



QA Testing Boot Camp

Chapter 1 - SDLC

Learning Outcomes

- Understand SDLC
- Understand Models
- Model Selection
- Costs Associated with/without Modeling

SDLC Overview – Types of Models

Classic Models - Waterfall Model

- Development is seen as flowing steadily downwards (like a waterfall) through the various phases
- More later...

SDLC Overview – Types of Models

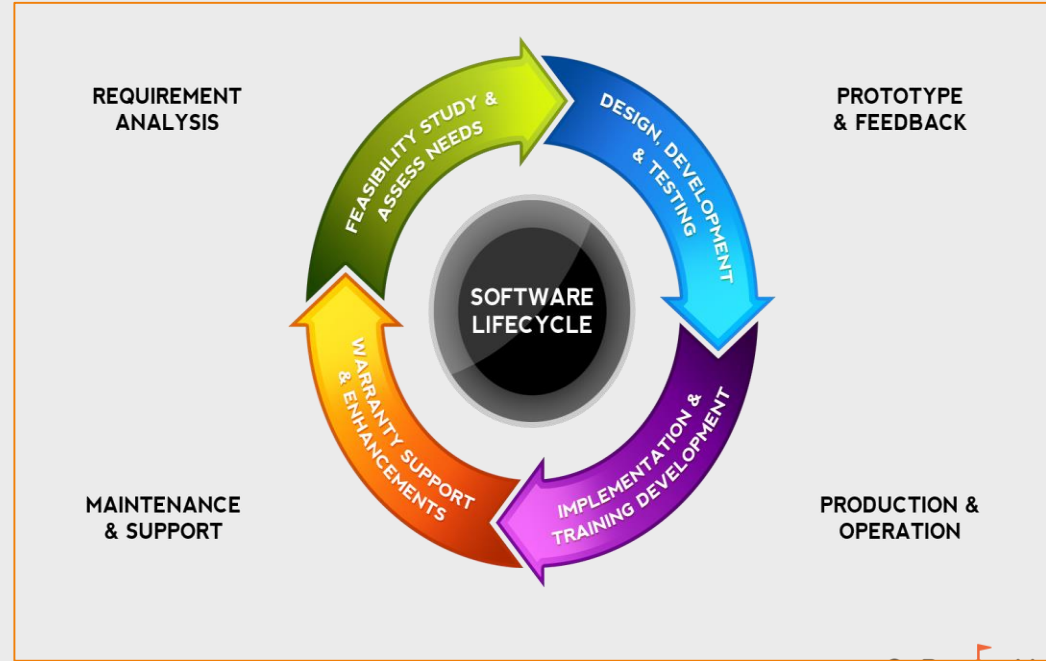
Incremental Models

- Software is built vs. written
- Software is designed, implemented, integrated and tested as a series of incremental builds

SDLC Framework

Framework

A framework, describes those activities that are performed at each phase during a software development project



Project Life Cycle – Needs Assessment

Types of Problems

- **Well-structured/Structured** -- constrained problems with convergent solutions, limited number of rules and principles within well-defined parameters.
- **Unstructured** -- multiple solutions, fewer parameters, and contain uncertainty about which concepts and rules.

Project Life Cycle – Entry and Feasibility Study

Purpose of this phase is to obtain a commitment to change (Entry) and to evaluate whether cost effective solutions are available to address the problem or opportunity that has been identified.(Feasibility)

Scenario 1

Users identify a system problem. They believe that they can design and implement a solution on their own. The proposed system will have little impact on others within the organization, and it will be transparent to the overall organization. In this scenario, the users are sufficiently motivated to bring about change.

Scenario 2

the solution to the identified issue will have a widespread impact on the overall organization. Under these circumstances, entry criteria are now critical. Information systems professionals must seek to establish themselves as legitimate change agents. Moreover, they must seek to foster a commitment to change among the stakeholders. If potential solutions will have a significant impact on task and social systems, a spirit of collaborative analysis and evaluation among stakeholders will be essential.

Feasibility

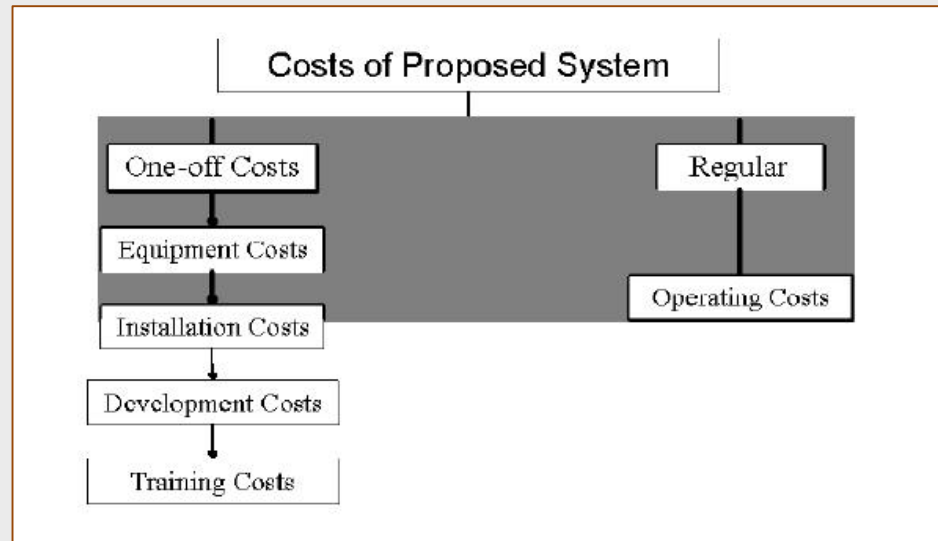
- Cost
- Benefits
- Team
 - Representative knowledge
 - Programmers, SA, Designers Users, Managers

Technical Feasibility

- Response time and volume
- Volume of transactions processed within the given time.
- File Capacity.
- Number of users supported.
- Organizational Adaptability .
- Behavioral feasibility
- Reduction in job stress
- Quality of output by employee
- Can the system handle the input data in its current state?
- Is current output usable?
- Economic feasibility including a cost/benefit analysis.

Project Life Cycle – Cost of Proposed System

Management should consider the impact and amount of proposed benefits



Project Life Cycle vs. SDLC

Project Life Cycle

The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases.

SDLC – Software Development Life Cycle

The overall process of developing information systems through a multi-step process from investigation of initial requirements through analysis, design, implementation and maintenance.

SDLC

There are following six phases in every Software development life cycle model:

- 1.Requirement gathering and analysis**
- 2.Design**
- 3.Implementation or coding**
- 4.Testing**
- 5.Deployment**
- 6.Maintenance**

1. Requirement Analysis and Design

- Business Requirements are gathered
 - Meetings with Stakeholders
 - Managers
 - Users
 - Designers/Programmers/Analysts
 - Output: Requirements Specification
 - Basis for rest of project

SRS

software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements

1. Requirement Analysis and Design

1.1 Needs Assessment

1.2 Entry and feasibility studies

1.3 Analysis of the existing system

- Though knowledge and understanding
- Consider organizational culture

2. Design Overview

- Software design from Specifications
- Hardware and system requirements
- System Architecture
- Formulation of strategic requirements
- Organizational and Job Design
- Output: System Design Specifications

3. Implement/Code

- Work divided into modules/units
- Actual coding is started
- Longest phase

4. Testing

- Requirement Testing
- Unit Testing
- Integration Testing
- System Testing

5. Deployment

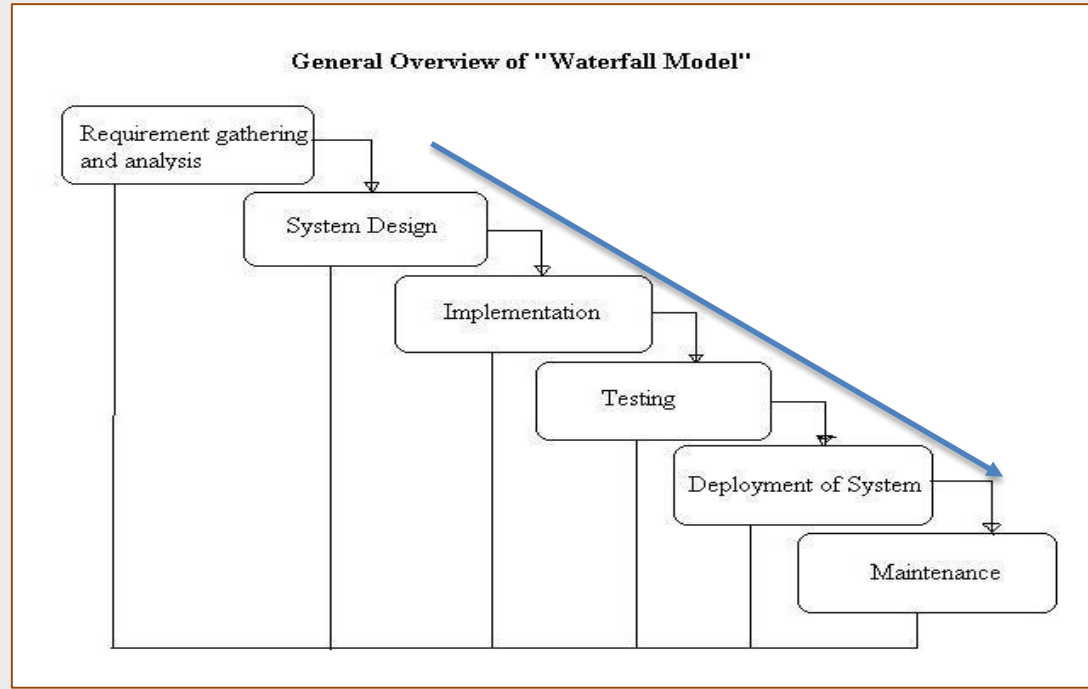
- Testing phase passed
- System is delivered/deployed
 - Beta testing
 - Parallel testing

6. Maintenance

- After Deployment
 - Discovered problems
 - Enhancements/Changes

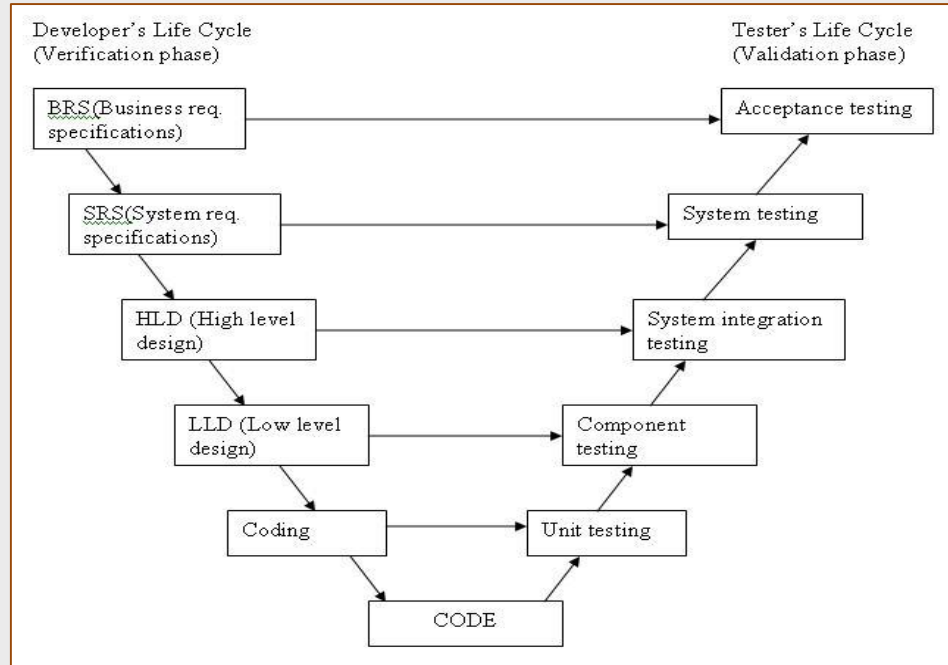
Development Models – Classic Waterfall

- Earliest of all models
- Cascade of phases
- Top-Down approach



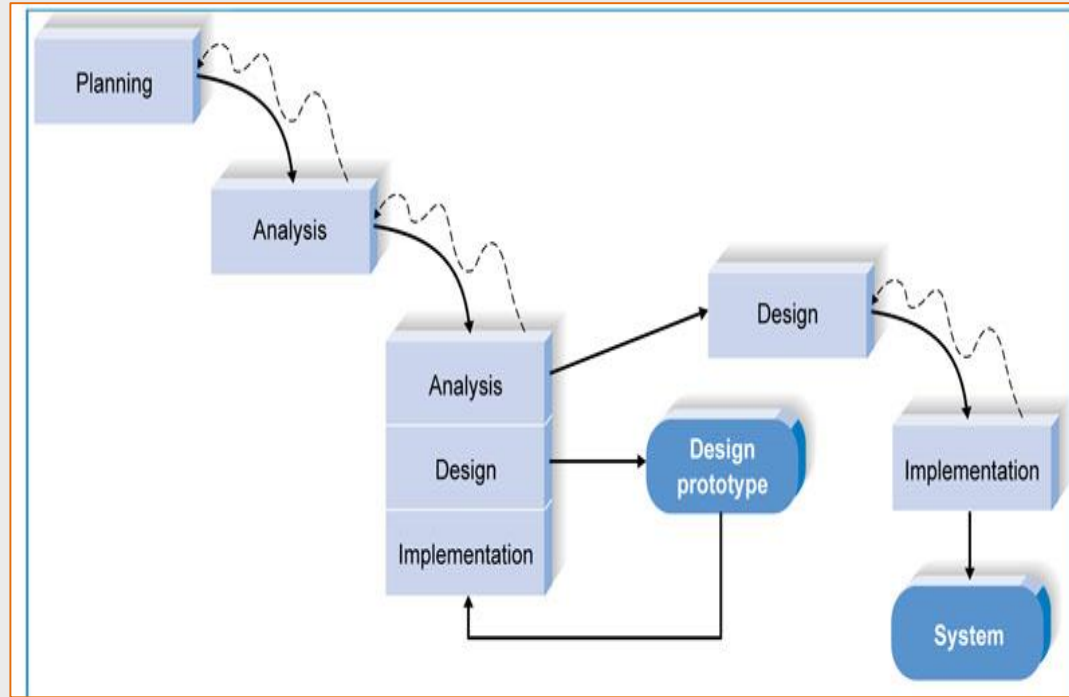
Development Models – V-Shaped

- V & V Model (Verification & Validation)
- Variant of the Waterfall Model
- Testing is planned with each phase



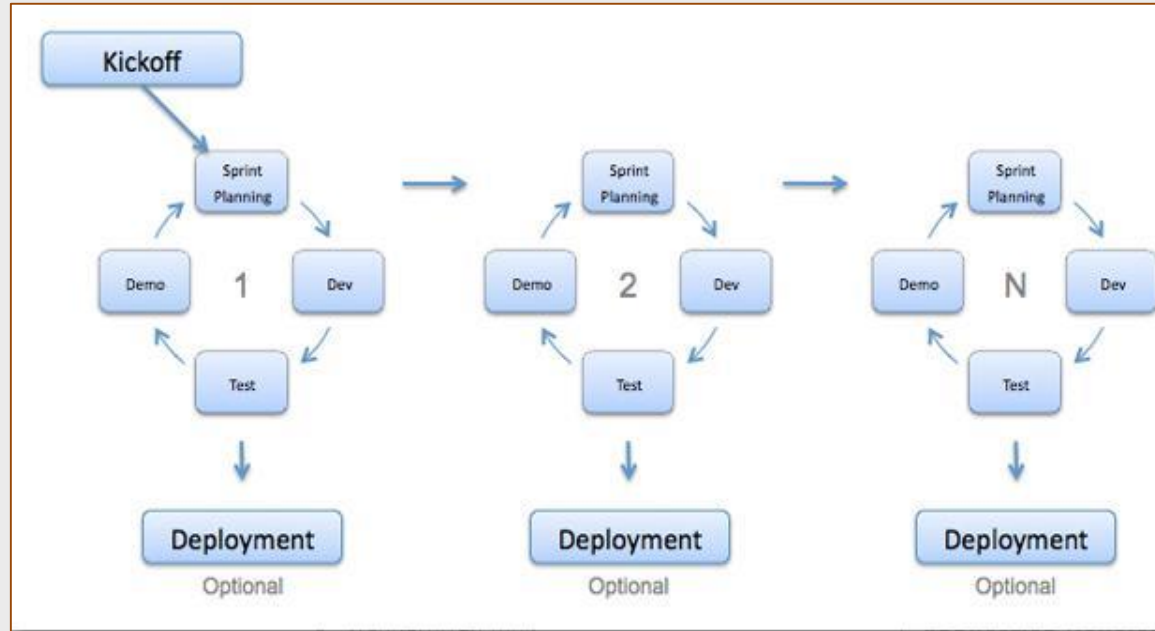
Development Models - Prototyping

- Prototype built to understand requirements
- Based on what is currently known
- Attractive for large complex systems



Development Methodology - Agile

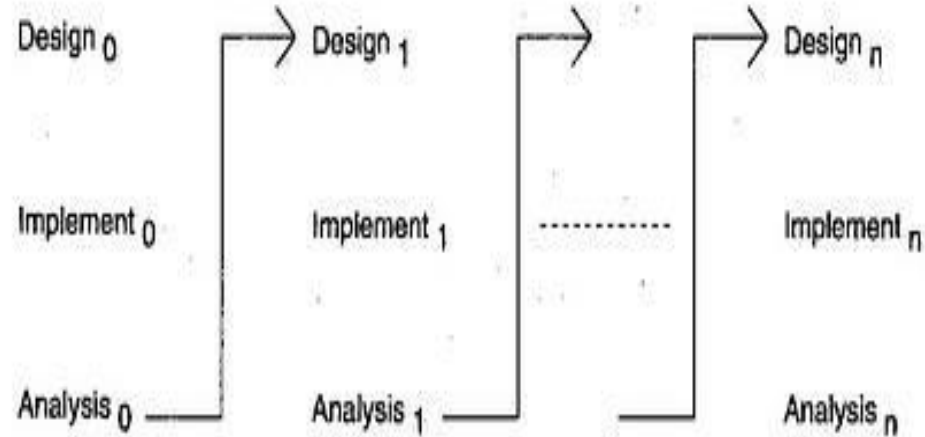
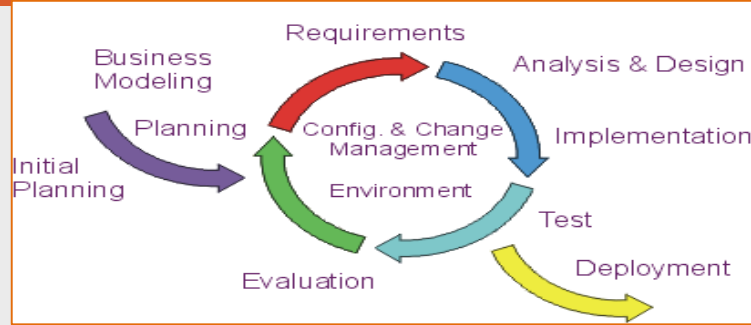
- Incremental Model
- Developed in very fast increment cycles
- Small, continuous releases
- Each release tested



Development Models - OOD

Object Oriented Development (OOD)

- Iterative Model
- Does not start with full requirements
- Iterations are planned
- Repetitive process until fully developed



Development Models - RAD

Rapid Application Development

- Incremental Model
- Mini-Projects – time boxed
- Whole system is develop in parallel

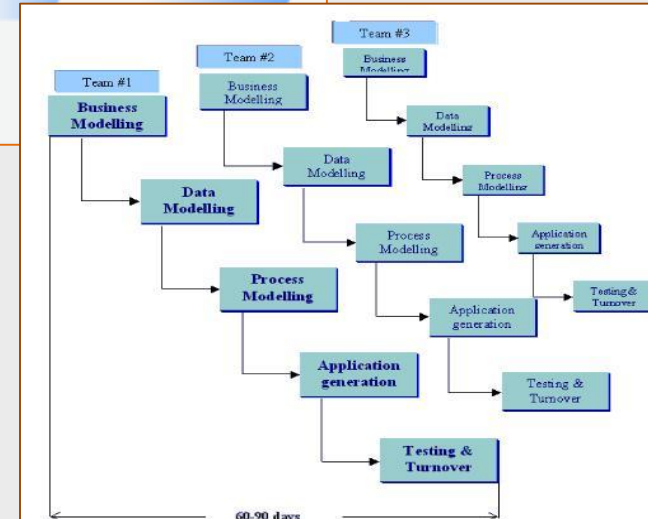
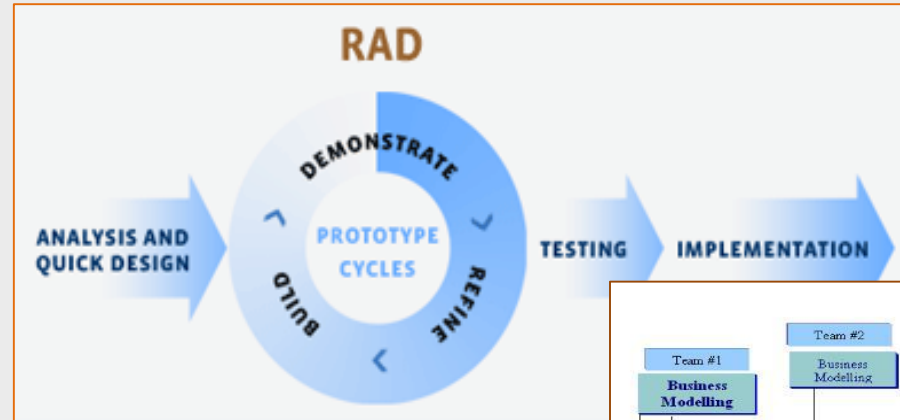
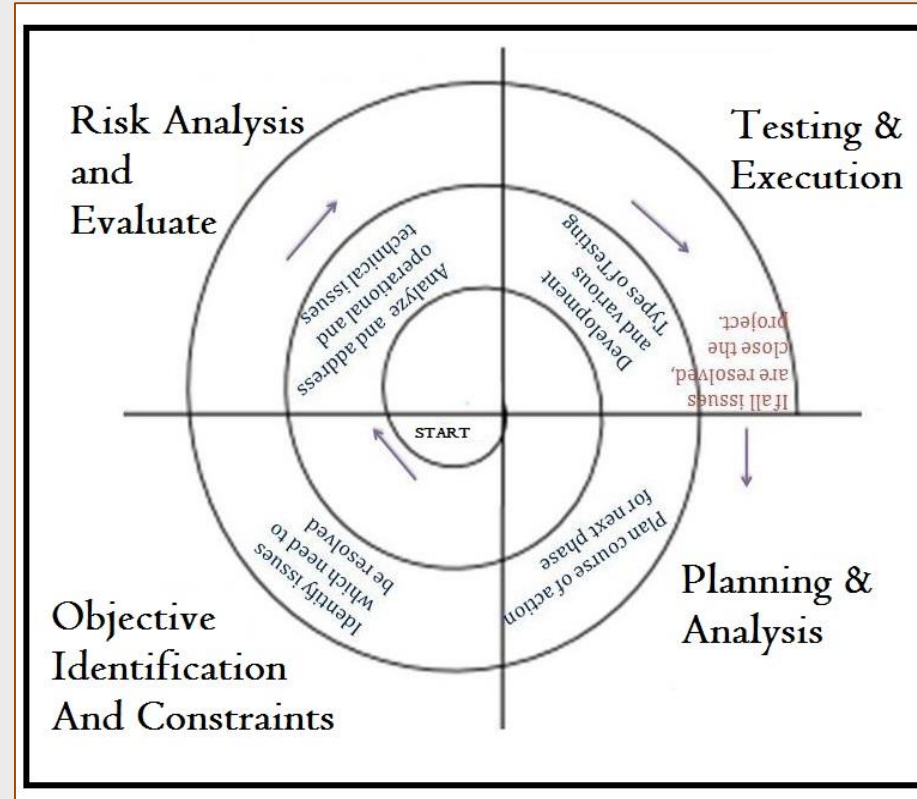


Figure 1.5 – RAD Model

Development Models – Spiral Model

- Similar to Incremental
- More emphasis placed on risk analysis
- Project repeated passes through each phase in iterations (Spirals) until completion



Standards

- SQA – [Software quality assurance IEEE 730](#)
- SCM – [Software configuration management IEEE 828](#)
- STD – [Software test documentation IEEE 829](#)
- SRS – **Software requirements specification IEEE 830**
- V&V – [Software verification and validation IEEE 1012](#)
- SDD – [Software design description IEEE 1016](#)
- SPM – [Software project management IEEE 1058](#)
- SUD – [Software User Documentation IEEE 1063](#)

Summary

In this chapter, we took a brief look at the Software Development Life Cycle (SDLC) which included touching on a few of the most commonly used methodologies. We looked at what the SDLC is, and is not, and how using a methodology can actually save costs in the long run.

Chapter Assessment

1. List the advantages and disadvantages of the V&V Model.
2. When would you choose to use the RAD Model?
3. Explain in detail what benefits are achieved with the Prototyping Model.
4. What the differences between the Spiral Model and the Water Fall Model?
5. What is meant by the cost of change when considering the Traditional Methods vs. the Agile Methods?

The background image shows a bright, modern office space. In the foreground, there is a large, round wooden table with several white chairs. In the background, there is a long wooden desk with red chairs, and large windows that look out onto a city skyline. The ceiling has exposed pipes and several hanging light bulbs.

QA Testing Boot Camp

Chapter 1 - SDLC