Implementation and Integration

- Implementation
- Unit Testing
- Integration
- Integration Approaches
Implementation Phase

• Implementation Phase involves writing actual code to implement system requirements according to the Design

• It has two distinct parts:
  • writing programs that implement the design.
  • conducting unit tests.
Characteristics of Good Programming

• Here we explain some of the software engineering practices that you should keep in your mind as you write your code.

• Naming Conventions and Coding Standards – Does the organization have its own naming conventions and coding standards? Does the programmer follow these naming conventions and coding standards?

• Reusable code – Does the programmer follow the patterns of the design so that the code is reusable?

• Documentation – Does the programmer write internal and external documentations which are clear, crisp, and understandable?
The Techniques of Unit Testing

- Unit testing is an integral part of Implementation Phase.
- The unit testing process has two techniques:
  - Static Testing
  - Dynamic Testing
Static Testing

- Static Testing does not require the system to be executed.
  - It checks/reviews Software Requirements Specification, Design documents (both architectural and detailed design), the program source code.
  - It checks for traceability.
  - It checks for the coding standards and naming conventions.
  - It checks whether the design methodology follows the OO principles.
  - It checks programs for the unused or overused program segments or variables.
Dynamic Testing

• Dynamic Testing requires the system to be executed.

  • It checks the prototypes and the program source code.

  • It checks for logical errors.

  • It checks for data validation (what will happen if the age in the input screen is given a negative integer?).

  • It checks non – functional characteristics such as performance, reliability, scalability, security etc.
What is Integration?

• During the *integration phase*, the various component modules are combined to build the software product. The modules may be integrated using top-down approach or bottom-up approach. Suppose the product is integrated bottom-up. If there is a major design fault, then it will show up late, necessitating an expensive rewrite. Conversely, if the modules are integrated top-down, then the lower level modules usually will not receive as thorough a testing as would be the case if the product were integrated bottom-up.

• Integration testing, system testing and acceptance testing are integral part of Implementation Phase.
What is Integration? (cont’d)

• The purpose of integration testing is to check that the modules combine correctly to achieve a product that satisfies its requirements.

• When integration testing is completed, system testing is performed. The functionality of the product as a whole (performance, security, robustness, safety, maintainability, scalability etc.) is checked against the specifications.

• The final aspect of this phase is acceptance testing. The software is delivered to the client, who tests the software on the actual hardware using actual data, as opposed to test data (sample data used by the developers).