Large Scale Systems Design
G52LSS

Lecture 19 – Implementation and Maintenance

- System Construction
- Documentation
- System Installation
- Support Plan

Learning outcomes: describe the most important aspects of the implementation phase in the SDLC: understand the different types of testing; appreciate the importance of good documentation and support plan for successful system implementation.

"If I had 6 hours to chop down a tree, I'd take 4 hours to sharpen the saw." – Abraham Lincoln

System Construction involves programming, testing, documenting and implementing new procedures.

Design Adequate Tests
- Test planning – set objectives, test cases, expected results, etc.
- Stub tests – correctness of control structures
- Unit tests – correctness of each module
- Integration tests – interaction between modules
- System tests – as part of the overall system
- Acceptance tests – satisfaction of organisational needs


d System Testing
- Requirements Testing – ensures integration does not provoke new errors
- Usability Testing – tests how easy and error-free the system is when in use
- Security Testing – assures that security functions are handled properly
- Performance Testing – assures system works under very high activity
- Documentation Testing – analysts check the accuracy of documentation

Acceptance Testing
- Alpha Testing – performed by users to assure they accept the system; frequently repeats earlier tests
- Beta Testing – uses real data, not test data: actual users monitor for errors or needed improvements

System Construction

Implementation

System Construction
Installation Process
Support Plan

Functional, Efficient and Robust System

Poor analysis, design and project management are more likely to be the causes of project failure instead of poor programming.

Unit Testing
- Black box testing – focuses on whether the unit meets requirements stated in specification
- White box testing – looks inside the module at actual code

Integration Testing
- User interface testing – tests each interface function
- Use-scenario testing – ensures that each use scenario works correctly
- Data flow testing – tests each process in a step-by-step fashion
- System interface testing – ensures data transfer between systems

Documentation

Developing good documentation takes time as it involves document design, text writing and editing, diagrams preparation, and document testing.

System Documentation – intended to help programmers and analysts understand and maintain the system after it is installed.

User Documentation – intended to help users operate the system.

Online documentation is becoming more important as it offers several advantages over paper documentation.
**Document structure design:**
- Step 1. Identify documentation topics – commands and menus, how to perform tasks, definition of important terms.

**Types of user documentation:**
- Procedure manuals – describe how to perform business tasks (print enrolment reports, accepting student enrolment).
- Tutorials – teach how to use components the system.
- Reference manuals – designed to learn a specific function of the system.

---

**System Installation**

Replacing the current system with a new one is a difficult task that demands careful planning and attention.

**Conversion** – the new system replaces the old system.

**Change management** – help people to adopt the new system and adapt to working with it.

**Conversion Style**
- Direct conversion – new system instantly replaces the old one.
- Parallel conversion – both systems are in operation simultaneously.

---

**Conversion Location**
- Pilot conversion – locations are selected for pilot conversion
- Phased conversion – sets of locations are converted one after another
- Simultaneous conversion – all locations are converted at the same time

**Conversion Modules**
- Whole system conversion – the entire new system is installed
- Modular conversion – install one module at a time of the new system

---

**Support Plan**

Post-installation activities include:
- **System support** – help users to use the system: on-demand training, online support, help desk
- **System maintenance** – refine the system to evolve as needed: requests for changes from problem reports, required enhancements, other projects, software/network changes, management decisions, etc.
- **Auditing** – expert examination of the system by someone not involved in the project to assess its reliability/ internal auditors, external auditors
- **Project assessment** – understand what was successful and what need to be improved about system and project activities: project team review and system review

---

**Types of Maintenance Approaches**
- **Corrective** – repair design and programming errors, accounts for most of the maintenance costs, adds little value, urgent
- **Adaptive** – modify system to environmental changes, adds more value
- **Perfective** – evolve system to solve new problems or take advantage of new opportunities. Sometimes considered as ‘new development’
- **Preventive** – safeguard system from future problems

The cost of maintenance is considerable (up to 80% of total budget for the whole project in some cases!) and is affected by a number of factors including: latent defects, number of users, quality of documentation, maintenance personnel, tools, well-structured programs.

---

**Additional Reading**

Chapter 13 and 14 of (Dennis et al., 2008)
Chapter 16 of (Kendall and Kendall, 2005)