

Kenyatta University
SCO 215: SYSTEM ANALYSIS AND DESIGN

SAD Lecture 8: Implementation

- The implementation phase consists of developing and testing the system's software, documentation, and new operating procedures.
- Managing the programming process is the major task of the systems analyst in this phase.
- While programmers work on programming, the systems analyst design a variety of tests to ensure that the system does what it was designed to do.
- During this phase, the systems analysts finalize the system documentation and develop the user documentation.

Implementation Activities

The major tasks performed during system implementation include:

- (1) Hardware acquisition and installation
- (2) Coding/programming/software acquisition
- (3) Testing
- (4) Training
- (5) Documentation
- (6) Installation/systems conversions or change over
- (7) Systems support

8.1. Hardware Acquisition

Entails analysis and evaluating sources of a hardware placing an order and acquiring the hardware

Installation involves site selection and preparation. Installation involves putting in place the equipment and software so that testing and conversion activities can occur. Site preparation activities include electrical and air-conditioning preparation, site layout and installation of the furniture and cables to connect and accommodate the equipment.

8.2. Coding

The process of translating the physical design specification into a working computer code or programs.

N/B: The deliverable of coding is an executable code (a working computer program) and accompanying documentation.

- The project manager's tasks during the process of programming:
 - assigning programming tasks,
 - coordinating the activities, and
 - managing the programming schedule.

8.2.1. Assigning Programming Tasks

- The project manager first groups together modules that are related, and then assigns the groups of modules to programmers on the basis of their experience and skill level.

- The project manager must deal with a mismatch between the available programming skills and the programming skills that are needed for the project.
- The best size of programming team is the smallest feasible one that can function as independent as possible.

8.2.2. Coordinating Activities

- Coordination can be done through both high-tech and low-tech means.
- The simplest approach is to have a weekly project meeting.
- Another important method is to create and follow standards.
- Many project teams set up three “areas” for programmers:
 - development area,
 - testing area, and
 - production area.
- Implement *change control* techniques:
 - keeping files and programs in different places according to completion status,
 - using a *program log* to keep track of program changes.
- Many CASE tools are set up to track the status of programs and help manage programmers.

8.2.3. Managing the Schedule

- The initial time estimates must be refined as the project progresses during construction.
- One of the most common causes for schedule problems is *scope creep*.
- Another common cause is unnoticed day-by-day slippages in the schedule.
- Typically, a project manager will create risk assessment that tracks potential risks that have an impact on the schedule and costs.

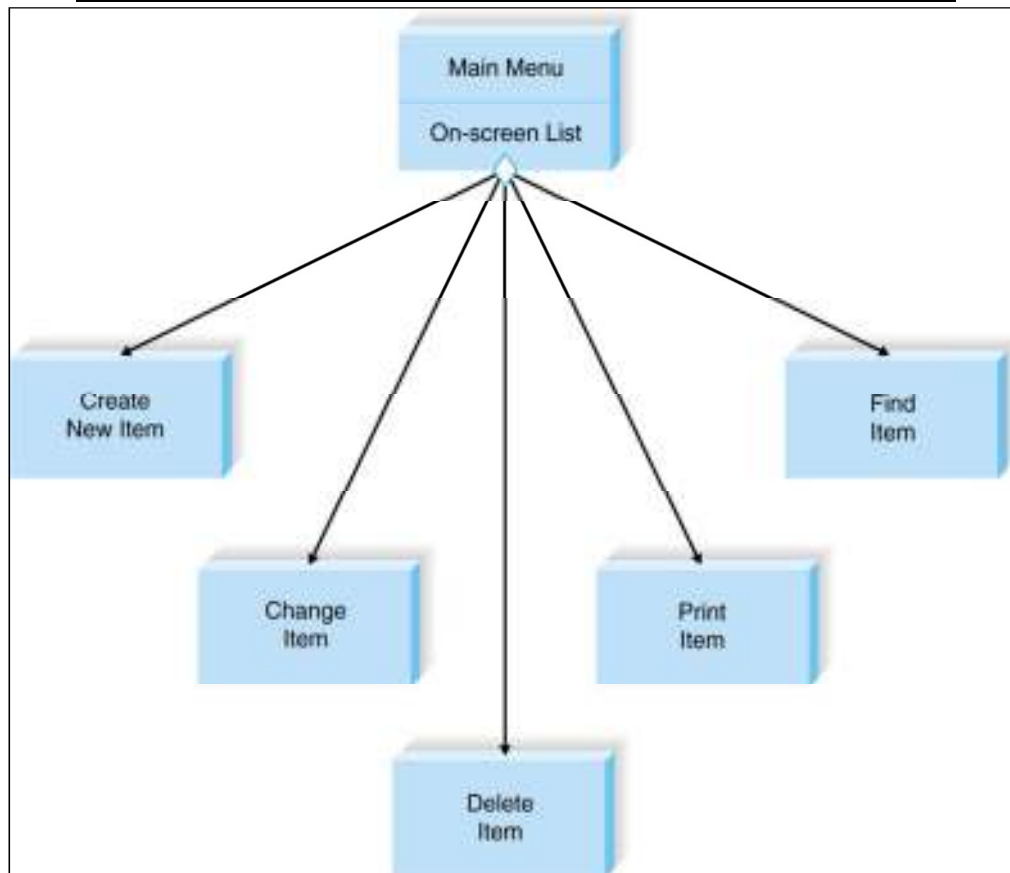
8.3. Testing

- Testing is a process of inspecting the developed system to find out whether it is executing its function correctly and also to find out whether it satisfies the user requirements.
- The attention paid to testing is justified by the high **costs** associated with **downtime** and failures caused by software bugs.
- Testing is done to ensure the system
 - is executing the specified functions
 - is error free
 - satisfies the requirements

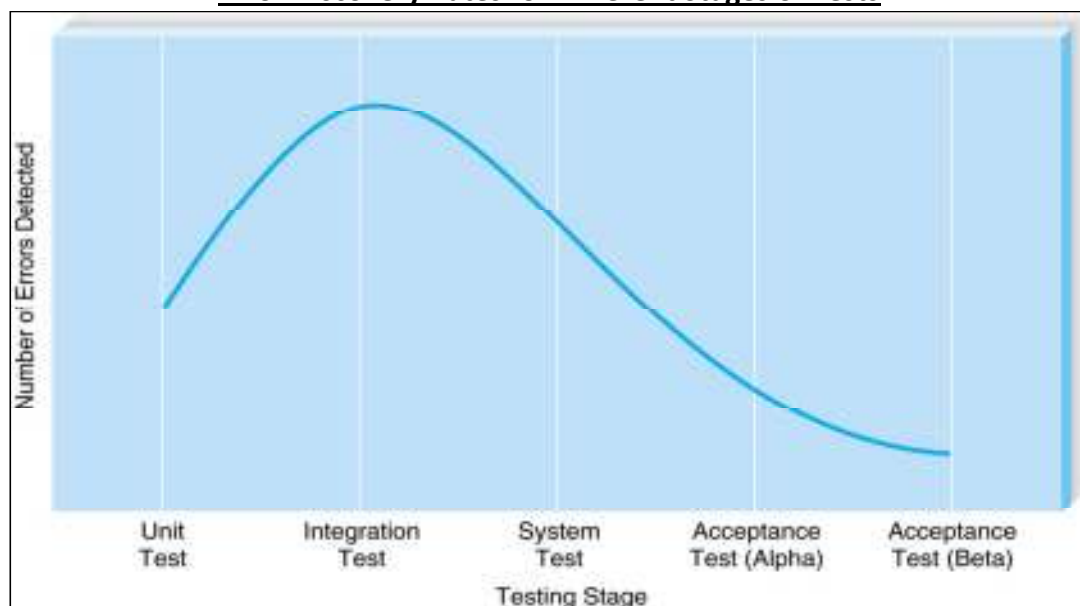
8.3.1. Test Planning

- Testing starts with the tester’s developing a *test plan* that defines a series of tests that will be conducted.
- A test plan describes a set of very specific *test cases* to examine, and defines the expected results.
- The tester develops a series of test cases to ensure that the quality of programs is validated.
- There are four general stages of tests: unit tests, integration tests, system tests, and acceptance tests.
- Other type of tests are: inspection, desk check/dry run, structured walkthrough, alpha test, beta test, black box testing and white box testing.

Each of the functions is a separate module that needs to be tested



Error Discovery Rates for Different Stages of Tests



8.3.2. Types of Tests

(i) Unit Tests

- *Unit tests* focus on one unit – a program or a program module that performs a specific

function that can be tested.

- There are two approaches to unit testing:
 - *Black-box testing*
 - The test plan is developed directly from the program specification.
 - *White-box testing*
 - The tester reviews the actual program code.

(ii) Integration Tests

- *Integration tests* assess whether a set of modules or programs that must work together do so without error.
- There are four approaches to integration testing:
 - user interface testing,
 - use scenario testing,
 - data flow testing, and
 - system interface testing.

(iii) System Tests

- *System tests* are usually conducted by the systems analysts to ensure that all modules and programs work together without error.
- System tests examine
 - how well the system meets business requirements,
 - usability,
 - security,
 - performance under heavy load, and
 - system's documentation.

(iv) User Acceptance Tests

- *Acceptance tests* are done primarily by the users.
- The goal of acceptance tests is to confirm that the system is complete, meets the business needs, and is acceptable to the users.
- Acceptance testing is done in two stages:
 - *Alpha testing* - conducted at developers site by a customer. The customer uses the software with the developer and usage problems are noted/recorded for resolution. It is conducted in a controlled environment, using test data.
 - *Beta testing* – users begin to use the system with real data and carefully monitor the system for errors. Beta testing is conducted at one or more customers' site. It is live testing in an environment not controlled by the developer. The customer records and reports difficulties and errors.

Forms of Beta Testing

1. **Recovery testing:** Forces the system to fail in a variety of ways in order to verify that recovery is properly performed.
2. **Stress testing:** Designed to confront a system with abnormal quantities of computer resources or transactions to see how it can cope with such abnormalities.
3. **Security testing:** This verifies that the protection/security mechanism built into the system will protect it from inappropriate access or unauthorized penetration

4. **Performance testing:** Conducted to evaluate how the system performs in the range of possible environment in which it may be used. e.g. runtime, response, quality of output etc.

8.4. TRAINING

Involves educating and training the systems users on how to use the new or the modified system, its benefits, necessary precautions..

8.4.1. Objectives of Training

1. Making the system more acceptable to users
2. Reducing the errors arising from operating the system through trial and error method
3. Reducing the cost of systems maintenance
4. Improving the security of the system by eliminating accidental destruction of data.

8.4.2. Potential Training Topics

- (a) Use of the system
- (b) General computer concepts
- (c) Organizational concepts
- (d) Systems management
- (e) System installation

8.4.3. Training Methodologies

The following methods can be used for training

- (1) Resident experts to give on job training
- (2) Use of Computer Aided/Assisted instruction (CAI)
- (3) Formal courses
- (4) Software help components
- (5) Tutorial seminars and workshops
- (6) Interactive training manuals
- (7) External sources such as Hardware and software vendors
- (8) Electronic Performance Support System – A component of a software package/ application in which training and educational information is embedded.

8.4.4. Deliverables of training

The end product of the training phase is:

- (a) Training manuals
- (b) Training plans

8.5. DOCUMENTATION

- The process of producing descriptive and instructive materials that indicate the use and the nature of Information System.
- There are two fundamentally different types of documentation:
 - *system documentation* is intended to help programmers and systems analysts

- understand the system and enable them to build it or maintain it;
 - *user documentation* is designed to help the user operate the system.
- User documentation should not be left until the end of the project.
- Time required to develop and test user documentation should be built into project plan.
- On-line documentation is becoming the predominant form.
- There are four key strengths of online documentation:
 - Searching for information is simpler.
 - The same information can be presented in many different formats.
 - It enables the user to interact with the documentation in many new ways.
 - It is significantly less expensive than paper documentation.

8.5.1. Uses of systems documentation

1. **Communication:** Documentation serves as a means of communication between the systems users and the systems developers.
2. **Backups:** Documentation safeguards against lose of the understanding of the system especially in cases where the originator of the system has left the company.
3. **Reference:** Documentation provides some reference material for purposes of reconstruction in an event of systems failure.
4. **Training:** It provides a basis for training new system users e.g. user manuals
5. **Check lists:** Documentation can effectively provide a checklist of items to cover both in design and subsequent audit.
6. **Maintenance:** Documentation provides a basis for enhancing or modifying the existing system.

8.5.2. Categories of documentation

(1) Analytic documentation

Consists of all the records and reports produced when a system is initiated e.g. feasibility study report and statement of the problem report

(2) System documentation

Is detailed information about a system, its internal working and functionality?

(3) Operations documentation

This specifies procedures required for running the system by the technical personnel (Computer operators, Programmers,).

Also known as technical documentation.

(4) User documentation

Written or other visual information about an information system, how it works and how it is to be used.

- There are three different types of user documentation:
 - *Reference Documents* are designed to be used when the user needs to learn how to perform a specific function.

- *Procedural Manuals* describe how to perform business tasks.
- *Tutorials* teach people how to use major components of the system.

8.5.3. Designing Documentation Structure

- The general structure used in online documentation is to develop a set of *documentation navigation controls* that lead the user to *documentation topics*.
- There are five general types of navigation controls for topics: table of contents, index, text search, intelligent agent, and Web-like links.

8.5.4. Writing Documentation Topics

- Topics start with clear titles, followed by introductory text that define the topic, and then provide detailed, step-by-step instructions.
- Many topics include screen images and “show me” examples.
- Most also include navigation controls to enable movement among topics.
- Some also have links to related topics.

References

1. Dennis, A., Wixom, B.H. & Roth, R.M. (2015). *Systems Analysis and Design, 6th Edition*. Wiley. Chapter 12 (Implementation).