

# Software Engineering & Information System Design



Course Code: CSE-3319

Lecture 6

# Software Quality Definition

---

Conformance to

- ❧ Explicitly stated functional and performance requirements
- ❧ Explicitly documented development standards, and
- ❧ Implicit characteristics that are expected of all professionally developed software

(Pressman)

# Software Quality Factors



- ❧ Low defects level when deployed
  - ❧ Zero defect most preferably
- ❧ High reliability
  - ❧ Capability of running without crashes
- ❧ Majority of the user satisfy with software when conducted the survey
- ❧ Effective customer support
- ❧ Rapid defect repair

# Software Qualities



- ❧ Correctness
- ❧ Reliability
- ❧ Robustness
- ❧ Performance
- ❧ User friendliness
- ❧ Verifiability
- ❧ Maintainability
- ❧ Reparability
- ❧ Safety

- ❖ Evolvability
- ❖ Reusability
- ❖ Portability
- ❖ Understandability
- ❖ Productivity
- ❖ Size
- ❖ Timeliness
- ❖ Visibility



# Benefits Of Software Quality



- ❧ Reduced maintenance cost
- ❧ Stable and useful product
- ❧ Satisfy customer needs
- ❧ Better chances for continuing releases
- ❧ Build corporate culture and identity
- ❧ Better chances for software and design reuse

# Software Quality Assurance



- ❧ Quality assurance – a set of establish framework (procedures & standard) used by an organization to achieve high quality software
- ❧ Essential activity for any business that produce product to used by others
  - ❧ It need to be planned and systematic
  - ❧ It does not just happen
- ❧ Assure each of the software qualities is met
  - ❧ Goal set in requirements specification
  - ❧ Goal realized in the implementation
- ❧ SQA is applied at every stage of the software process
- ❧ Quality standard is selected to apply to the software process

# Principle Of SQA



1. Set the standard and quality attributes that a software product must meet
  - ❧ The goal to achieve
2. Measure the quality of software product
  - ❧ There is a way to determine how well the product conforms to the standards and quality attributes
3. Track the values of the quality attributes
  - ❧ It is possible to assess how well we are doing
4. Use the information of software quality to improve the quality of future software product
  - ❧ There is a feedback into the software development process

# SQA Activities



- ❧ Application of Technical Methods
- ❧ Conduct of Formal Technical Reviews
- ❧ Software Testing
- ❧ Enforcement of Standards
- ❧ Control of Change
- ❧ Measurement
- ❧ Record keeping and Reporting



# Software Testing



- ❧ Testing is a process of executing a program with the intent of finding errors, locate errors and to proof system correctness
- ❧ Testing is conducted based on the developed test cases
  - ❧ Capture actual output
  - ❧ Compare actual output with expected output
    - ❧ Actual == Expected : Test case succeed
    - ❧ Actual != Expected : Test case failed
- ❧ All test cases result must be recorded

# Software Testing(Cont)



- ❧ To identify errors in software developed
- ❧ Test case design methods produces tests to be used on the software
- ❧ Development of test strategy for a set of tests that uncovers all possible errors
- ❧ Thorough testing is not as effective as expected in most cases

# Testing Objective



- ❧ To design tests that systematically uncover different classes of errors and to do so with a *minimum amount of time and effort*

# Testing Principles



- ❧ All test should be traceable to customer requirements.
- ❧ Tests should be planned before testing begins.
- ❧ Testing should begin with individual components and move towards to integrated cluster of components.
- ❧ The most effective testing should be conducted by an independent third party.



# Who Should Test The Software?

---

- ❧ Developer (individual units)
- ❧ Independent test group (ITG)
  - ❧ removes conflict of interest
  - ❧ reports to SQA team

# Test Cases



- ❧ Test case : unit of testing activity
- ❧ Test cases have 3 parts :-
  - ❧ Goal
    - ❧ Aspect of the system being tested
  - ❧ Input and system state
    - ❧ Data provided to the system under stated condition
  - ❧ Expected behavior
    - ❧ The output or action the system should take according to its requirements

# Type Of Test Cases



- ❧ Test cases are derived for
  - ❧ Valid and expected input
  - ❧ Invalid and unexpected input
  - ❧ Test if the system does less than specified requirement
  - ❧ Test if the system does more than specified requirement

# White Box Testing



- ❧ Derived from knowledge of program's structure & implementation
- ❧ *Structural testing* - analyse code & use knowledge of the structure of a component to derive test data
- ❧ Logical paths are tested by providing test cases that exercise specific sets of conditions and/or loops



# White Box Testing

(Continued)



- ❧ Thorough white box testing would lead to “100 percent correct programs”
- ❧ Exhaustive testing are impractical - too many tests!
- ❧ A limited number of logical paths can be selected and exercised
- ❧ Important data structures can be probed for validity

# White Box Test Cases



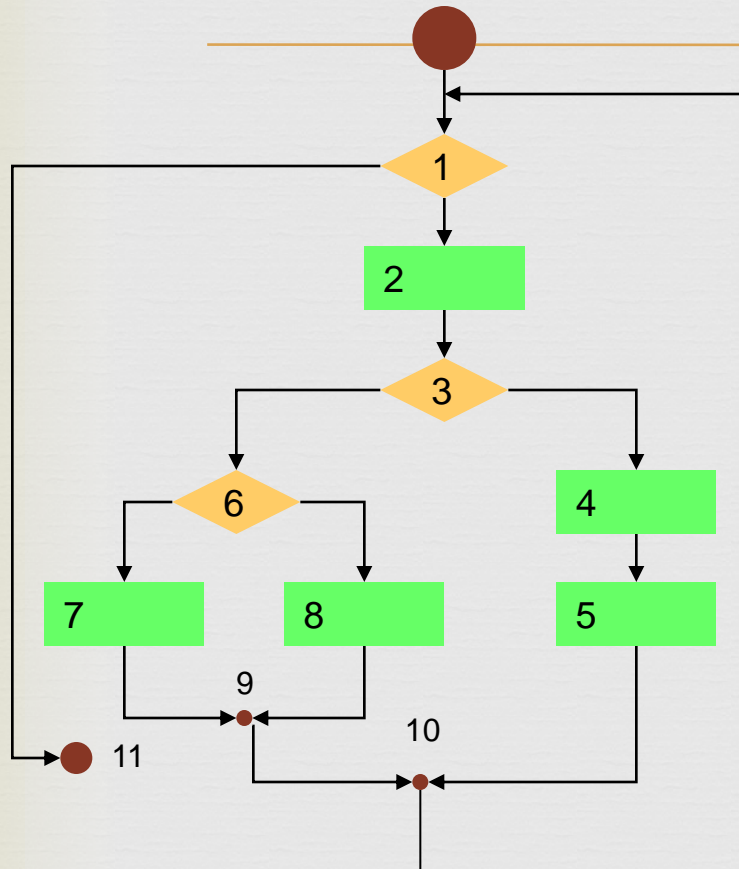
- ❧ Guarantee that all independent paths have been exercised at least once
- ❧ Exercise all logical decisions on their true and false sides
- ❧ Execute all loops at their boundaries and within their operational bounds
- ❧ Exercise internal data structures to ensure their validity

# White Box Testing Techniques

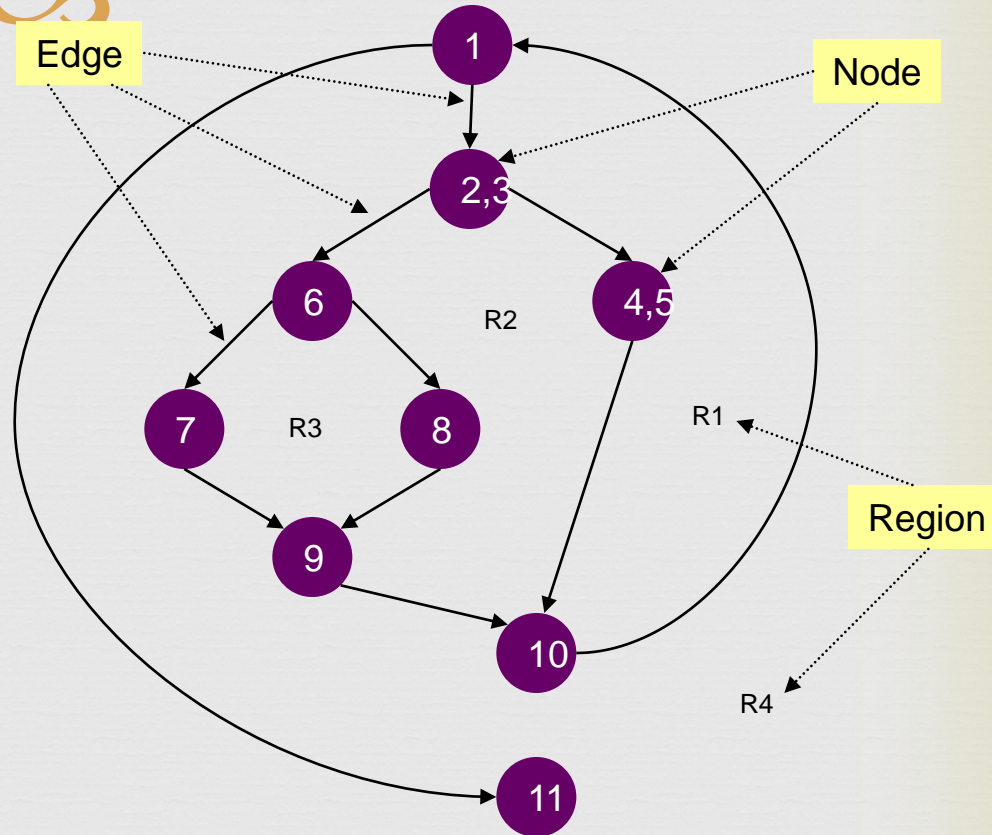
---

- ❧ Basis path testing
  - ❧ Flow graph notation
  - ❧ Cyclomatic complexity
  - ❧ Derived test cases
  - ❧ Graph metrics
- ❧ Control structure testing
  - ❧ Condition testing
  - ❧ Data Flow testing
  - ❧ Loop testing

# Flow Graph Notation



$\mathcal{B}$





# Cyclomatic Complexity



- ❧ Provide quantitative measure for program logical complexity.
- ❧ Defined number of independent path
  - ❧ Any path that introduce one set of processing statements or new condition
- ❧ Eg :-
  - ❧ Path 1 : 1-11
  - ❧ Path 2 : 1-2-3-4-5-10-1-11
  - ❧ Path 3 : 1-2-3-6-8-9-10-1-11
  - ❧ Path 4 : 1-2-3-6-7-9-10-1-11

# How Is Cyclomatic Complexity Computed?

1. Number of regions
  - The flow graph has 4 regions
2.  $V(G) = E - N + 2$ 
  - E : Number of flow graph edges
  - N : Number of flow graph nodes
    - ⌘  $V(G) = 11 \text{ edges} - 9 \text{ nodes} + 2 = 4$
3.  $V(G) = P + 1$ 
  - P : Number of predicate nodes
    - ⌘  $V(G) = 3 \text{ predicate nodes} + 1 = 4$

# Discussion on White Box Testing

---

## ⌘ Advantages

- ⌘ Find errors on code level
- ⌘ Typically based on a very systematic approach, covering the complete internal module structure

## ⌘ Constraints

- ⌘ Does not find missing or additional functionality
- ⌘ Does not really check the interface
- ⌘ Difficult for large and complex module