

Concept of Information Systems and Software : Information gathering, requirement and feasibility analysis, data flow diagrams, process specifications, input/output design, process life cycle, planning and managing the project.

Information Gathering

Information Gathering Strategies

The strategy consists of identifying information sources, evolving a method of obtaining information from the identified sources and using an information flow model of organization.

Information Sources

The main sources of information are users of the system, forms and documents used in the organization, procedure manuals, rule books etc, reports used by the organization and existing computer programs(If Any).

Feasibility Analysis

Feasibility analysis is used to assess the strengths and weaknesses of a proposed project and give directions of activities that shall improve a project and achieve desired results.

Requirements Determination Sub-activities :

1. **Requirements Anticipation.** The systems analyst hypothesizes that particular requirements are relevant based on his or her previous experience with other similar systems and knowledge about the problem domain.
2. **Requirements Elicitation.** The systems analyst uses this activity to gather the essential requirements from the user employing a variety of techniques, such as interviews, questionnaires, group brainstorming meetings, and voice and e mail.
3. **Requirements Assurance.** The systems analyst uses the activity of requirements assurance to validate and verify the requirements with the user as being the real and essential requirements. A user walk through in which the systems analyst and the user together review documented requirements in detail is one assurance technique.
4. **Requirements Specification.** This is the activity that the systems analyst uses to explicitly catalog and document the requirements either during or after the elicitation and assurance activities.

Methods Used To Gather An Information System's Requirements

Interviews involve at least one systems analyst and at least one user conversing about the information system's requirements. Interviewing for requirements is similar to your interviewing for a job position or a television talk show host interviewing a guest. **Observation** is the act of the systems analyst going to a specific location to observe the activities of the people and machinery involved in the problem domain of interest.

Questionnaires are feedback forms designed to gather information from large groups of people.

Creating a **prototype** of the information system can be done on an individual level or in a group setting. The idea is to explore system alternatives by developing small working models of the proposed system so that user reactions can be gathered.

System Requirements Specification (SRS)

SRS is obtained after excessive discussions with the user. System requirements specification specifies what Information requirements will be provided. It does not specify how the system will be designed.

Developing A Document Flow Diagram

Example Word Statement

"Our company receives many items from several vendors each accompanied by a delivery note. A receiving office receives the item and checks the delivery note with corresponding order. Any discrepancy is reported to purchase office. The items received along with items received note (with details of items) is sent to the inspection office."

Entities Identified-Vendors, Receiving office, Inspection office

Documents identified - Delivery note, discrepancy note, Items Received note.

Using these a document flow diagram is drawn.

The diagram is interpreted as follows:

1. Vendors deliver items to receiving office accompanied by a delivery note.
2. Receiving Office sends items to inspection office.
3. Receiving office sends discrepancy note to Purchase office.

ENTITIES: Vendor, Receiving office, Inspection office and purchase office

DOCUMENTS: Delivery note, items received and discrepancy note.

Data Flow Diagram (DFD)

DFD has entities and data flows, DFD specifies processing performed by some of the entities. It specifies which entities generate documents and also indicate their flow. Data stores which are referred while processing data and in which processed data may be written or stored are also represented in the Diagram.

- Entities are, originators of data and "consumers" of data.
- Vendor, Inspection office and purchase office are entities in the diagram.
- Data flows are delivery note, items received note and discrepancy note.
- A circle is used to depict a process.
- A pair of parallel lines depict a store.

Input/Output Design

Design Phases

- Architectural design: Identify sub-systems.
- Abstract specification: Specify sub-systems.
- Interface design: Describe sub-system interfaces
- Component design: Decompose sub-systems into components.
- Data structure design: Design data structures to hold problem data.
- Algorithm design: Design algorithms for problem functions

Coupling

It is the measure of the strength of the interconnections between system components. Loose coupling means component changes are unlikely to affect other components. Shared variables or control information exchange lead to Tight coupling. Loose coupling can be achieved by state decentralization (as in objects) and component communication via parameters or message passing. Coupling and Inheritance

Types of Coupling

- Content coupling (high)

Content coupling is when one module modifies or relies on the internal workings of another module (e.g., accessing local data of another module). Therefore changing the way the second module produces data (location, type, timing) will lead to changing the dependent module

- Common coupling

Common coupling is when two modules share the same global data (e.g., a global variable). Changing the shared resource implies changing all the modules using it.

- External coupling

External coupling occurs when two modules share an externally imposed data format, communication protocol, or device interface.

- Control coupling

Control coupling is one module controlling the flow of another, by passing it information on what to do (e.g., passing a what-to-do flag). Stamp coupling (Data-structured coupling) Stamp coupling is when modules share a composite data structure and use only a part of it possibly a different part (e.g., passing a whole record to a function that only needs one field of it). This may lead to changing the way a module reads a record because a field, which the module doesn't need, has been modified.

- Data coupling

Data coupling is when modules share data through, for example, parameters. Each datum is an elementary piece, and these are the only data shared (e.g., passing an integer to a function that computes a square root).

- Message coupling (low)

This is the loosest type of coupling. It can be achieved by state decentralization (as in objects) and component communication is done via parameters or message passing. (see Message passing).

- No coupling

Modules do not communicate at all with one another.

Software Life Cycle Models

The software life cycle is a general model of the software development process, including all the activities and work products required to develop a software system. A software life cycle model is a particular abstraction representing a software life cycle. Such a model may be:

- **activity-centered** focusing on the activities of software development
- **entity-centered**-focusing on the work products created by these activities

Waterfall Model

This prescribes a sequential execution of a set of development and management processes, with no return to an earlier activity once it is completed. Some variants of the waterfall model allow revisiting the immediately preceding activity ("feedback loops") if inconsistencies or new problems are encountered during the current activity.

V-Model

Another variant of the waterfall model-the V-model-associates each development activity with a test or validation at the same level of abstraction. Each development activity builds a more detailed model of the system than the one before it, and each validation tests a higher abstraction than its predecessor

Spiral Model

Spiral model addresses the weakness of the waterfall model, spiral model focuses on addressing risks incrementally by repeating the waterfall model in a series of cycles or rounds :

- Concept of Operation
- Software Requirements
- Software Product Design
- Detailed Design
- Code
- Unit Test
- Integration Test
- Acceptance Test
- Implementation

Planning And Managing The Project

The SRO, or identifier of the potential project should:

1. Define and justify the need for the project
2. Specify, quantify and agree desired outcome and benefits
3. Appoint a Project Manager and if appropriate set up a Project Board

The project management team should:

1. Plan how to deliver the required outcomes and benefits
2. Decide how to manage relationships with key stakeholders
3. Decide how to project manage the delivery process
4. Determine resource requirements and ensure they can be made available when required SRO/Project
5. Develop Business Case to enable the SRO/Project Board to decide whether project is cost and risk justified.

The SRO should ensure:

- Post project reviews are carried out to measure the degree to which benefits have been achieved
- The Business Case is updated to reflect operational reality
- Potential improvements/changes/opportunities identified in the reviews fed into the strategic planning process for consideration

Planning checklist

1. Confirm scope and purpose of the plan:
2. Define the deliverables:
3. Identify and estimate activities
4. Schedule the work and resources
5. Identify risks and design controls
6. Document and gain approval for the plan