Coding and Testing

Courtesy:

Roger Pressman, Ian Sommerville &

Prof Rajib Mall

Coding:

- The objective of the coding phase is to transform the design of a system into code in a high-level language , and then to unit test this code.
- Software development organization adhere to well defined and standard style of coding- **Coding Standards.**
  - A coding standard gives a uniform appearance to the codes written by different engineers.
  - It facilitates code understanding and code reuse.
  - It promotes good programming practices.

Coding:

- **Coding standards** are mandatory for the programmers to follow.
- Compliance of their code to coding standards is verified during **code inspection.**
- Any code that does not conform to the coding standards is rejected during **code review** and the code is reworked by the concerned programmer.
- In contrast, **coding guidelines** provides some general suggestions regarding the coding style to be followed but leave the actual implementation of these guidelines to the discretion of the individual developers.

Coding Standards and Guidelines:

- Organization usually develop their own coding standards and guidelines depending on
  - what suits their organization best and
  - based on the specific types of software they develop.
- **Representative Coding Standards:**
  - Rules for limiting the use of global.
  - Standard headers for different modules
  - Naming conventions for global variables, local variables, and constant identifiers.
  - Convention regarding error return values and exception handling mechanisms

Example: Standard headers for different modules

- An example of header format that is being used in some organization:
  - Name of the module
  - Date on which the module was created
  - Author's name
  - Modification history
  - Synopsis of the module
  - Different functions supported in the module, along with their input/output parameters
  - Global variables accessed/modified by the module

Coding Guidelines:

- The representative coding guidelines followed by the Software development organization are as ..
  - Do not use coding style that is too difficult to understand.
  - Do not use identifier for multiple purposes
  - Code should be well documented
  - Length of any function should not exceed 10 source lines
  - Do not use GO TO statements

Code Review:

- Eliminating an error from code involves: **Testing (detecting failures), debugging (Locating errors), and then correcting the errors.**
- Code review and testing are both effective defect removal mechanisms.
  - The **code review** is much more cost effective strategy to eliminate errors from code as it directly detect errors.
  - **Testing** only helps detect failures and significant effort is needed to locate the error during debugging.
- The following two types of reviews are carried out on the code of a module:
  - Code inspection
  - Code walkthrough

Code Walkthrough:

- Code walkthrough is an **informal code analysis technique.**
- The main objective of code walkthrough is **to discover the algorithmic and logical errors** in the code.
- Discussion in the walkthrough meeting should focus on discovery of errors and avoid deliberations on how to fix the discovered errors.
- The size of the team performing code walkthrough should consist of three to seven members and **managers should not be a part of walkthrough meetings.**

Code Inspection:

- During code inspection, the code is examined to check for the presence of some common programming errors.
- The principal aim of code inspection is
  - to check for the presence of some common types of errors that usually creep into code due to programmer mistakes and oversights and
  - to check whether coding standards have been adhered to.
- The list of commonly committed errors can be used as a checklist during code inspection to look out for possible errors.

Code Inspection:

- The following is the checklist of some classical programming errors which can be used during code inspection:
  - Use of uninitialized variables
  - Use of incorrect logical operators or incorrect precedence among operators.
  - Incomplete assignments
  - Jumps into loops
  - Non-terminating loops
  - Array indices out of bounds
  - Improper storage allocation and deallocation
  - Mismatch between actual and formal parameters in procedure calls
  - Dangling reference caused when reference memory has not been allocated

Characteristics of Testable Software

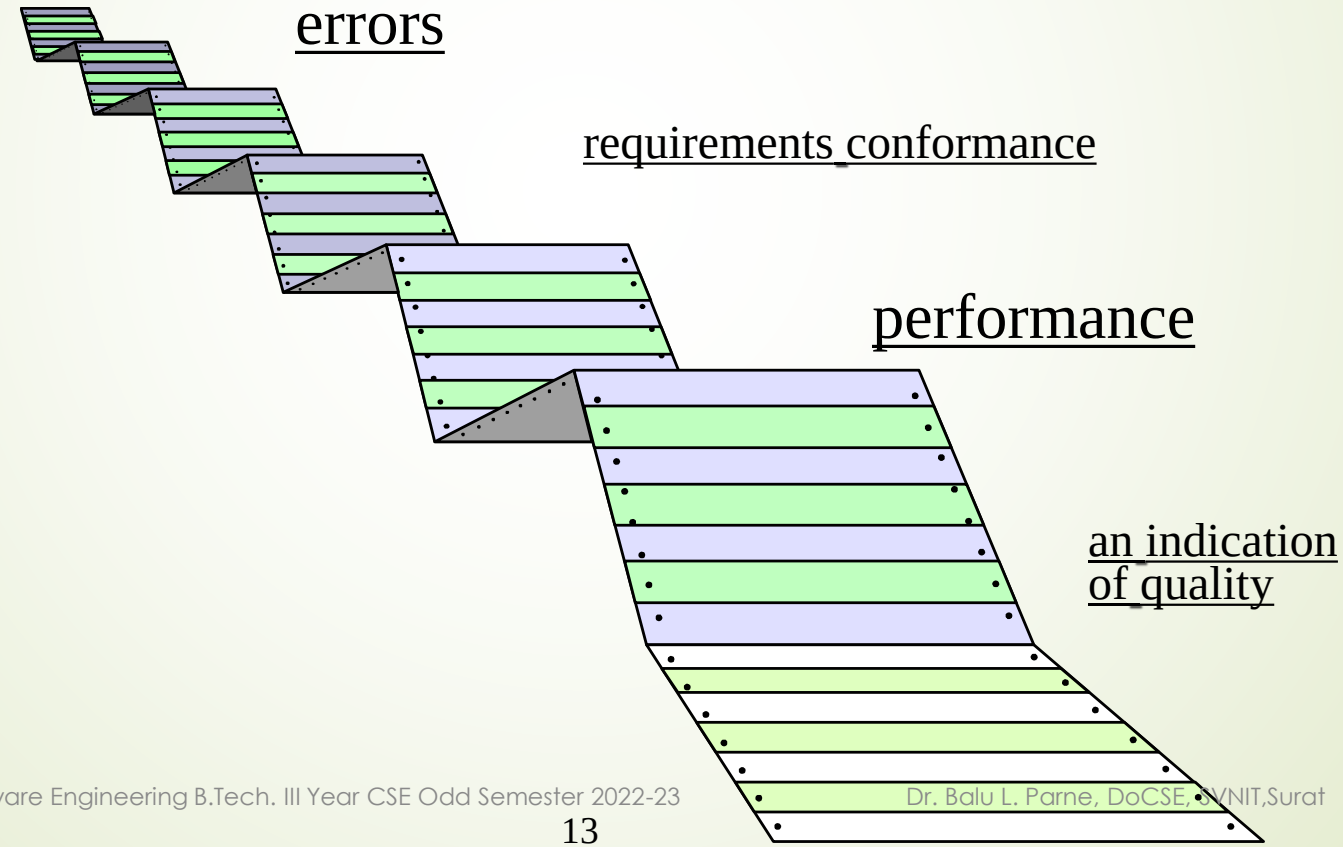
- **Operability**
  - The better it works (i.e., better quality), the easier it is to test
- **Observability**
  - Incorrect output is easily identified; internal errors are automatically detected
- **Controllability**
  - The states and variables of the software can be controlled directly by the tester
- **Decomposability**
  - The software is built from independent modules that can be tested independently

Characteristics of Testable Software

- **Simplicity**
  - The program should exhibit functional, structural, and code simplicity
- **Stability**
  - Changes to the software during testing are infrequent and do not invalidate existing tests
- **Understandability**
  - The architectural design is well understood; documentation is available and organized

Software Testing

Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user.



Who Tests the Software?

developer

independent tester

Understands the system
but, will test "gently"
and, is driven by ***"delivery".***

Must learn about the system,
but, will attempt to break it
and, is driven by "**quality".**

Thank You.