Software Testing

Overview

- Definition of Software Testing
- Problems with Testing
- Benefits of Testing
- Effective Methods for Testing

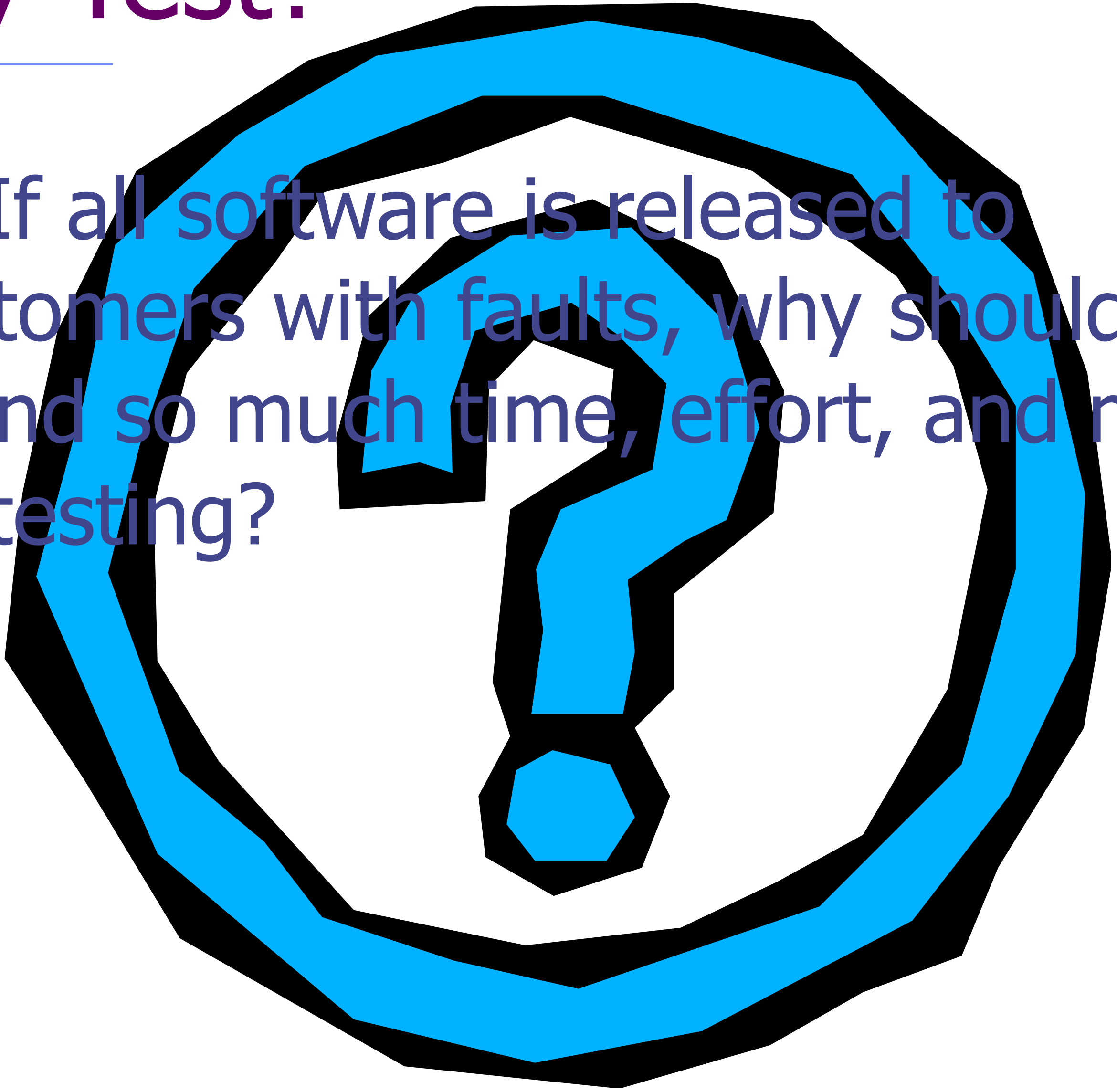
Definition of Software Testing

Software testing is the process of executing a software system to determine whether it matches its specification and executes in its intended environment.

“Program testing can be a very effective way to show the presence of bugs, but it is hopelessly inadequate for showing their absence” [Dijkstra, 1972]

Why Test?

Q: If all software is released to customers with faults, why should we spend so much time, effort, and money on testing?



Cost of Delaying the Release of a Software Product

- Timing is another important factor to consider.
- New products: The first to the market often sells better than superior products that are released later.

Beta Testing

- Customers test for free!
- Seems to give you test cases representative of customer use.
- Helps to determine what is most important to the customers.
- Can do more configuration (environment) testing than in your testing lab.

Problems with Beta Testing

- Most beta testers are “techies” who have a higher tolerance of bugs. They do not represent the average customer.
- Beta testers usually won’t report: usability problems, bugs they don’t understand and bugs that seem obvious.
- Takes much more time and effort to handle a user reported bug.

Cutting Testing Costs can Increase other Costs

- Customer support can be very expensive. Less bugs = less calls.
- Customers will look for more reliable solutions.
- Software organizations must perform cost benefit analysis’ to determine how much to spend on testing.

Problems with Testing

Since it is impossible to find every fault in a software system, bugs will be found by customers after the product is released.

Another Problem

In many software companies, testers are ill-equipped to test software. For example: My last co-op firm (which will remain unnamed).

- Testing done almost entirely by untrained co-ops.
- Testers were responsible for creating black-box test plans without being given formal specifications.
- Testers were not provided with tools to automate test plans.

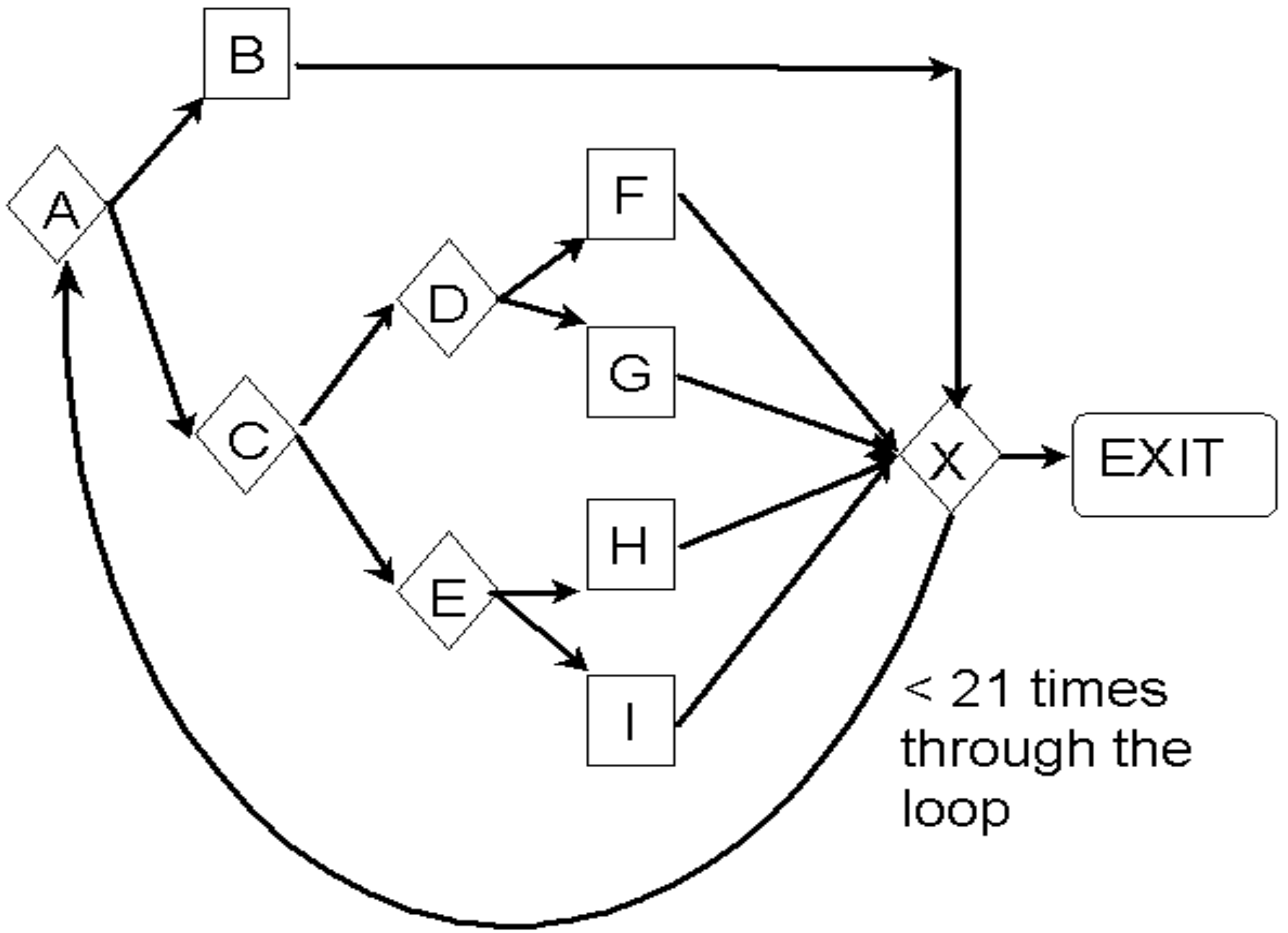
Reasons that Bugs Escape Testing

- User executed untested code.
- User executed statements in a different order than was tested.
- User entered an untested combination of inputs.
- User’s operating environment was not tested.

Why Can’t Every Bug be Found?

- Too many possible paths.
- Too many possible inputs.
- Too many possible user environments.

Too Many Possible Paths

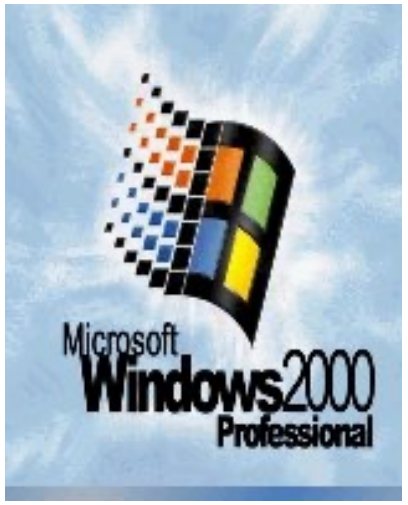


Too Many Possible Inputs

- Programs take input in a variety of ways: mouse, keyboard, and other devices.
- Must test Valid and Invalid inputs.
- Most importantly, there are an infinite amount of sequences of inputs to be tested.

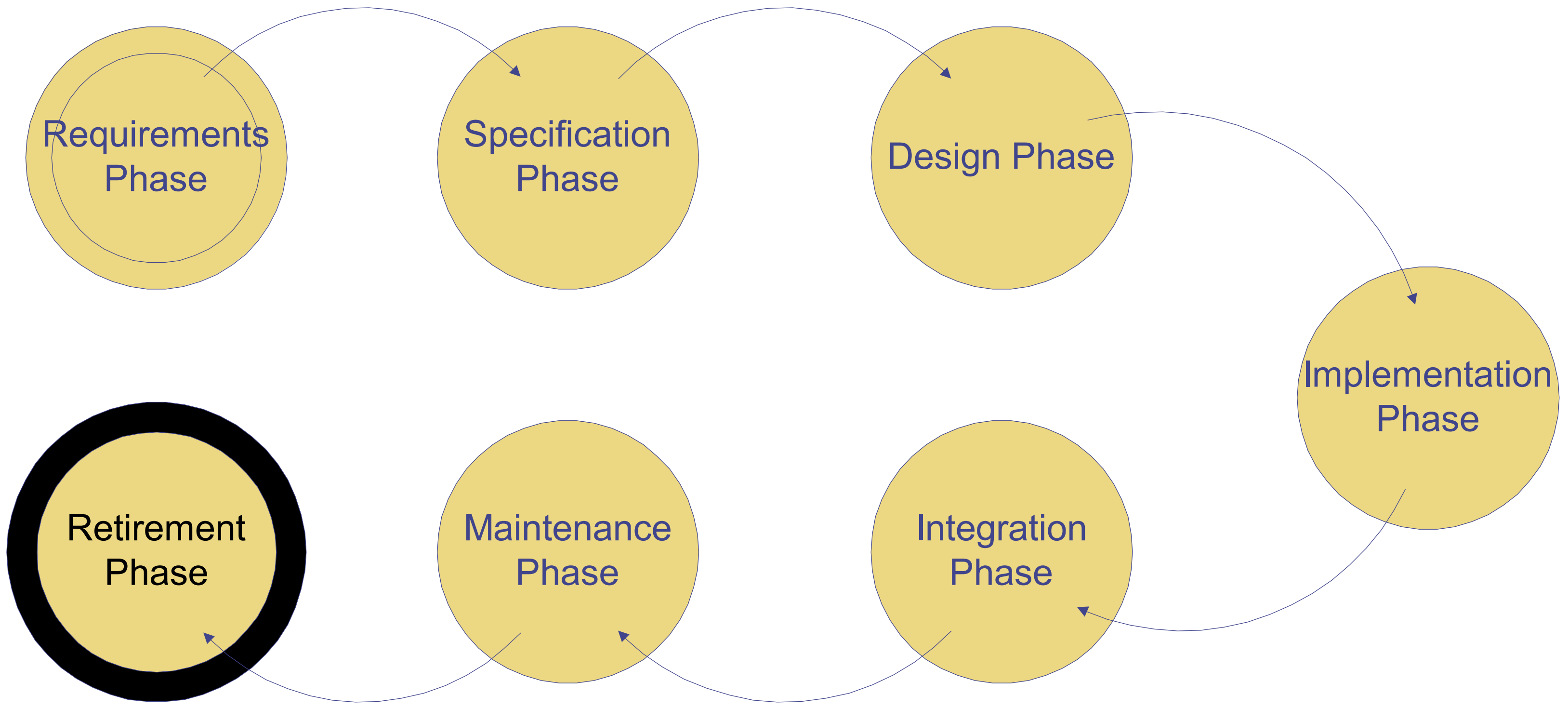
Too Many Possible User Environments

- Difficult to replicate the user’s combination of hardware, peripherals, OS, and applications.
- Impossible to replicate a thousand-node network to test networking software.



How is Testing Done

Phases of the Software Process



Why No Testing Phase?

- Testing must be done at every phase.
- Testing of a phase must be build upon and checked against the results of the previous phase.
- Non-execution based testing is done in early phases (before executable code is produced).
- Execution and non-execution based testing can be done in later phases.

Nonexecution-Based Testing

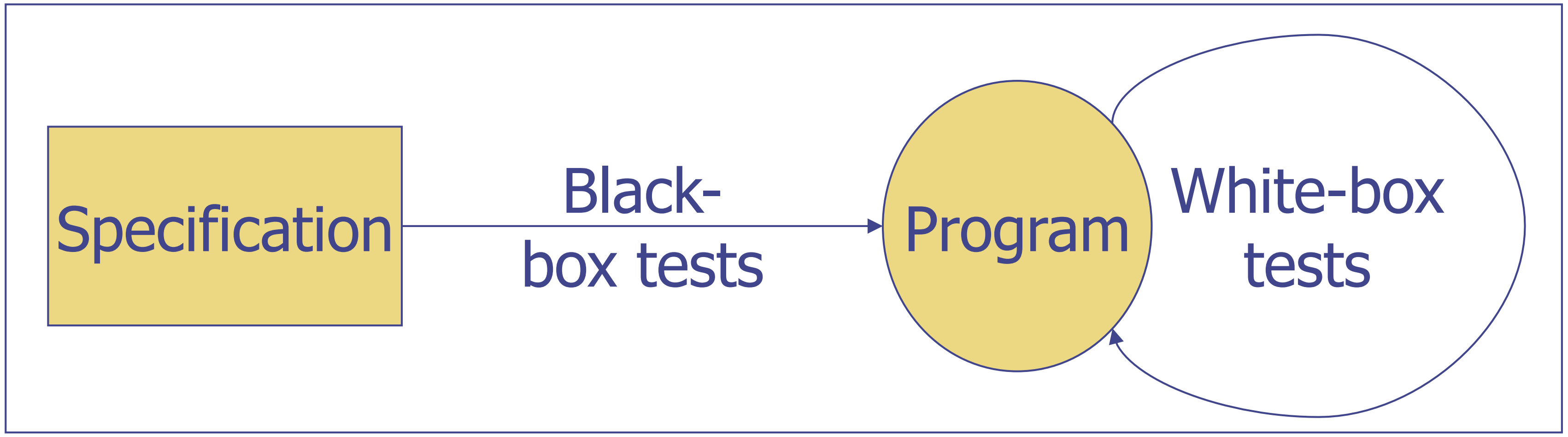
- Walkthroughs
- Inspections
- Walkthroughs are shorter and more informal than inspections.
- Goal of both is to record faults, not to correct them.

Execution-Based Testing

- Utility
- Reliability
- Robustness
- Performance
- Correctness

Black-Box / White-Box Testing

- Black-box tests are driven by the program's specification
- White-box tests are driven by the program's implementation



Test Automation

- If a manual test costs $X to run the first time, it will cost $X to run every time thereafter.
- An automated test can cost 3 to 30 times $X the first time, but will cost about $0 after that.

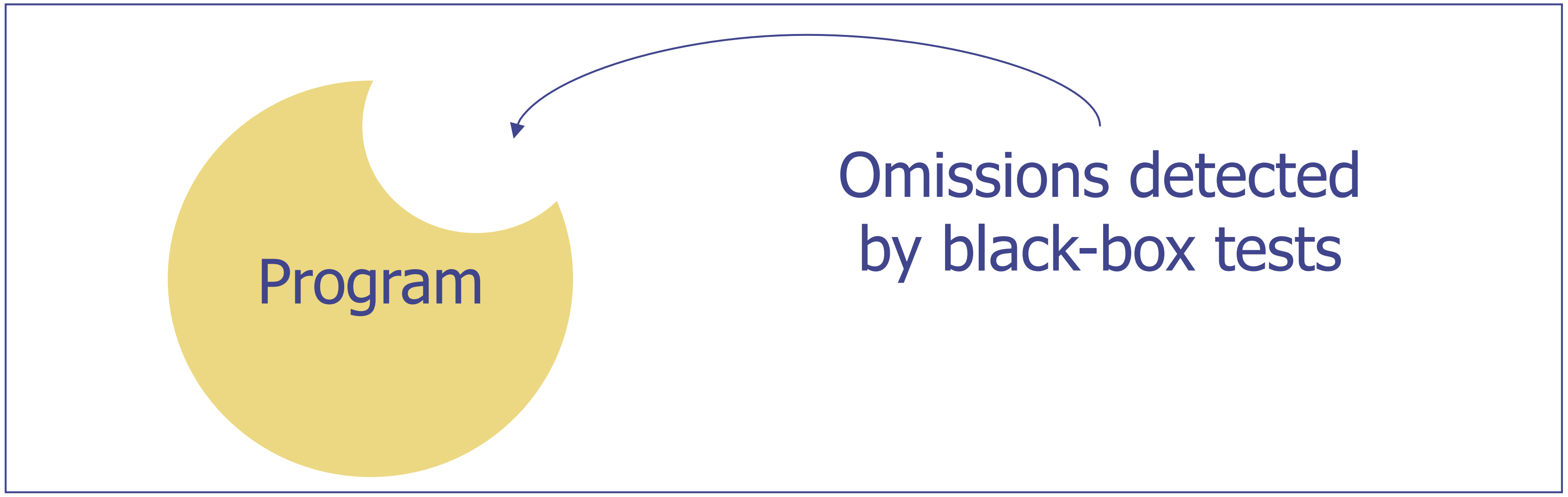
Any Questions?

Too Many Possible Paths

- There are 5 paths from A to X without passing through the loop.
- There are 5^{20} paths from A to X after passing through the loop 20 times.
- There are $5 + 5^2 + 5^3 + \ldots + 5^{20} = 100$ trillion possible paths in this program.
- If you could test a path per second it would take more than 3 million years!

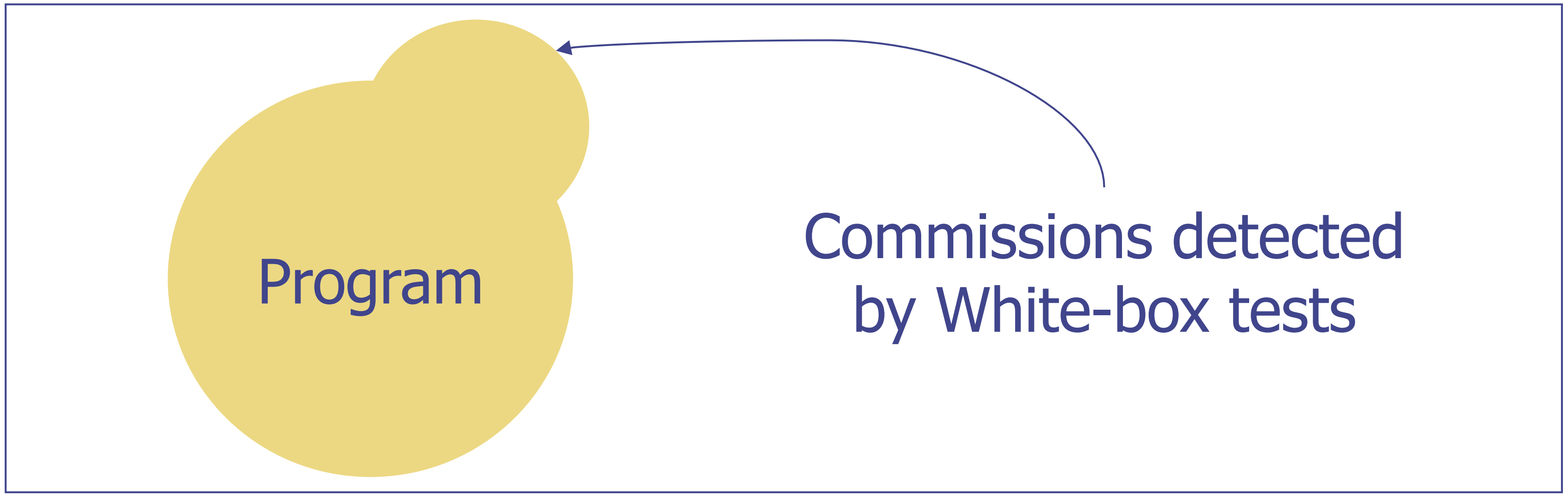
Black Box Testing

- Checks that the product conforms to specifications
- Cannot determine how much code has been tested



White Box Testing

- Allows tester to be sure every statement has been tested.
- Difficult to discover missing functionality.


Program
Commissions detected by White-box tests