Software Engineering

Coding

Coding phase

- Undertaken when design phase is completed
- Input: design document
  - Module structure (e.g. structure chart)
  - Module specification, e.g. data structures & algorithms for each module
- Coding phase
  - Modules implemented
  - Each module tested in isolation (unit-testing)

Coding standards

- Some standard style of coding
  - Software development organizations require their programmers to adhere to the standard
  - Often formulated by individual software development organizations
  - Advantages :
    - Gives a uniform appearance to the codes written by different engineers.
    - It enhances code understanding.
    - It encourages good programming practices.

Examples of coding standards

- Rules for limiting the use of global variables
  - What type of data can be declared global
- Naming conventions for global variables, local variables, constant identifiers
- Contents of headers for each module
  - Exact format of header usually specified, e.g.
  - Name of module, date of creation, programmer's name, modification history, synopsis, ...
  - Functions supported, along with input / output parameters
  - Global variables accessed / modified by module

Examples of coding standards

- Error return conventions and exception handling mechanisms
- Code should be well-documented
- Things to avoid
  - Cryptic and difficult to understand coding style – it causes debugging difficult and expensive
  - Using an identifier (variable name) for multiple purposes
  - Obscure side effects, e.g. A function modifying global variables or parameters passed by reference
  - Goto statements
  - Very big function body

Code inspection

- Carried out after a module has been coded
- Objectives
  - Discovery of commonly made general errors
  - Ensure adherence to coding standards

- Code examined for presence of common errors
  - Writing a procedure that modifies a parameter
  - Infinite loops, modification of loop control variables inside loop, array indices out of bounds, use of uninitialized variables, forgetting to de-allocate memory, …

Code inspection contd...

- Some classical programming errors which can be checked during code inspection
  - 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.
  - Comparison of equality of floating point variables.

Code walkthrough

- Informal code analysis technique
  - Select some test cases and simulate execution of the code by hand for these test cases
  - Errors detected as early as possible

- Guidelines for code inspection / walkthrough
- Usually done by members of development team itself, manager usually not present

Idea is to attack the product to detect defects in advance, but not the developer.

Software documentation

- Various documentations developed in addition to source code and executable files
- Why need good documents ?
  - Enhance understandability and maintainability of a software product
  - Help in case of manpower turnover
- Two classes of documents
  - Internal documents
  - External / supporting documents

Internal documentation

- Provided within the source code itself
  - Meaningful variable names
  - Code indentation, code structuring
  - Use of enumerated data types & constant identifiers
  - Use of user-defined data types
  - Module / function headers and comments

- Good internal documentation is suggested in coding standards

External documentation

- Documentation other than those present in the source code
  - SRS, design document, test document
  - Users' manual, installation manual
- Consistency of documents with each other
  - Consistency of terminology
  - All documents should be up-to-date
- Textual documents should be readable
  - Readability determines understandability, maintainability

After coding phase

- After all modules of a system have been coded and unit tested
  - Integration of modules according to an integration plan
  - During each integration step, a number of modules added to the partially integrated system and system tested again

- Once all modules have been integrated and tested, system testing
  - Fully integrated system is tested against the requirements recorded in the SRS document