Large Scale Systems Design
G52LSS

Lecture 2 – Large Systems Development

- Phases of the SDLC
- Methodologies for SDLC
- Selecting Methodologies

Learning outcomes:
- describe phases, steps and deliverables of the SDLC
- understand the different structured and non-structured SDLC methodologies
- appreciate advantages and disadvantages of the different SDLC methodologies.

Phases of the SDLC

The Systems Development Life Cycle is a gradual refinement process traditionally consisting of 4 phases:

- Planning
- Analysis
- Design
- Implementation

Each phase is composed of steps that rely on techniques
Each step is meant to produce a specific deliverable

Planning Phase

Opportunity Identification

System Request

Feasibility Analysis
(technical, economic, organisational)

Project Management

Project Workplan

Analysis Phase

Develop Analysis Strategy

Define Requirements

Use Cases, Process Model, Data Model

Systems Proposal
**Design Phase**

- Design Selection
- Architecture Design
- Interface Design
- Data Storage Design
- Program Design

**System Specification**

**Implementation**

- System Construction
- Installation Process
- Support Plan

**Functional, Efficient and Robust System**

**Methodologies for SDLC**

- **Structured Design**
  - Waterfall development
  - Parallel Development

- **Rapid Application Development**
  - Phased Development
  - Prototyping
  - Throwaway Prototyping

- **Agile Development**
  - Extreme programming
  - Other agile methodologies include: Crystal, Scrum, Adaptive Software Development, Dynamic Systems Development, Feature Driven Development.

**Waterfall Development**

Diagram from [Dennis et al. 2000](#)
Parallel Development
Division into subprojects

High level design

More specific designs

Phased Development
Versions-based process

Prototyping
Performs phases concurrently and repeatedly

Prototype (not a 1st version)

User tries a finished system

Throwaway prototyping
Uses disposable design prototypes

High level design

Not fully working system
**Extreme Programming**

System grows iteratively

Diagram from (Dennis et al. 2006)

---

**Exercise 2.1** Suppose you are a project manager using the waterfall development methodology on a large and complex project. Your manager has just read a recent article in a magazine that advocates replacing the waterfall methodology with prototyping and comes to your office requesting you to switch approaches. What would you say? Exercise taken from (Dennis et al. 2006)

---

**Selecting Methodologies**

- Difficult because no methodology is always the best
- Standards and practices vary between organisations
- Important criteria are the following

<table>
<thead>
<tr>
<th>Ability to develop systema that...</th>
<th>WF</th>
<th>PA</th>
<th>PH</th>
<th>PR</th>
<th>TP</th>
<th>XP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unclear requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unfamiliar technology</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short schedule</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule visibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The above methodologies can also be combined to produce hybrid SDLC methodologies that would be more adequate for certain project development scenarios.

**Exercise 2.2** Suppose you were to combine throwaway prototyping with parallel development. What diagram would illustrate the hybrid methodology? What would be the pros and cons of the hybrid methodology? Exercise taken from (Dennis et al. 2006)
Additional Reading

Chapter 1 of (Dennis, Wixom and Roth, 2006)
Chapter 1 of (Kendall and Kendall, 2005)