Chapter 3

System Development Methodologies
Systems Development Methodologies

- Evolution of the methodologies
- Some popular methodologies
- Phased development
- Selecting the right methodology
- Methodology goals
Evolution of Methodologies

Recommended way of doing something:

- Observe
- Formulate a Hypothesis
- Predict
- Test
Evolution of the Methodologies

Fig 3.1: The Evolution of Methodologies

- Phased development
- Rapid application development (RAD)
- Prototyping
- The traditional system development life cycle
- The systems approach
- The scientific method
Some Popular Methodologies

The traditional system development life cycle

- The “waterfall cycle” stages
- Circular pattern

Fig 3.2: The System Life Cycle as a Sequence of Stages and a Repetitive Process
Prototype

- A prototype is a model that contains the essential elements of an object to be produced in the future, and is used as a pattern.

- Types of Prototype
  - Evolutionary prototype
  - Requirement prototype

- Prototype Tools
  - Integrated application generators
    - Menu, reports, screens etc.
  - Toolkits
    - Report generators, screen generators, DBMS, spreadsheets, CASE/Tools
Some Popular Methodologies

Prototyping

- Evolutionary prototype

Fig 3.3: An Evolutionary Prototype

P  Preliminary investigation

A  Analysis

User review

D  Design prototype

User approves prototype

C  Construct prototype

Complete system components and install prototype

Cutover: Put system in production
Some Popular Methodologies

Prototyping

• Requirements prototype

Fig 3.4: A Requirements Prototype
Strengths of Prototype

- Enhanced communication with user
- User needs easier to determine
- Errors are detected earlier
- User plays an active role in system development
- Less time and effort are required
- Implementation is much easier
Weaknesses

- Haste to deliver the prototype may produce shortcuts in analysis, solution evaluation, testing, and documentation
- Users may have unrealistic expectation
- Evolutionary prototype after a production system may not be that efficient
- May not reflect good design
Rapid Application Development

- Information Engineering is a top-down methodology for developing systems that begins with enterprise planning and strategic planning for information resources.
- RAD is a life cycle strategy that is intended to provide much faster development and higher-quality results.
RAD Stages

- Requirement Planning
  - Users work with systems developers to identify business problems to be solved

- User Design
  - Users play a big role in the non-technical design of new systems assisted by system developers

- Construction
  - Developers use special software tools to develop prototypes, which are reviewed by users for further refinements

- Cutover
  - The new system is quickly put into use following through planning
RAD Features

- Use of Skilled With Advanced Tools (SWAT) Teams
- Technology
- User Involvement
SDLC Vs. RAD Life Cycle

a. Traditional life cycle

- Information systems department
- User community
- Analysis
- Design
- Coding
- Testing
- Cutover

b. RAD life cycle

- User community
- Information systems department
- Requirement planning
- User design
- Construction
- Cutover

Phased Development

- Is a derivation of the staged approach of the traditional SDLC, and also recognizes the need to incorporate the repetitive, looping nature of prototyping, and assumes the use of modern day tools that form the basis for RAD.
Some Popular Methodologies

Phased development

Fig 3.6: The Phased Development Model

P: Preliminary investigation

A: Analysis

User review

C: Preliminary construction

D: Design

Final construction

System test and installation

Cutover: Put system in production
Phased Development Stages
Preliminary Investigation

- Perform enterprise analysis
- Specify system boundaries, determine system objectives and goals, and evaluate project risk
- Evaluate feasibility study and obtain approval to proceed
- Conduct JAD sessions to confirm preliminary findings

Analysis
- Analyze functional requirements
- Document the functional requirements

Design
- Design new system components
- Design interfaces with other systems
- Document the new system design
Phased Development

Preliminary Investigation Stage

- Perform enterprise analysis
- Specify system boundaries and objectives, and evaluate project risk
- Evaluate feasibility and obtain approval to proceed
- Conduct JAD sessions to confirm preliminary findings
Phased Development Stages
Analysis

- Analyze functional requirements
- Document the functional requirements
  - Project dictionary
    - A collection of system documentation that the project team members create throughout SDLC
  - Repository
    - Describes system documentation that resides in an electronic form
Phased Development Stages

Design

- Design New System Components
- Design interfaces with other systems
- Document the new system design
- New system components are defined in reverse sequence
Phased Development Stages
Preliminary Construction

- Construct new system software modules and test data
- Demonstrate to users and project sponsors
Phased Development Stages
Final Construction

- Plan and prepare physical facilities
- Plan, obtain and install hardware
- Obtain software tools, and purchased software
- Build the test files and production database
- Plan and prepare training materials and documentation
Phased Development Stages
Test and Installation

- Design and perform system test
- Install components
- Conduct a user review/acceptance test
Phased Development

Preliminary construction stage

- Construct new system software modules and test data

Final construction stage

Installation activities

**Fig 3.11: Installation Activities Begin in the Analysis Phase**

<table>
<thead>
<tr>
<th>Preliminary Investigation Stage</th>
<th>Functional Delivery I</th>
<th>Functional Delivery 2</th>
<th>Functional Delivery n</th>
<th>Final Construction Phase</th>
<th>System Test and Installation Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A D PC</td>
<td>A D PC</td>
<td>A D PC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Plan and prepare the physical facilities
- Plan, obtain, and install the development environment hardware
- Obtain software tools, and code or obtain application software
- Build the test files and production database
- Plan and prepare training materials and documentation
Selecting the Right Methodology

System influence

Table 3.2: Evaluation of Alternate Development Methodologies Based on System Characteristics

<table>
<thead>
<tr>
<th>System Characteristics</th>
<th>Traditional SDLC</th>
<th>Prototyping</th>
<th>RAD</th>
<th>Phased Development</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scope</strong></td>
<td>Small, medium, or large scope</td>
<td>Small or medium scope</td>
<td>Small, medium, or large scope</td>
<td>Small, medium, or large scope</td>
</tr>
<tr>
<td><strong>Risk</strong></td>
<td>Low risk</td>
<td>Medium risk</td>
<td>Medium to high risk</td>
<td>Medium to high risk</td>
</tr>
<tr>
<td><strong>Complexity</strong></td>
<td>Little or no complexity</td>
<td>Medium to great complexity</td>
<td>Medium to great complexity</td>
<td>Medium to great complexity</td>
</tr>
</tbody>
</table>
Ensure consistency

- System requirements specifications
- System design specifications
- The system test plan