Coding

Code Review

Code review for a model is carried out after the module is successfully compiled and the 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.

In general, two types of reviews are carried out on the code of a module-

- Code walkthrough
- Code Inspection

Code Walkthrough

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.

After a module has been coded, successfully compiled and all syntax errors eliminated, 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 (i.e. trace execution through each statement and function execution).

The members note down their findings to discuss these in a walk through meeting where the coder of the module is present.

Although, it is an informal analysis technique, several guidelines have evolved over the years for making this naïve but useful analysis technique more effective. These guidelines are based on personal experience, common sense, and several subjective factors.

Code Walkthrough

Walkthrough Guidelines:

- The team performing code walk through should not be either too big or too small. Ideally, it should consist of between three to seven members.
- Discussion should focus on discovery of errors and not on how to fix the discovered errors.
- In order to foster cooperation and to avoid the feeling among engineers that they are being evaluated in the code walk through meeting, managers should not attend the walk through meetings.

Code Inspection

The aim of code inspection is to discover some common types of errors caused due to oversight and improper programming.

During code inspection the code is examined for the presence of certain kinds of errors, in contrast to the hand simulation of code execution done in code walk throughs.

The classical error of writing a procedure that modifies a formal 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, rather than by simply hand simulating execution of the procedure.

In addition to the commonly made errors, adherence to coding standards is also checked during code inspection.

Good software development companies collect statistics regarding different types of errors commonly committed by their engineers and identify the type of errors most frequently committed. Such a list of commonly committed errors can be used during code inspection to look out for possible errors.

Code Inspection

Following is a list of 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 equally of floating point variables, etc.

Clean Room Testing

- Clean room testing was pioneered by IBM. This type of testing relies heavily on walk throughs, inspection, and formal verification.
- The programmers are not allowed to test any of their code by executing the code other than doing some syntax testing using a compiler. The software development philosophy is based on avoiding software defects by using a rigorous inspection process.
- The name ‘clean room’ was derived from the analogy with semi-conductor fabrication units. In these units (clean rooms), defects are avoided by manufacturing in ultra-clean atmosphere.
- In this kind of development, inspections to check the consistency of the components with their specifications has replaced unit-testing.

Clean Room Testing

The clean room approach to software development is based on five characteristics:

Formal specification: The software to be developed is formally specified. A state transition model which shows system responses to stimuli is used to express the specification.

Incremental development: The software is partitioned into increments which are developed and validated separately using the clean room process. These increments are specified, with customer input, at an early stage in the process.

Structured programming: Only a limited number of control and data abstraction constructs are used. The program development process is process of stepwise refinement of the specification.

Static verification: The developed software is statically verified using rigorous software inspections. There is no unit or module testing process for code components

Statistical testing of the system: The integrated software increment is tested statistically to determine its reliability. These statistical tests are based on the operational profile which is developed in parallel with the system specification.

Software Documentation

When various kinds of software products are developed then not only the executable files and the source code are developed but also various kinds of documents such as users' manual, software requirements specification (SRS) documents, design documents, test documents, installation manual, etc are also developed as part of any software engineering process. All these documents are a vital part of good software development practice.

Good documents are very useful and serve the following purposes:

- Good documents enhance understandability and maintainability of a software product. They reduce the effort and time required for maintenance.
- Use documents help the users in effectively using the system.
- Good documents help in effectively handling the manpower turnover problem. Even when an engineer leaves the organization, and a new engineer comes in, he can build up the required knowledge easily.
- Production of good documents helps the manager in effectively tracking the progress of the project. The project manager knows that measurable progress is achieved if a piece of work is done and the required documents have been produced and reviewed.

Software Documentation

Different types of software documents can broadly be classified into the following:

Internal documentation is the code comprehension features provided as part of the source code itself. Internal documentation is provided through appropriate module headers and comments embedded in the source code. Internal documentation 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.

External documentation 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.