Coding

Coding

- Introduction
- Programming Principles
- Coding Conventions

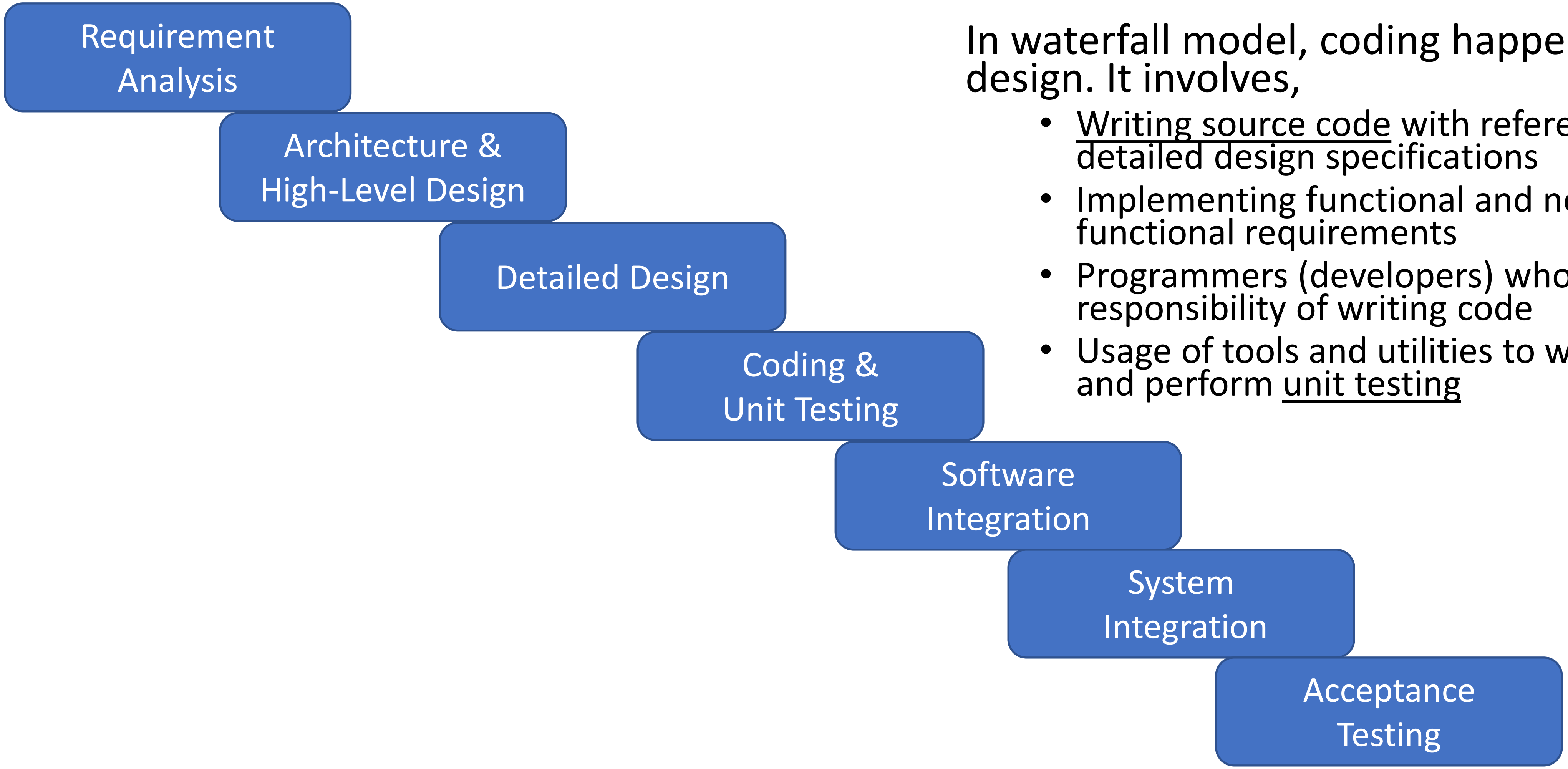
Coding

Coding refers to writing programs (program logic, data base script, data base procedures, and related configuration files)

Important:

1. *Write programs that work properly and satisfy functional, non-functional and interface requirements.*
2. *In addition, write standard code by following coding standards and code optimization guidelines or best practices.*

Waterfall Model (devised by Royce in 1970)



In waterfall model, coding happens after design. It involves,

- Writing source code with reference to detailed design specifications
- Implementing functional and non-functional requirements
- Programmers (developers) who take the responsibility of writing code
- Usage of tools and utilities to write code and perform unit testing

Advantages of writing standard code

- **Ease of Understanding:** It is easy to understand when all programmers write standard code
- **Maintainability:** It is easy to make enhancements or changes
- **Ease of Fixing Defects:** It is easy to identify the location of defects (bugs) and fix them.
- **Development Cost:** Can reduce the development cost (less rework, ease of training, avoids conflicts)

Programming – Art or Science?

- Programming is science because it involves engineering principles and guidelines.
- It can also be considered as art because there are opportunities for programmers to be creative and innovative.

Programming Principles

- **Validity** - Should give correct results for all possible inputs.
- **Consistency** – It should be consistent always in terms of behavior and results
- **Maintainability** – Should take less time (and cost) to fix defects(bugs) or add new features (should have proper internal documentation (or comments) and external documentation (program specification, technical reference))
- **Readability** – Programs should be easy to read and understand (should have internal documentation (or comments))
- **Usability** – Programs must be usable for the intended purpose without any challenges

Programming Guidelines

There are programming guidelines that are common across programming languages. For example,

1. Follow coding standards and write standard code
2. Make the code readable and easy to understand by including comments
3. Whenever you make changes to source code, make corresponding changes to comments as well
4. Follow design specifications
5. Keep the modules or programs small because large programs or modules are very challenging to maintain

Coding Conventions

Coding conventions are a set of rules or best practices or guidelines you agree to follow in your project. Adhering to coding conventions provide the following advantages.

1. Writing reusable code
2. Ease of maintenance and hence low maintenance costs
3. Ease of defect identification and defect fixing
4. Ease of understanding (for new team members)
5. Ease of knowledge transfer during handover (from an existing team member to a new team member)

Coding Conventions – Key Factor

1. Naming Convention (folder name, file name, variable name, function name, method name, …) – Use easy-to-understand but not extremely long names instead of short names such as a, b, and c.
2. Understand the 'letter case' because some naming conventions require 'upper case' (E.g, CUSTOMER) where as other naming conventions permit 'Customer' or 'customer'
3. Multiple Word Identifiers
    a) Some coding conventions allow multiple words with delimiters because blank space is not allowed by most programming languages. (E.g., book_ticket ( ), cancel_ticket ( ))
    b) Some other coding conventions allow names such as 'BookTicket ( ), CancelTicket ( ))

Naming Conventions - Benefits

1) Naming conventions make the program self-explanatory by providing information about a variable or function or method in its name (E.g. Calculate_Discount ( ))
2) Improves consistency among all developers in a project
3) Provides clarity
4) Gives a professional touch to the source code
5) Helps avoid situations where two programmers use the same name for different purposes
6) Ease of understanding

Rule of Thumb

Rule of Thumb refers to programming practices learnt through experience. Senior programmers in IT organizations follow various ‘rules of thumb’ or best practices derived based on their experience.

Examples:

1. Choose an appropriate data type to suit the maximum and minimum value of a variable or parameter
2. Avoid declaring all variables as global variables
3. Close data base connections right after use.

Adding Comments in Source Code

1. Comments are not compiled. However, comments improve readability.
2. Comments are consumed by document generation tools to generate a comprehensive document on source code
3. Add clear comments when you write code.
4. Update the comments whenever you make changes to code so that the comments are not outdated.
5. Use complete sentences to write comments. Use uniform style and structure.

Using Blank Lines and White(Blank) Spaces

- Two Blank Lines – To separate two sections or two functions or methods or interfaces
- One Blank Line – To separate sections within a function or method or interfaces
- Blank Space (or White Space) – Use it after a comma when you specify a list of parameters or arguments. (E.g., (a, b, c)),
- Use blank space when you use binary operators (E.g., a + b instead of a+b or a+ b or a +b) (Binary operator is an operator that has two operands)

Code Reviews

Formal Inspection (E.g. Fagan Inspection) - Very structured approach based on a review checklist. Multiple reviewers take part and review line by line. It is a structured process of finding defects in code. Review comments are recorded and tracked to closure.

Lightweight Walkthrough (E.g. Peer Review) – This review is not as formal and structured as formal inspection. A peer (another developer in the project) reviews the code, and shares observations.

Pair Programming (Agile – Extreme Programming Practice) - Two developers sit together and write code. Review happens in real time.

Automated Code Reviews (E.g., Static Analysis) – No human intervention. A software performs the review. Static Analysis tools review the source code without executing it and report issues in the code (such as unused variable, unused packages, and other violations)

Question 1

Programmers need to write code in such a way that,

A. Coding standards and conventions are followed
B. Functional and non-functional requirements are implemented in code
C. The source code has adequate comments for ease of understanding
D. All of the above

ANSWER: D

Question 2

Which of the following is not a benefit of writing standard code?

A. High probability of writing defect free code
B. Ease of adding new features or making changes to existing features
C. Ease of knowledge transfer when a new team member is replacing an existing team member
D. Ease of understanding

ANSWER: A

Question 3

What is the advantage of code reuse?

A. It can save time, reduce the number of lines in the code and minimize coding efforts

B. It can eliminate the need for testing

C. It ensures layering and low coupling

D. None of the above

ANSWER: A

Question 4

Which of the following types of code review is the fastest of all?

A. Fagan Inspection
B. Peer Review
C. Static Analysis
D. None of the above

ANSWER: C

Question 5

What is the main advantage of following naming conventions while coding?

A. Eliminates the need of writing comments in code

B. Improves the performance of code

C. They use words in English

D. Programs that follow naming conventions are self-explanatory because they are easy to understand

ANSWER: D

Summary

- Introduction
- Programming Principles
- Coding Conventions

Thank You!