

Static Testing

Relies on the manual examination or tool-driven evaluation of work products

Almost any work product can be tested:

- specifications, including business requirements, functional requirements, and security requirements
 - epics, user stories, and acceptance criteria
 - architecture and design specifications
 - code
- testware, including test plans, test cases, test procedures, and automated test scripts
 - user guides
 - web pages
- contracts, project plans, schedules, and budget planning
- configuration set up and infrastructure set up
- models, such as activity diagrams, which may be used for model-based testing

Finds defects in work products directly, with much less effort

Can be used to improve the consistency and **internal** quality of work products

Typical defects that are easier and cheaper to find and fix:

- requirement defects
- design defects
- coding defects
- deviation from standards
- incorrect interface specifications
- security vulnerabilities
- gaps or inaccuracies in test basis traceability or coverage
- **most types of maintainability defects**
(possible to find only by static testing)

Enables the early detection of defects before dynamic testing is performed

Finding and fixing the defects promptly is almost always much cheaper than using dynamic testing to find defects

Dynamic Testing

Requires the execution of the software being tested

What is being tested:

- functional behavior of software system
- memory/CPU usage
- overall performance of the system

Identifies failures caused by defects when the software is run

Typically focuses on **externally** visible behaviors

Defects to find and fix (those **not prevented** by static testing):

- functional defects
- non-functional defects

Reveals the uncovered defects that are complicated and cannot be covered by static testing

Cost of finding and fixing defects is high