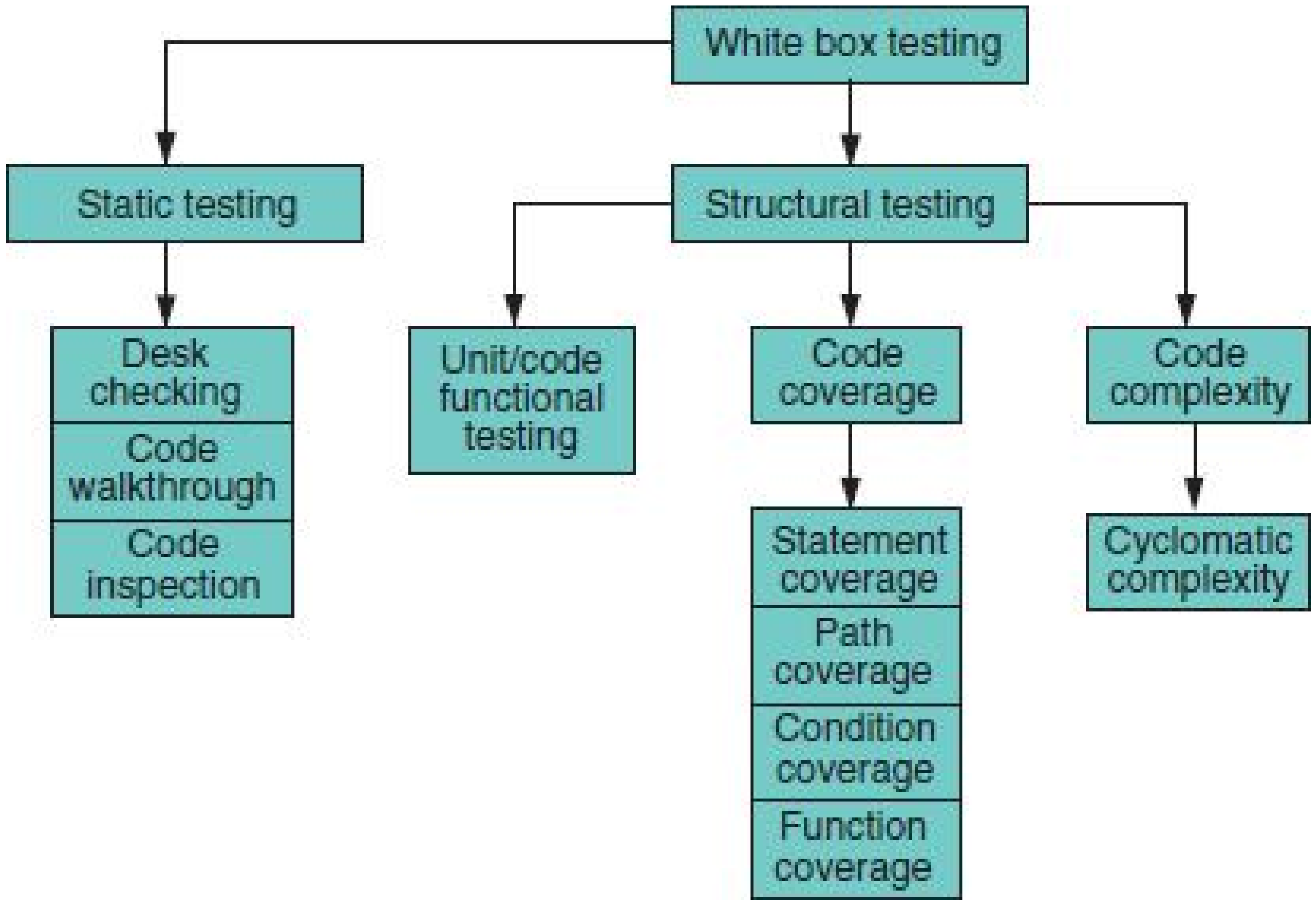
White Box Testing

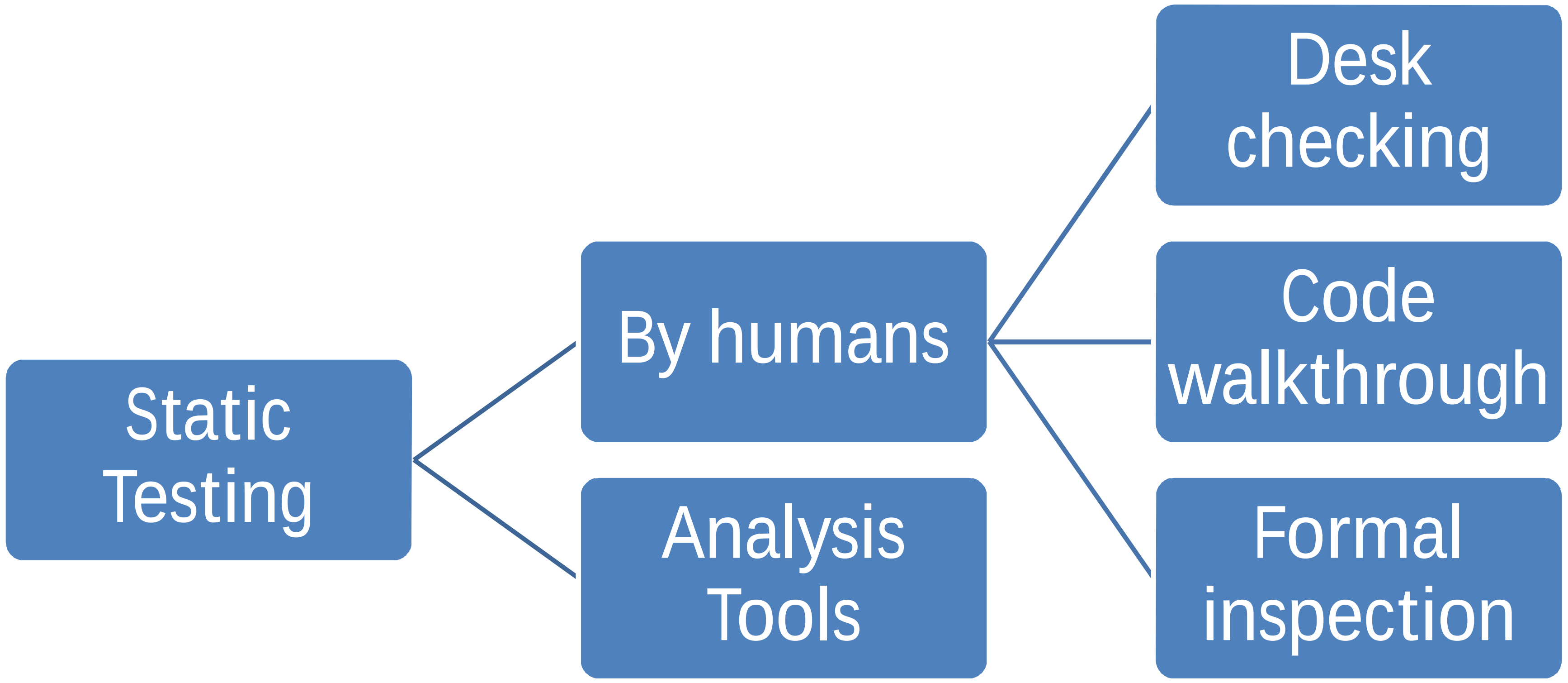
Classification of White Box Testing



Static Testing

- only the source code of the product, not the binaries or executables.
- does not involve executing the programs
  - The code works according to the functional requirements
  - The code has been written in accordance with the design developed earlier in the project life cycle
  - The code for any functionality has been missed out
  - The code handles errors properly

Types of static testing



Static testing by Humans

- Humans read the program to detect errors.
- Advantages
  - Sometimes human can find errors that computers cannot
  - By making humans read and evaluate the program, we can get multiple perspectives and therefore have more number of problems identified
  - A human evaluation of code can compare it against the specification or design and thus ensure that it does what is intended to do.
  - A human evaluation can detect many problems at one go and can try to identify the root cause of the problems
  - By making humans test the code before execution, computer resources can be saved
  - A proactive methods like static testing minimizes the delay in identification of the problems

Desk checking of the code

- Method to verify the portion of the code for correctness
- Verification is done by comparing the code with the design or specification
- Completely relies on author thoroughness, diligence and skills
- No structured or formalism to ensure completeness
- Mo maintaining of log

- Advantages
  - The programmer knows the code and programming language well and hence is best suited to read the program
  - Less scheduling and logistics overheads
  - Reduces delay in defect detection and correction
- Disadvantages
  - Person dependent, not scalable or reproducible
  - Tunnel visioning of developers and have blind spots
  - Developers prefer writing new code and don’t like testing

Code walkthrough

- Group oriented (as against desk checking)
- Brings in multiple perspectives
- Multiple specific roles (author, moderator, inspector, etc.), asks questions to the author.
- If the author unable to answer, he takes and find the answers
- Completeness is limited.

Formal Inspection / Fagan Inspection

- Group-oriented activity
- Highly formal and structured
- Has specific roles
- Requires thorough preparation

Roles in Formal Inspection

- Author
  - Author of the work product
  - Makes available the required material to the reviewers
  - Fixes defects that are reported
- Moderator
  - Controls the meeting(s) – defect logging meeting
- Inspectors (reviewers)
  - Prepare by reading the required documents
  - Take part in the meeting(s) and report defects
- Scribe
  - Takes down notes during the meeting
  - Assigned in advance by turns
  - Can participate to review to the extent possible
  - Documents the minutes and circulates them to participants

Process

- Typical documents circulated:
  - Program code
  - Design / program specifications
  - SRS (if needed)
  - Any applicable standards (e.g., coding standards)
  - Any necessary checklists (e.g., code review checklist)

- Advantages
  - Thorough, when prepared well
  - Brings in multiple perspectives
  - Has been found to be very effective
- Disadvantages
  - Logistically difficult
  - Time consuming
  - May not be possible to exhaustively go through the entire code

Static Analysis Tools

- Reduce manual work
- Errors found may be:
  - Whether they are reachable codes
  - Variables declared but not used
  - Mismatch in definition and assignment of values to variables
  - Illegal or error prone typecasting of variables
  - Use of non portable or architecture dependent programming constructs
  - Memory allocated but not having corresponding statements for freeing them up memory
  - Calculation of cyclomatic complexity

Structural Testing

- Takes into account the code, code structure, internal design and how they are coded.
- It’s the actual run by the computer on the built product.
- Types:
  - Unit functional testing
  - Code coverage testing
  - Code complexity testing.

Unit /code functional test

- Initially by knowing the i/p and o/p the tester does a quick test – checks mistakes
- Modules with logic or condition, the developer can build a “debug version” by placing some Intermediate statements – assure those intermediate statements are removed later
- To run the product under debugger or IDE – allows the developer to stop at the end of each instruction, view and modify the contents

Thank You