

## **System Study/Requirement Determination**

### **Data & Fact Gathering Techniques/Fact Finding Techniques:**

The specific methods that an analyst uses for collecting data about the requirement of a new system are called fact finding techniques.

#### **1. Interview:**

In this technique analyst collects information from individuals. It is a formal meeting where the analyst can obtain information about the operation of the present system and requirements of planned system.

##### **Advantages**

- i. It is helpful for gathering information from individuals who do not communicate effectively by writing.
- ii. It allows discovering areas for unrealistic expectation, misunderstanding to the proposed system.
- iii. Analyst can observe the interviewee's non verbal communication.
- iv. This method gives the analyst the opportunity to motivate the interview to response freely & openly to questions.

##### **Disadvantages**

- i. It is very time consuming.
- ii. Success of interview is dependent on system analyst's human relation skills.
- iii. Interviewing may be impractical due to location of the interviewees.

##### **Types of interview**

###### **i. Structured Interview:**

In structured interview interviewer has specific set of questions to ask. All questions prepared in advance and answers are already available.

###### **ii. Unstructured Interview:**

In this method questions and corresponding responses are open ended. This are conducted with a general goal or subject.

##### **Planning for interview**

1. Purpose of interview should be clear. The purpose of meeting clearly explained to the all participant so that relevant document can be supplied in advance.
2. Proper time, duration and place for interview must be selected before so this will allow the participants to scheduled work accordingly.

#### **2. Group Communication/Discussion:**

When information is required from face to face communication but there is not enough time to conduct personal interview, group interviews can be held.

### **Advantages**

- i. There are many persons presents so more types of ideas and views can be discussed in short time.
- ii. The comments of one person may prompt other person to contribute facts which they thoughts.

### **Disadvantages**

- i. The group may be dominated by a few persons.
- ii. The situation could lead to a verbal fight between the persons and may need moderation.
- iii. Internal politics of an organisation may determine what is said and what is left thus resulting in false picture.

## **3. Questionnaires:**

Questionnaires are special purpose documents that allow the analyst to collect information and opinions from respondence. This is more structured and formal method of collecting data.

### **Advantages**

- i. When it is used for group of users, it is relatively cheap.
- ii. A questionnaire can be administered to larger no of individual simultaneously.
- iii. The respondence feels greater confidence.

### **Disadvantages**

- i. All the questions given in questionnaire are usually not answered completely.
- ii. It is not possible to observe and analyse the respondence body language.
- iii. Good questionnaires are difficult to prepare.
- iv. Cost may be high.

Types of questionnaires –

- I. Structured – Answer and question are fixed
- II. Unstructured – Not fix pattern

## **4. On site observation:**

It allows the analyst to gain information which cannot be obtained by other fact finding methods. It is also useful when analyst need to actually observe how documents are handled, what processes are carried on etc.

“On site observation provide close view of working of the real system. The analyst can observe people, objects, documents and occurrence of events.”

### **Advantages**

- i. Data collection by observation highly reliable.
- ii. It is relatively inexpensive

### **Disadvantages**

- i. People usually feel uncomfortable when being watched.
- ii. It is very time consuming.
- iii. Some activities may take place odd time causing a scheduling inconvenience for system analyst.

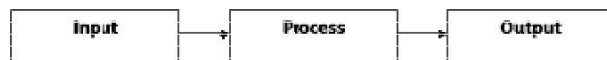
## 5. Record Viewer.

Many kinds of records and reports can provide analysts with valuable information about organization

and operations. In record reviews analysts examine information that has been recorded about the system and users.

Records includes written policy manuals, regulations, and standards operating procedures used by most organizations.

### Elements of software system:



A system's objectives are expressed in terms of output is needs to produce, the inputs are the data which are processed while the output i.e. information are the outcome of the process.

In addition there are two more components

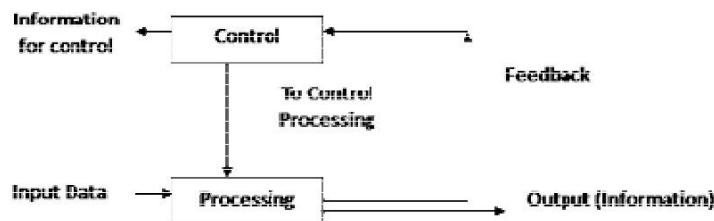
#### i. Control

It makes the system to operate within tolerable performance levels. Control of the system is the decision maker that controls the activities of accepting input, process & producing the output.

#### ii. Feedback

It is the information on how well a system is performing & it is essential for system modifications. System may use feedback for control the information generated by comparing results with acceptable level of performance (standards) and informing the control elements of the difference is termed as feedback.

In feedback control the output is fed back to input. It allows to be majored against some standards and making adjustment in the processing accordingly.



## SRS ( System Requirement Specification)

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety

of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated. It is usually signed off at the end of requirements engineering phase.

### What should the SRS address?

The basic issues that the SRS writer(s) shall address are the following:

- a) *Functionality*. What is the software supposed to do?
- b) *External interfaces*. How does the software interact with people, the system's hardware, other hardware, and other software?
- c) *Performance*. What is the speed, availability, response time, recovery time of various software functions, etc.?
- d) *Attributes*. What are the portability, correctness, maintainability, security, etc. considerations?
- e) *Design constraints imposed on an implementation*. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

### What are the characteristics of a great SRS?

An SRS should be

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for importance and/or stability
- Verifiable
- Modifiable
- Traceable

**Correct** - This is like motherhood and apple pie. Of course you want the specification to be correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

**Unambiguous** - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

**Complete** - A simple judge of this is that it should be all that is needed by the software designers to create the software.

**Consistent** - The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

**Ranked for Importance** - Very often a new system has requirements that are really marketing wish lists. Some may not be achievable. It is useful provide this information in the SRS.

**Verifiable** - Don't put in requirements like - "It should provide the user a fast response." Another of my favorites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

**Modifiable** - Having the same requirement in more than one place may not be wrong - but tends to make the document not maintainable.

**Traceable** - Often, this is not important in a non-politicized environment. However, in most organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?

Software analysis and design includes all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human readable and understandable documents, to which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.