Chapter 1
Introduction to Effective Software Testing

Objectives

- How the software testing has been evolved?
- What are the goals of Software testing?
- Model for testing process.
- Complete testing is not possible.
- Various Schools of Software testing.

Evolution of Software Testing

Software Testing 1.0

- In this phase, software testing was considered to be just a single phase to be performed after coding of the software in SDLC. No test organization was there. A few testing tools were present but their use was limited due to the high cost. Management was not concerned with the testing as there was no quality goal.

Software Testing 2.0

- In this phase, software testing gained the importance in SDLC and the concept of early testing was also started. Testing was evolving into the direction of planning the test resources. Many testing tools were also available in this phase.

Evolution of Software Testing

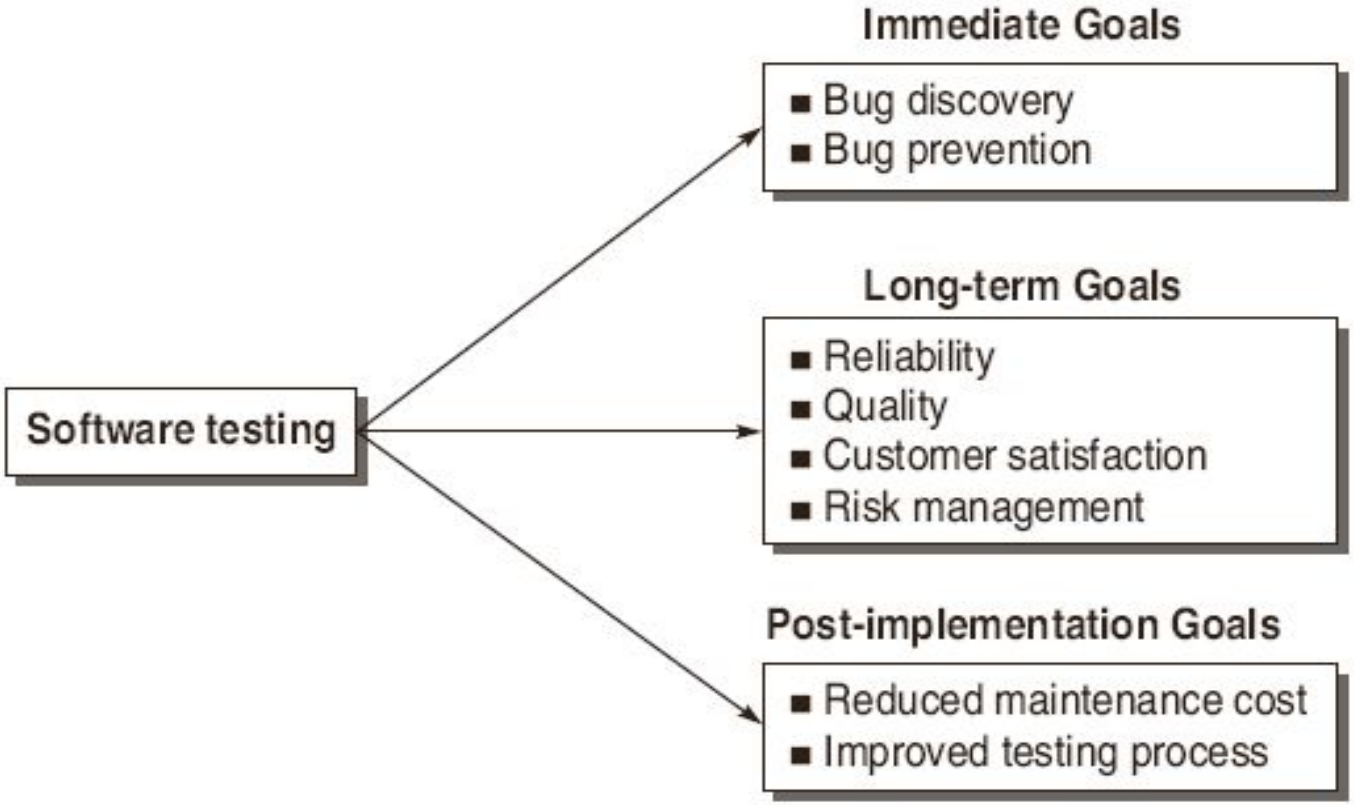
Software Testing 3.0

- In this phase, software testing is being evolved in the form of a process which is based on the strategic effort. It means that there should be a process which gives us a road map of the overall testing process. Moreover, it should be driven by the quality goals in mind so that all controlling and monitoring activities can be performed by the managers. Thus, management is actively involved in this phase.

Software Testing Myths

- *Testing is a single phase in SDLC performed after coding*.
- *Testing is easy*.
- *Software development is of more worth as compared to testing*.
- *Complete testing is possible*.
- *The testing starts after the program development*.
- *The purpose of testing is to check the functionality of the software*.
- *Anyone can be a tester*.

Software Testing Goals



Software testing

Immediate Goals

- Bug discovery
- Bug prevention

Long-term Goals

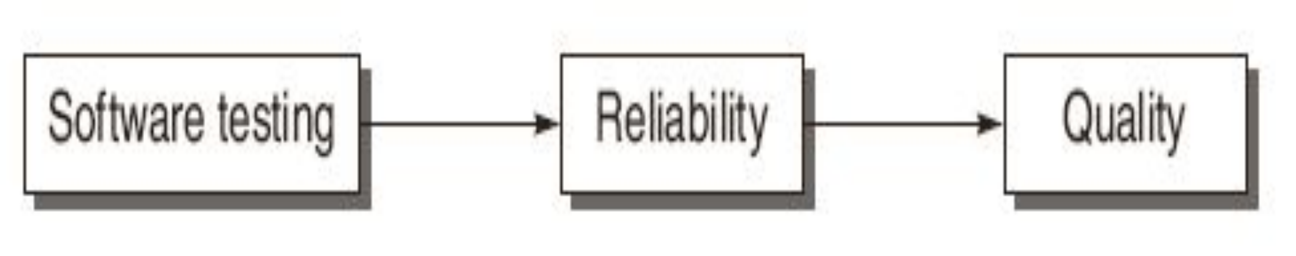
- Reliability
- Quality
- Customer satisfaction
- Risk management

Post-implementation Goals

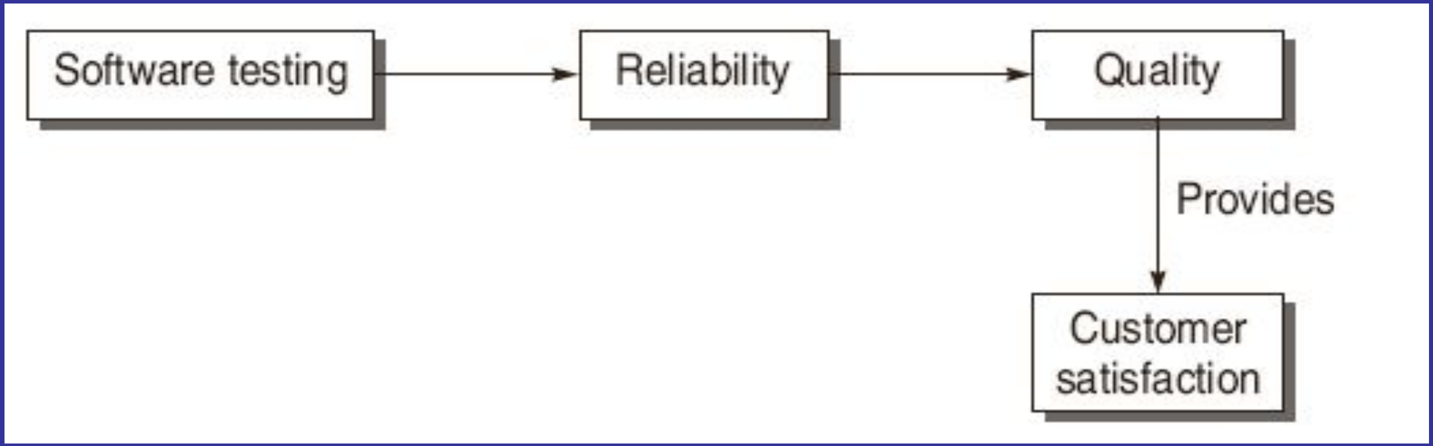
- Reduced maintenance cost
- Improved testing process

Figure 1.2 Software testing goals

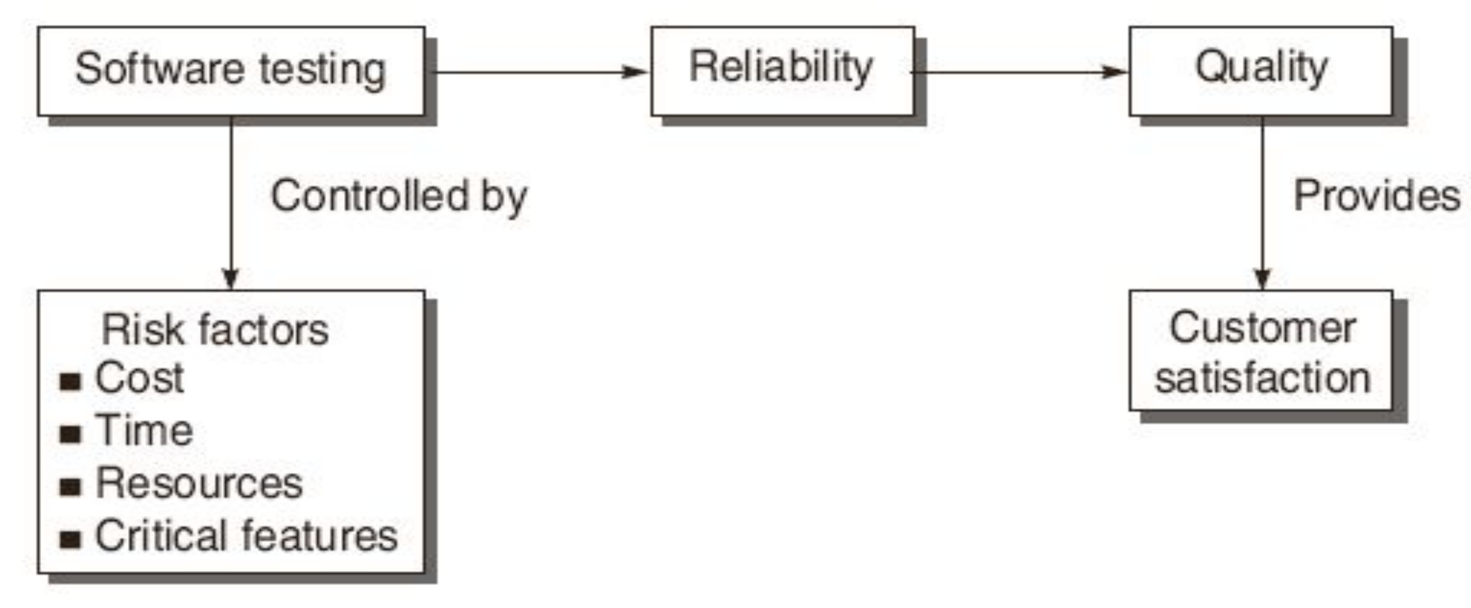
Testing produces Reliability and Quality



Quality leads to customer satisfaction



Testing controlled by Risk factors



Software Testing Definitions

- *“Testing is the process of executing a program with the intent of finding errors.”*

  - **Myers [2]**

- *“A successful test is one that uncovers an as-yet-undiscovered error.”*

  - Myers [2]

- *“Testing can show the presence of bugs but never their absence.”*

  - **W. Dijkstra [125]**.

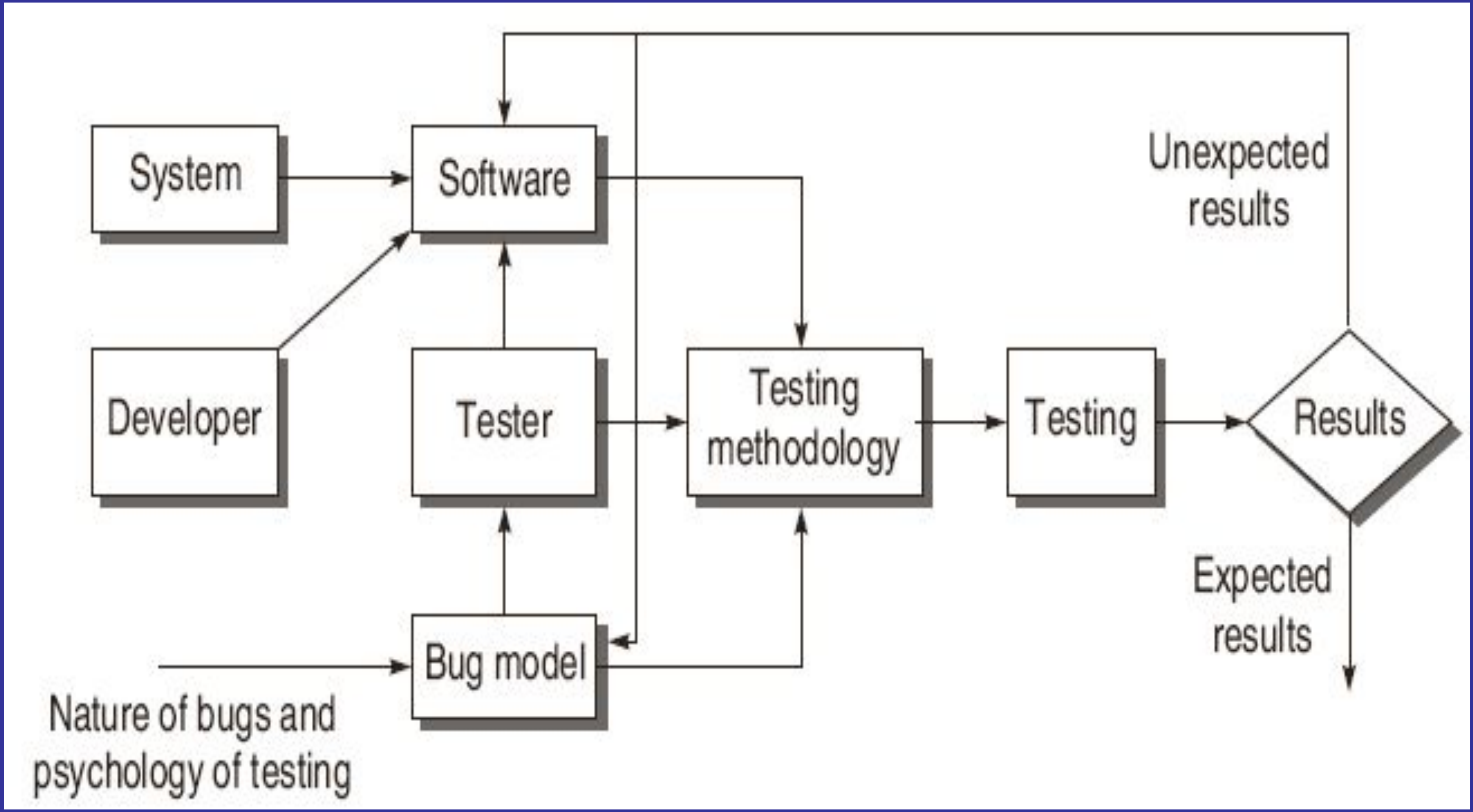
Software Testing Definitions

- *“Testing is a concurrent lifecycle process of engineering, using and maintaining testware (i.e. testing artifacts) in order to measure and improve the quality of the software being Tested.”*

  - *Craig* [117]

- *“Software testing is a process that detects important bugs with the objective of having better quality software.”*

Model for Software Testing



Software and software Model

- Software complexity
- Easy to design
- Easy to code
- Testable at every point
- Avoid unnecessary complexities

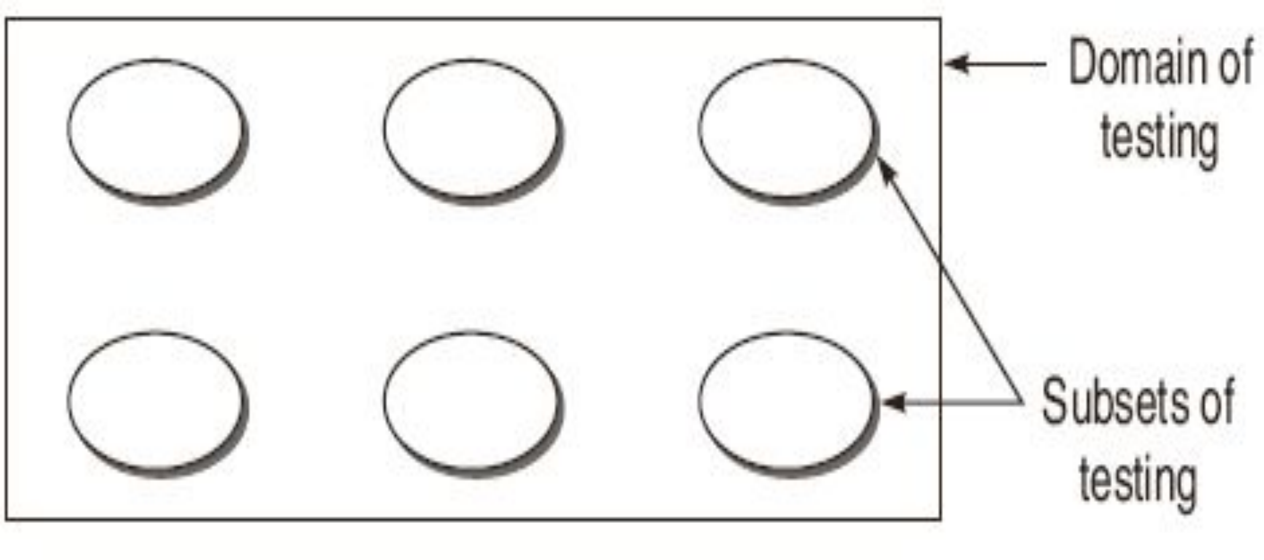
Bug Model

- Perception of the kind of bugs expected
- Helps in deciding testing strategy
- If we get incorrect result then bug model need to be modified.

Testing methodology and testing

- Incorporates testing strategy and testing tactics
- Roadmap for overall testing process
- If we don’t get required result then testing methodology need to be modified

Effective Software Testing vs Exhaustive Software Testing

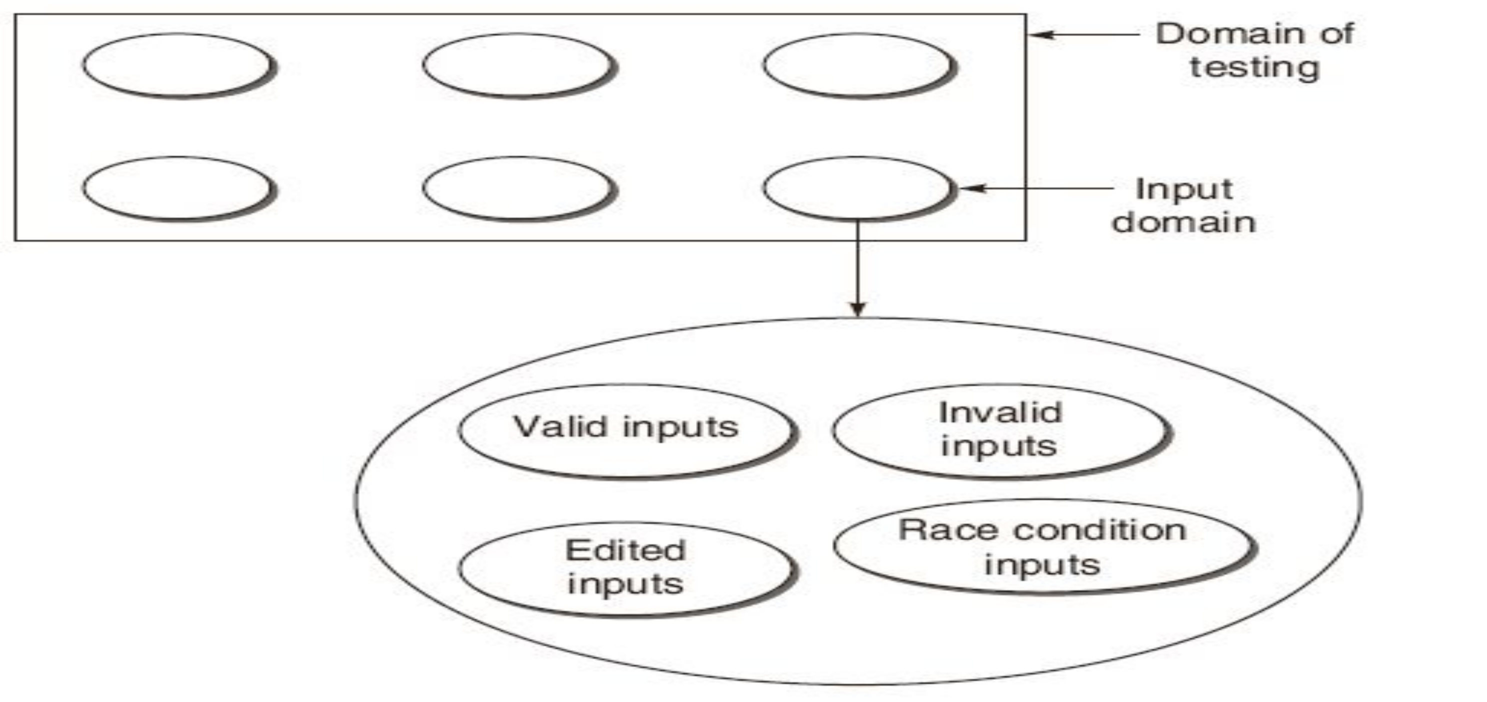


Effective Software Testing vs Exhaustive Software Testing

- The domain of possible inputs to the software is too large to test.

- Valid Inputs
- Invalid Inputs
- Edited Inputs
- Race Conditions

Effective Software Testing vs Exhaustive Software Testing



Valid inputs

- Adding of two digit no.s
- -99 to 99 i.e (199total)
- Total no. of test case combinations =199*199=39601

Invalid inputs

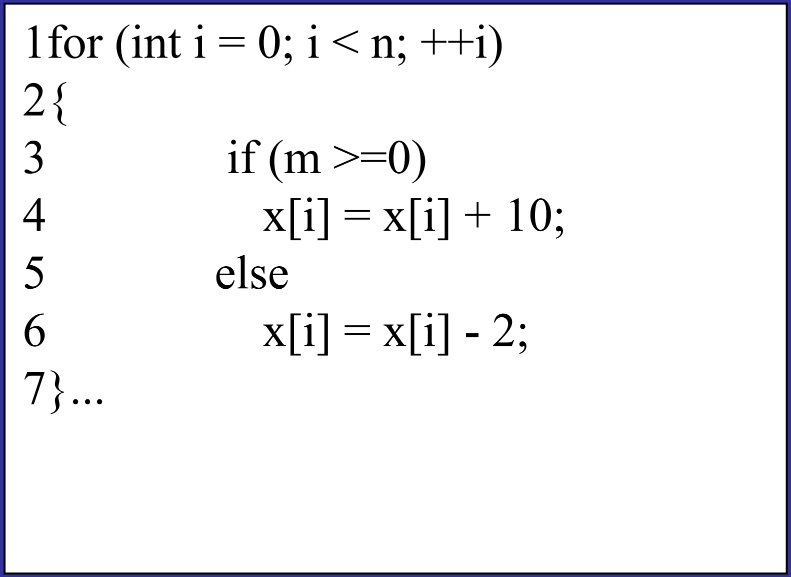
- Numbers out of range
- Combinations of alphabates and digits
- Combination of all alphabates
- Combination of control characters
- Combination of any other key on the keyboard

Edited inputs

- Adding of spaces when giving the input to the programs
- Non functioning of a program
- Buffer overflow and system crashes

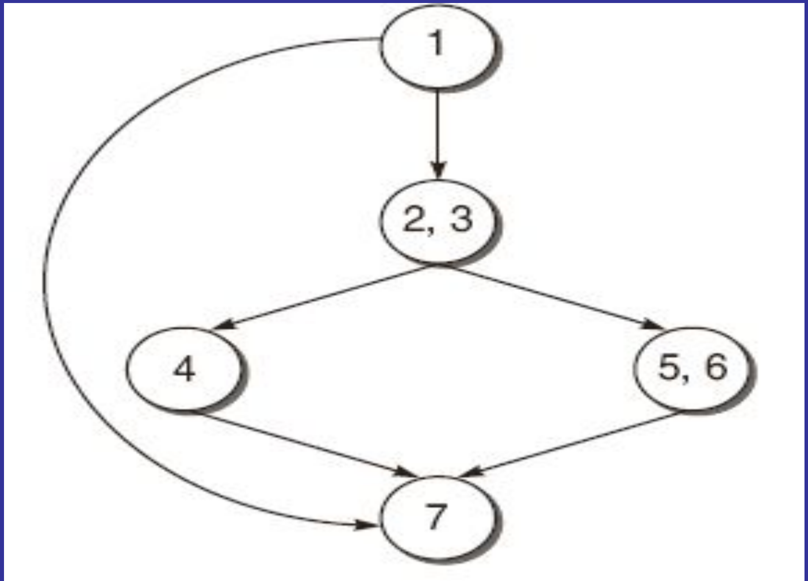
Effective Software Testing vs Exhaustive Software Testing

There are too many possible paths through the program to test.



```
for (int i = 0; i < n; ++i)
{
            if (m >=0)
              x[i] = x[i] + 10;
           else
              x[i] = x[i] - 2;
}...
```

Effective Software Testing vs Exhaustive Software Testing

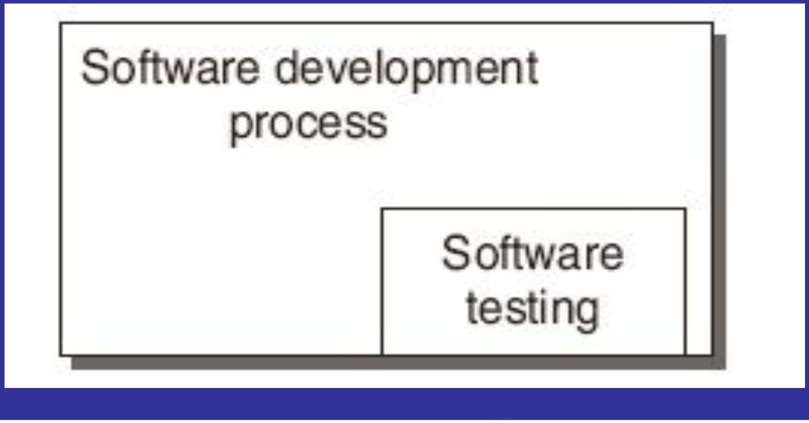


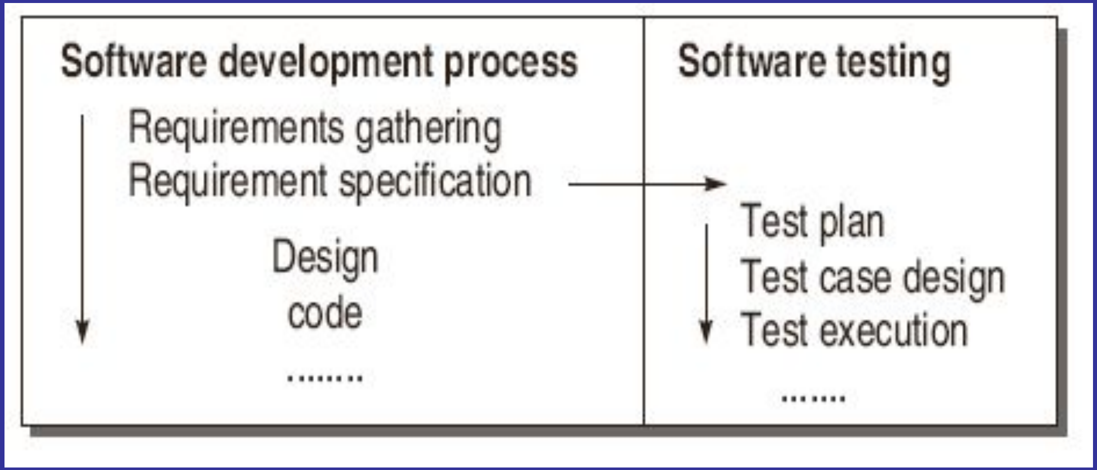
- if n is 20, then the number of paths will be $2^{20} + 1$, i.e. 1048577.

Effective Software Testing vs Exhaustive Software Testing

- Every Design error cannot be found.

Software Testing as a Process





Software Testing as a Process

- An organization for the better quality software must adopt a testing process and consider the following points:
- Testing process should be organized such that there is enough time for important and critical features of the software.
- Testing techniques should be adopted such that these techniques detect maximum bugs.
- Quality factors should be quantified so that there is clear understanding in running the testing process. In other words, process should be driven by the quantified quality goals. In this way, process can be monitored and measured.
- Testing procedures and steps must be defined and documented.
- There must be scope for continuous process improvement.