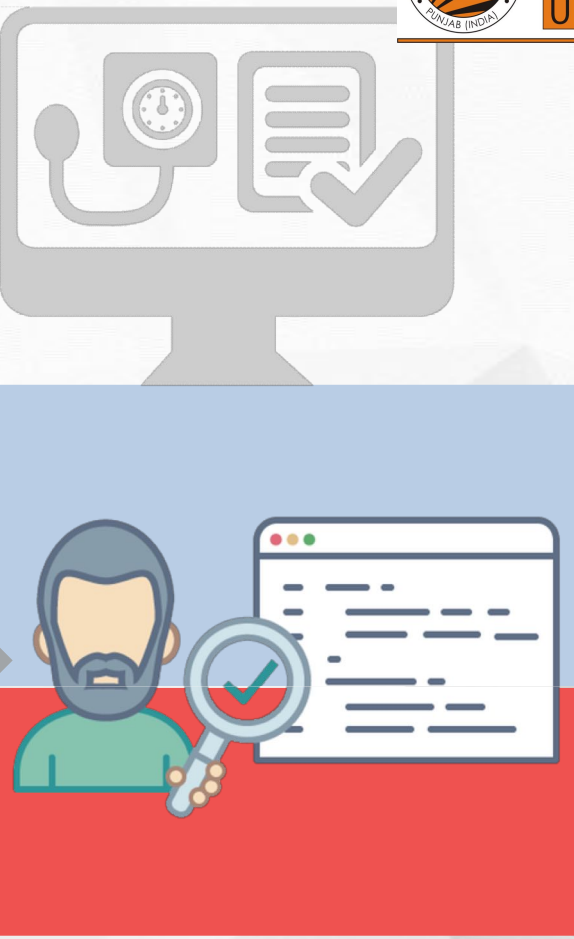
LOVELY PROFESSIONAL UNIVERSITY

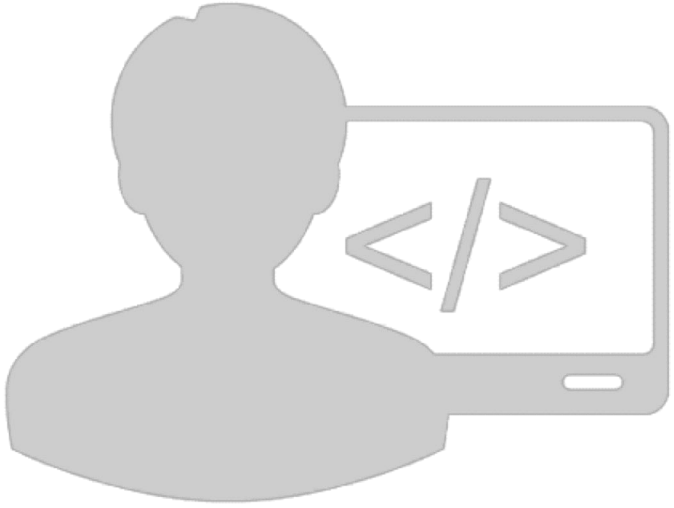
Unit 4.4
Software Engineering

Software Coding & Documentation



Coding Standards

Good software development organizations normally require their programmers to adhere to some well-defined and standard style of coding called coding standards.



Ambler's Law of Standards

- **Industry Standards > organizational standards > project standards > no standards**

- The more commonly accepted a standard the easier it is for team members to communicate
- Invent standards when necessary, butdon't waste time creating something that you won't be able to use later.
- All languages have recommended coding standards available. It is well worth your effort to find and use industry standards
- Push for organizational standards whenever possible

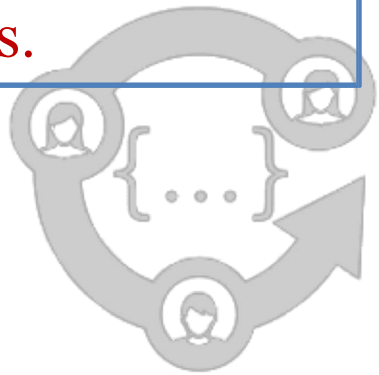
Coding Standards

- Most software development organizations formulate their own coding standards that suit them most, and require their engineers to follow these standards strictly.
- The purpose of requiring all engineers of an organization to adhere to a standard style of coding is the following:

A coding standard gives a uniform appearance to the codes written by different engineers.

It enhances code understanding.

It encourages good programming practices.



Not enforced by compilers

Coding Standards

A coding standard lists several rules to be followed such as, the way variables are to be named, the way the code is to be laid out, error return conventions, etc.

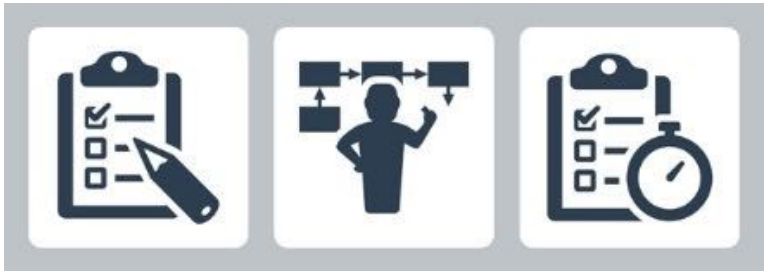
The following are some representative coding standards

1. Rules for limiting the use of global

These rules list what types of data can be declared global and what cannot.

2. Naming conventions for global & local variables & constant identifiers

A possible naming convention can be that global variable names always start with a CAPITAL letter, local variable names are made of small letters, and constant names are always CAPITAL letters.



Coding Standards

3. Contents of the headers preceding codes for different modules

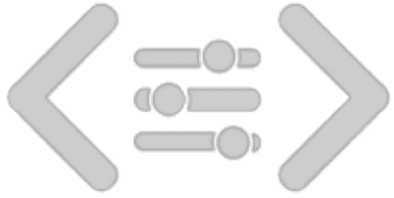
- The information contained in the headers of different modules should be standard for an organization.
- The exact format in which the header information is organized in the header can also be specified.

The following are some standard header data
Module Name, Creation Date, Author's Name
Modification history, Synopsis of the module
Global variables accessed/modified by the module
Different functions supported, along with their input/output parameters

Coding Standards

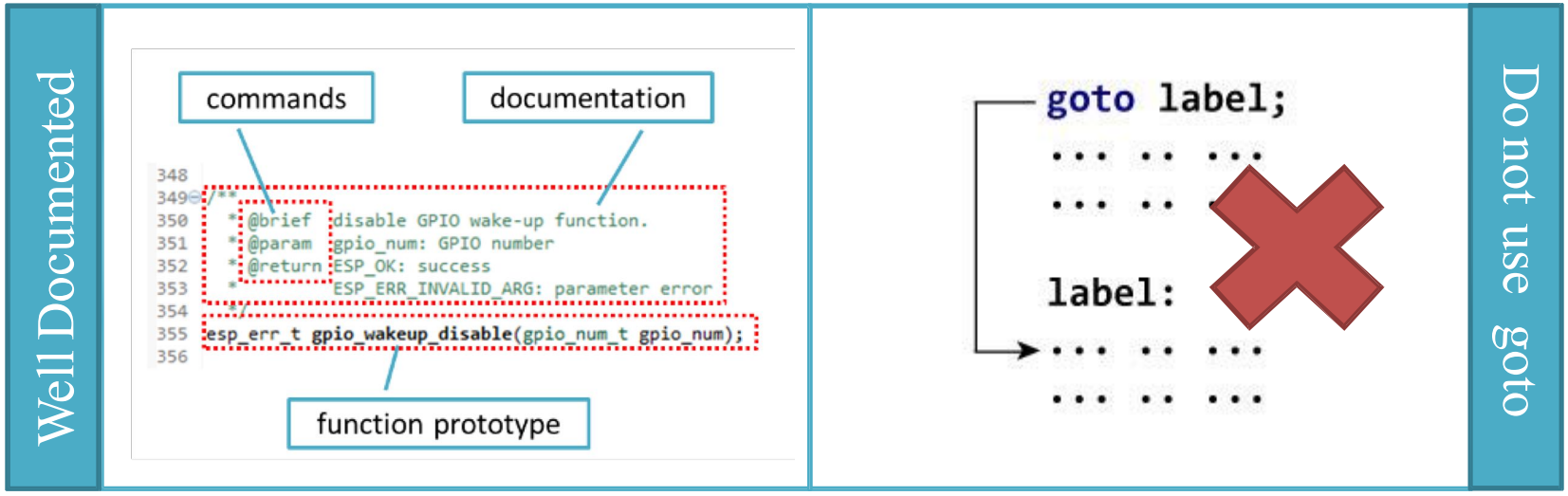
4. Error return conventions and exception handling mechanisms

- The way error conditions are reported by different functions in a program are handled should be standard within an organization.
- For example, different functions while encountering an error condition should either return a 0 or 1 consistently.



Coding guidelines

- The following are some representative coding guidelines
- Do not use a coding style that is too clever or too difficult to understand
- Do not use an identifier for multiple purposes
- The code should be well-documented
- The length of any function should not exceed 20 source lines
- Do not use goto statements



Coding guidelines Cont.

Avoid obscure (unknown) side effects:

- An obscure side effect is one that is not obvious from a casual examination of the code.
- Obscure side effects make it difficult to understand a piece of code.
- For example, if a global variable is changed obscurely in a called module or some file I/O is performed which is difficult to infer from the function's name and header information, it becomes difficult for anybody trying to understand the code.
- The side effects of a function call include modification of parameters passed by reference, modification of global variables, and I/O operations

Coding guidelines

Whenever possible, reuse standards and guidelines, don’t reinvent them

Software Faults

- Quite inevitable
- Many reasons
  - Software systems with large number of states
  - Complex formulas, activities, algorithms
  - Customer is often unclear of needs
  - Size of software
  - Number of people involved

Types of Faults

Algorithmic	Logic is wrong Code reviews
Syntax	Wrong syntax; typos Compiler
Computation/ Precision	Not enough accuracy
Documentation	Misleading documentation
Stress/Overload	Maximum load violated
Capacity/Boundary	Boundary cases are usually special cases
Timing/Coordination	Synchronization issues Very hard to replicate
Throughput/Performance	System performs below expectations
Recovery	System restarted from abnormal state
Hardware & related software	Compatibility issues
Standards	Makes for difficult maintenance

Software Quality

Software Quality remains an issue

Who is to blame?

Customers blame developers

Arguing that careless practices lead to low-quality software

Developers blame Customers & other stakeholders

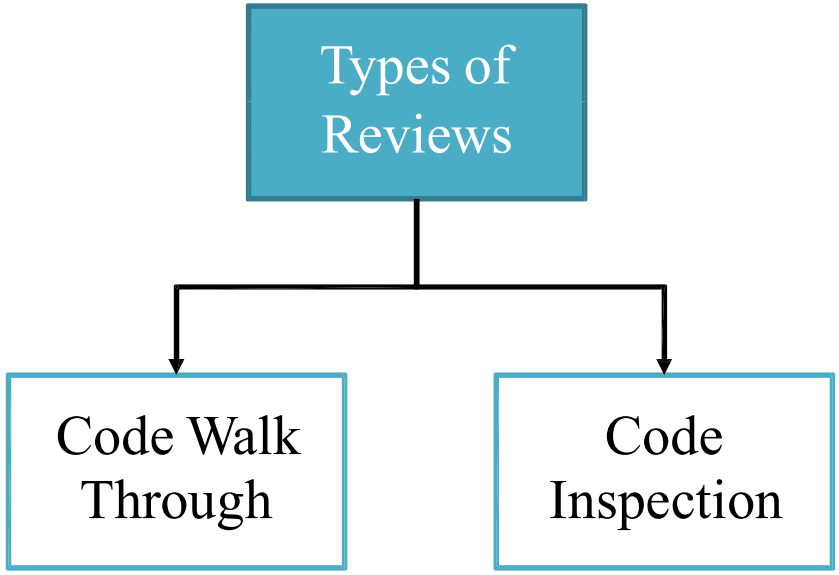
Arguing that irrational delivery dates and continuous stream of changes force the to deliver software before it has been fully validated

Who is Right? Both – and that's the problem

Code Review
Code Walk Through
Code Inspection

Code Review

- Code Review is carried out after the module is successfully compiled and all the syntax errors have been eliminated.
- Code Reviews are extremely cost-effective strategies for reduction in coding errors and to produce high quality code.

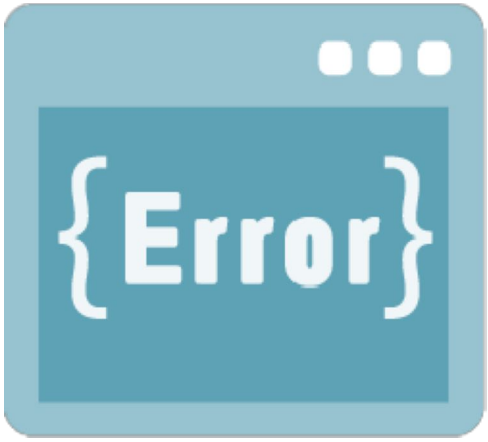


Code Walk Through

- Code walk through is an informal code analysis technique.
- The main objectives of the walk through are to discover the algorithmic and logical errors in the code.
- A few members of the development team are given the code few days before the walk through meeting to read and understand code.
- Each member selects some test cases and simulates execution of the code by hand
- The members note down their findings to discuss these in a walk through meeting where the coder of the module is present.

Code Inspection

- The aim of Code Inspection is to discover some common types of errors caused due to improper programming.
- In other words, during Code Inspection the code is examined for the presence of certain kinds of errors.
  - For instance, consider the classical error of writing a procedure that modifies a parameter while the calling routine calls that procedure with a constant actual parameter.
  - It is more likely that such an error will be discovered by looking for these kinds of mistakes in the code.
- In addition, commitment to coding standards is also checked.



Few classical programming errors

- Use of uninitialized variables
- Jumps into loops
- Nonterminating loops
- Incompatible assignments
- Array indices out of bounds
- Improper storage allocation and deallocation
- Mismatches between actual and formal parameter in procedure calls
- Use of incorrect logical operators or incorrect precedence among operators
- Improper modification of loop variables

What is Documentation ?

To Document each Activity you perform

Is it True ?

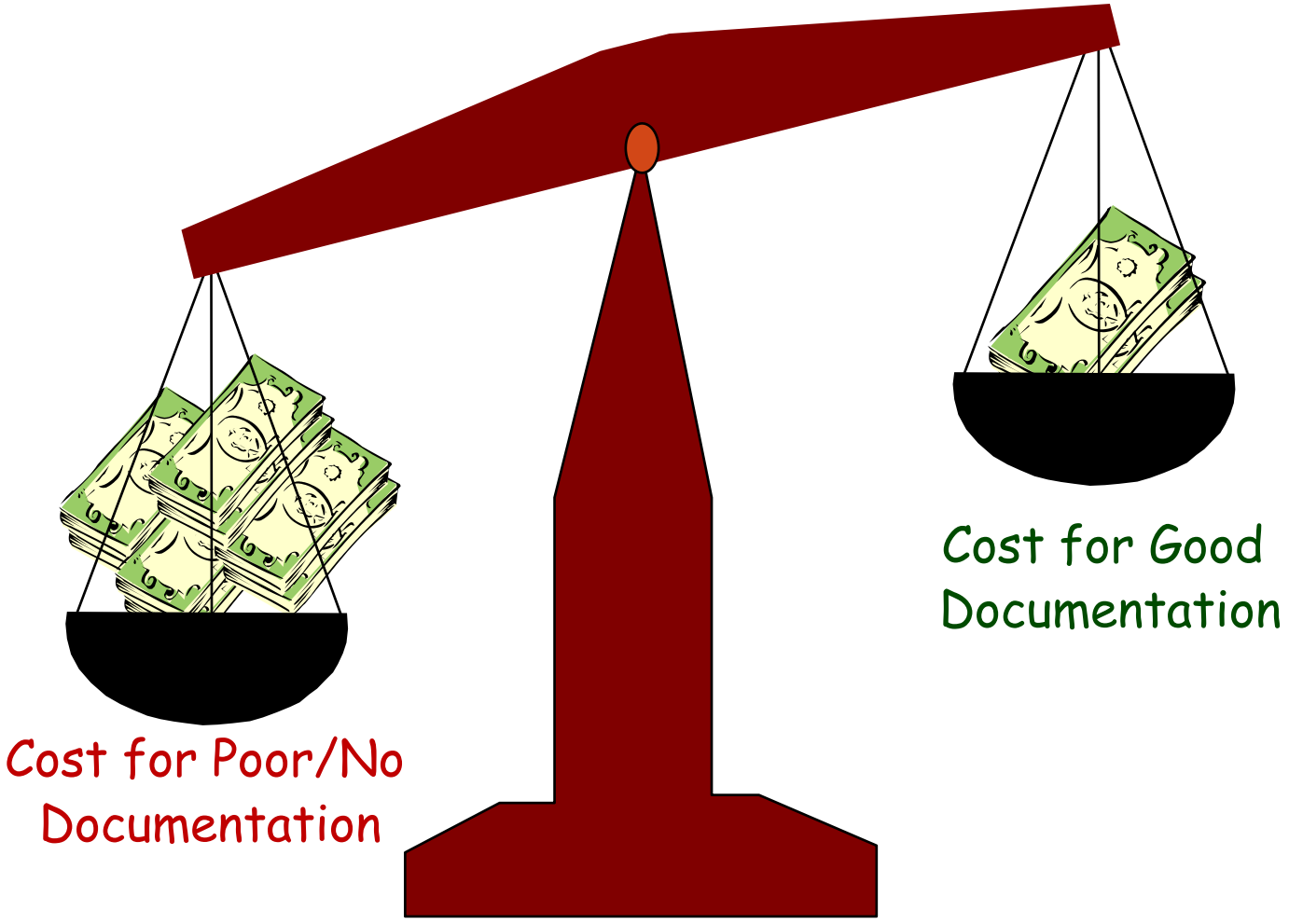
No, Documentation is a Process, Which comprises of Following :

- Recording of Data
- Approval of Documents
- Issuance and Disposal of Documents
- Retrievability of Documents
- Presentation of Documents
- Review of Documents

The purpose of documentation

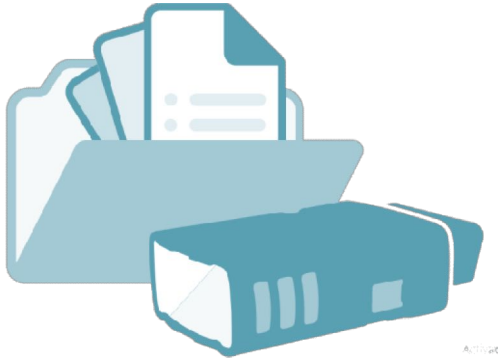
- Proof of fact
- Record
- Regulatory requirement
- Quality maintenance and improvement (Better risk management and robust quality systems)
- Control the processes (Resource optimization)
- "Your documentation is an advertisement for your work"
- End User requirements
- To improve performance
- Enables important messages to be communicated clearly and accurately.

Why Documentation ?



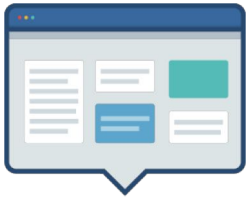
Software Documentation

- When various kinds of software products are developed, various kinds of documents are also developed as part of any software engineering process e.g.
  - Users' manual,
  - Software requirements specification (SRS) documents,
  - Design documents,
  - Test documents,
  - Installation manual, etc
- Different types of software documents can broadly be classified into the following:
  - Internal documentation
  - External documentation



Internal Documentation

- It is the code perception features provided as part of the source code.
- It is provided through appropriate module headers and comments embedded in the source code.
- It is also provided through the useful variable names, module and function headers, code indentation, code structuring, use of enumerated types and constant identifiers, use of user-defined data types, etc.
- Even when code is carefully commented, meaningful variable names are still more helpful in understanding a piece of code.
- Good organizations ensure good internal documentation by appropriately formulating their coding standards and guidelines.



External Documentation

- It is provided through various types of supporting documents
  - such as users' manual
  - software requirements specification document
  - design document
  - test documents, etc.
- A systematic software development style ensures that all these documents are produced in an orderly fashion.