Complexity **:** Per dollar, pound or euro spent, software products contain more complexity than other engineered artifacts.

Conformity **:** The 'traditional' engineer usually works with physical systems and materials like cement and steel. These physical systems have complexity, but are governed by consistent physical laws. Software developers have to conform to the requirement of human clients. It is not just that individuals can be inconsistent. Organizations, because of lapses in collective memory, in internal communication or in effective decision making, can exhibit remarkable,'organizational stupidity'.

Flexibility **:** That software is easy to change is seen as a strength. However, where the software system interfaces with a physical or organizational system, it is accommodate the other components rather than vice versa. Thus software systems are particularly subject to change.

Categorizing Sps:

In project management, there are many categories that need to plan as well while planning the project. You will see each category in detail that how we can define the categories of the project. You will see categories like scope and significance, type of the project, level of technology, size, and scale of operations, ownership, and control, implementations, and purpose of the project are generally used categories.
If you're planning a project and want to implement then you can consider these categories. Projects are often categorized on the basis of their scope, size, speed of implementation, location, type, and technology. The project can be classified on the grounds of the following.

Scope and Significance :
The projects are generally classified on the basis of coverage and magnitude of their operations. So on the basis of scope projects can be National or International.

1. **National Projects –**
   There are also projects which are undertaken either by the government itself or assigned

to private entrepreneurs in a country. In a country like India Public and Private sectors coexist to undertake major and minor projects. Government projects and private projects operate in vastly different environments, associated with different advantages and disadvantages. The only purpose of the National Project is the growth and development of the economy and maintenance of existing standards of living.

2. **International Projects –**
   The projects which are embarked on by "Foreign investors" either by establishing a solitary or a branch of their unit or by mere participation in the equity of any domestic company are called International Projects. These can be in the form of joint ventures, MNC's, and collaborations between two companies.

Type :

According to the type, projects can be industrial and non-industrial.

1. **Industrial –**
   These are those projects which are undertaken with a view to developing the economy.
2. **Non-Industrial –**
   These projects can be related to welfare and maintenance of a standard of living in an economy.

Level of Technology :

Technology plays a significant role in managing projects. Projects can be sub-divided into four categories on the basis of technology. These are as follows.

- **Conventional Technology Projects –**
  These are the projects which use acquainted and known technology in the continuous process. e.g. steel, cement, sugar, chemicals, and fertilizers, etc.
- **Non-Conventional Technology –**
  Such kinds of projects apply if not the latest at least contemporary mode technology e.g. projects using cranes i.e. a mechanical way of lifting.
- **High-Tech Project –**
  Huge investments are made in technology in these types of projects, e.g., space projects, nuclear power projects, etc.
- **Low Investment Projects –**
  These types of projects demand low investment in technology e.g., cosmetics and household utilities, etc.

Size and Scale of Operations :

On the basis of size and scale of operations, projects can be large scale, medium scale, and small scale.

- **Small Scale Projects –**
  These are the projects which can be completed within a time period of 1-2 years and with investment below Rs. 5 crores.
- **Medium Scale Projects –**
  These are the projects which can be completed within a time period of 2-5 years and with investment between Rs. 5 to Rs. 10 crores.
- **Large Scale Projects –**
  These are the projects which can be completed within a time period of 5-10 years and with investment over and above Rs. 100 crores.

Ownership and Control :

Projects can be divided into 3 categories according to their governance.

- **Public Sector Projects –**
  These are fully owned and controlled by the government e.g., generating power and extracting minerals, etc.

- **Private Sector Projects –**
  These are fully owned by individuals and companies e.g., newspapers and magazines, etc.
- **Joint Sector Projects –**
  These projects are run and controlled by both government and private individuals are under this category.

Speed of Implementation :
According to the speed of implementation, projects can be normal, crash, and disaster projects.

- **Normal Projects –**
  In this category, an adequate time is allowed for implementation. It requires minimal capital costs.
- **Crash Projects –**
  In this category, additional capital is incurred to save time.
- **Disaster Projects –**
  In this category, naturally capital cost will go up, but project time will get drastically reduced. Failure of quality is accepted.

Purpose :
There is always a purpose for everything. So, the projects are classified according to purpose as follows.

- **Rehabilitation Projects –**
  These projects are undertaken by financially sound investing groups to service sick units. It is very risk and success are very less in such projects.
- **Balancing Projects –**
  These are undertaken to cope with changes in the supply side of economies of factors of production, to eliminate the underutilization of the actual capacities, and enhance efficiency and effectiveness.
- **Maintenance Projects –**
  These projects involve overhauling the machinery, repairs, and patching up activities at regular intervals.
- **Modernization Projects –**
  Modernization of old plants is required to cope with the dynamic environment.

Others :
Some other types of projects are as follows.

1. **Capacity Expansion Projects –**
   This involves enlarging the existing capacity of the products.
2. **Employees Welfare Project –**
   The objective of such projects is to install infrastructural facilities for improving working conditions and labor relations as well as to develop the skills of the staff.

Project as a System:

It is also known as Project Management Information system (PMIS) which is the coherent organization to execute projects successfully.

A PMIS is typically one or more software applications and depending upon an organization's operational requirements a methodical process for collecting and using project information. These electronic systems help plan,execute and close project management goals.

Management Control:

The project control cycle

Management, in general, can be seen as the process of setting objectives for a system and then monitoring the system to see what its true performance is. In Figure 1.2 the 'real world' is shown as being rather formless. Especially in the case of large undertakings, there will be a lot going on about which management should be aware. As an example, take an IT project that is to replace locally held paper-based records with a centrally-organized database. It might be that staff in a large number of offices that are geographically dispersed need training and then need to use the new IT system to set up the back-log of manual records on the new database. It might be that the system cannot be properly operational until the last record has been transferred. It might also be the case that the new system will be successful only if new transactions can be processed within certain time cycles. The managers of the project ought to be asking questions about such things as how effective training has been, how many records have still to be transferred to the new database and transfer rates. This will involve the local managers in data collection. Bare details, such as 'location X has processed 2000 documents' will not be very useful to higher management: data processing will be needed to transform this raw data into useful information. This might be in such forms as 'percentage of records processed', 'average documents processed per day per person' and 'estimated completion date'.

In the example above, the project leader might examine the 'estimated completion date' for completing data transfer for each branch and compare this with the overall target date for completion of this phase of the project. In effect they are comparing actual performance with one aspect of the overall project objectives. They might find that one or two branches are not going to complete the transfer of details in time, and would then need to consider what to do (this is represented in Figure 1.2 by the box making decisions/plans). One possibility would be to move staff temporarily from one branch to another. If this is done, there is always the danger that while the completion date for the one branch is pulled back to before the overall target date, the date for the branch from which staff are being moved is pushed forward beyond that date. The project manager would need to calculate carefully what the impact would be in moving staff from particular branches. This is modelling the consequences of a potential solution.

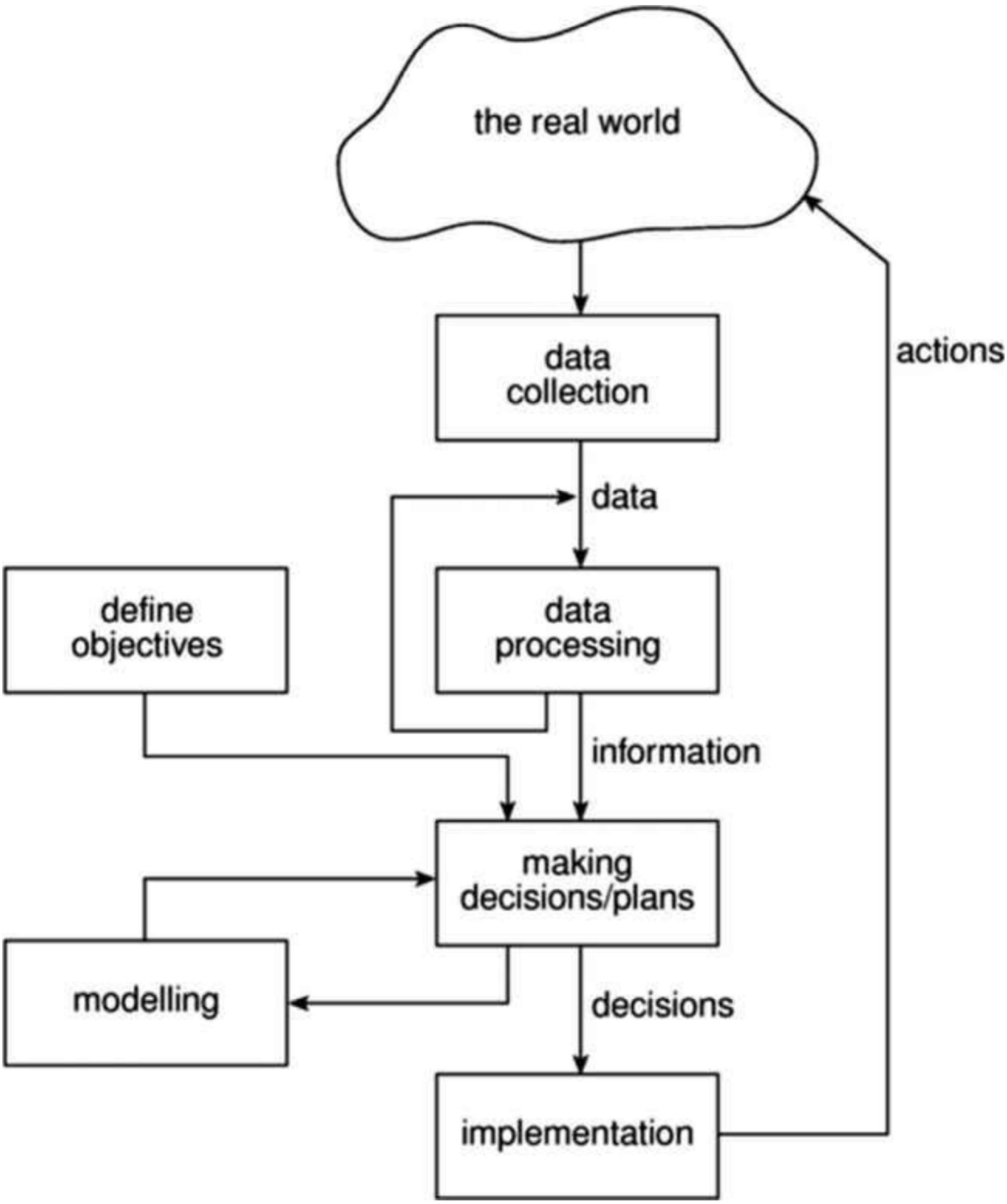


Figure 1.2 The project control cycle.

Several different proposals could be modelled in this way before one was chosen for implementation.

Having implemented the decision, the situation needs to be kept under review by collecting and processing further progress details. For instance, the next time that progress is reported, a branch to which staff have been transferred might still be behind in transferring details. This might be because the reason why the branch has got behind in transferring details is because the manual records are incomplete and another department, for whom the project has a low priority, has to be involved in providing the missing information. In this case, transferring extra staff to do data input will not have accelerated data transfer.

Objectives

Project objectives should To have a successful software project, the manager and the project team members be clearly defined. must know what will constitute success. This will make them concentrate on what is essential to project success.

There might be more than one set of users of a system and there might be different groups of staff who are involved its development. There is a need for well defined objectives that are accepted by all these people. Where there is more than one user group, then a project authority needs to be identified. Such a project authority has overall authority over what the project is to achieve.

This authority is often held by a project steering committee, which has overall responsibility for setting, monitoring and modifying objectives. The project manager still has responsibility for running the project on a day to day basis, but has to report to the steering committee at regular intervals. Only the steering committee can authorize changes to the project objectives and resources.

Measures of effectiveness

Effective objectives are concrete and well defined. Vague aspirations such as 'to improve customer relations' are unsatisfactory. Objectives should be such that it is obvious to all whether the project has been successful or not. Ideally there should be measures of effectiveness, which tell us how successful the project has been. For example, 'to reduce customer complaints by 50%' would be more satisfactory as an objective than 'to improve customer relations'.

Sub-objectives and goals

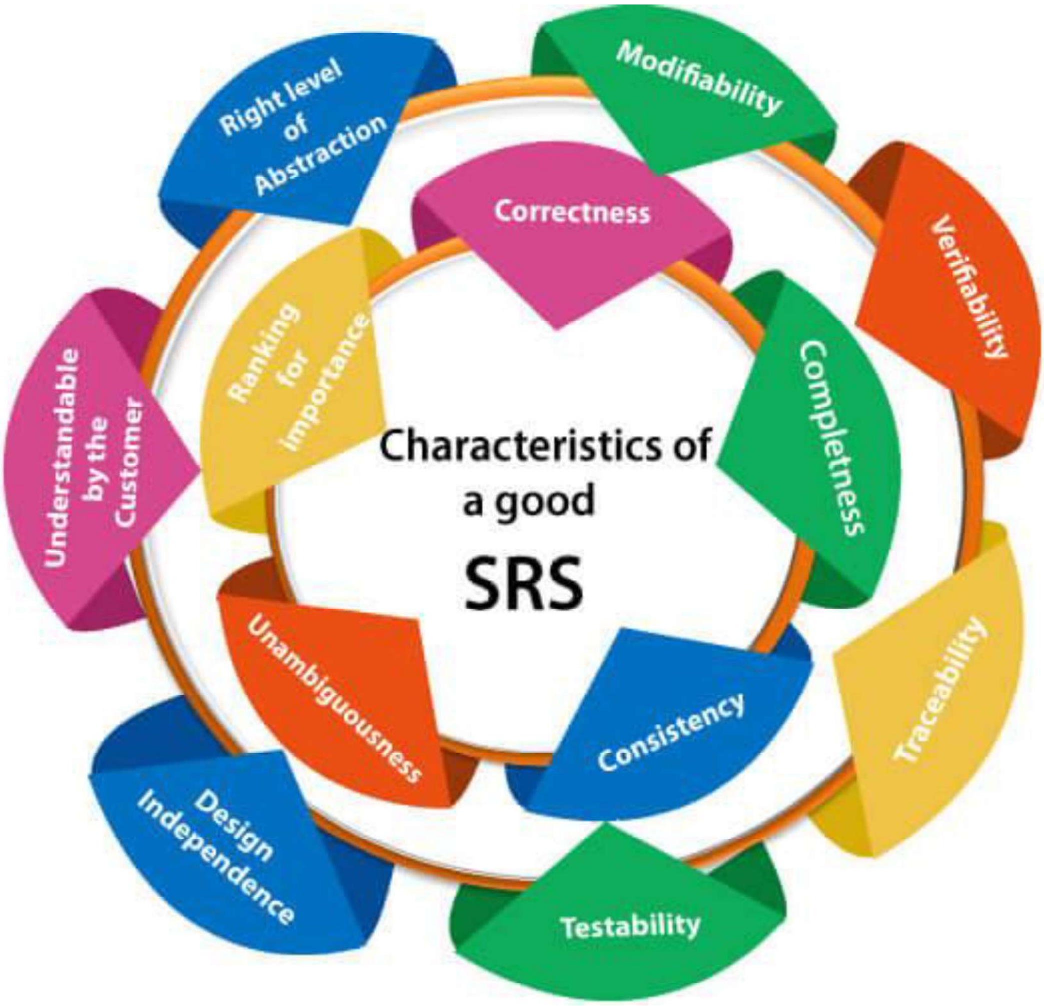
In order to keep things manageable, objectives might need to be broken down into sub-objectives. Here we say that in order to achieve A we must achieve B, C and D first. These sub-objectives are also known as goals, steps on the way to achieving an objective, just as goals scored in a football match are steps towards the objective of winning the match.

Requirement Specification:

The production of the requirements stage of the software development process is **Software Requirements Specifications (SRS)** (also called a **requirements document**). This report lays a foundation for software engineering activities and is constructing when entire requirements are elicited and analyzed. **SRS** is a formal report, which acts as a representation of software that enables the customers to review whether it (SRS) is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. It serves several goals depending on who is writing it. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system. The two methods create entirely various situations and establish different purposes for the document altogether. The first case, SRS, is used to define the needs and expectation of the users. The second case, SRS, is written for various purposes and serves as a contract document between customer and developer.

Characteristics of good SRS



Following are the features of a good SRS document:

1. Correctness: User review is used to provide the accuracy of requirements stated in the SRS. SRS is said to be perfect if it covers all the needs that are truly expected from the system.

2. Completeness: The SRS is complete if, and only if, it includes the following elements:

(1). All essential requirements, whether relating to functionality, performance, design, constraints, attributes, or external interfaces.

(2). Definition of their responses of the software to all realizable classes of input data in all available categories of situations.

Note: It is essential to specify the responses to both valid and invalid values.

(3). Full labels and references to all figures, tables, and diagrams in the SRS and definitions of all terms and units of measure.

3. Consistency: The SRS is consistent if, and only if, no subset of individual requirements described in its conflict. There are three types of possible conflict in the SRS:

(1). The specified characteristics of real-world objects may conflicts. For example,

(a) The format of an output report may be described in one requirement as tabular but in another as textual.

(b) One condition may state that all lights shall be green while another states that all lights shall be blue.

(2). There may be a reasonable or temporal conflict between the two specified actions. For example,

(a) One requirement may determine that the program will add two inputs, and another may determine that the program will multiply them.

(b) One condition may state that "A" must always follow "B," while other requires that "A and B" co-occurs.

(3). Two or more requirements may define the same real-world object but use different terms for that object. For example, a program's request for user input may be called a "prompt" in one requirement's and a "cue" in another. The use of standard terminology and descriptions promotes consistency.

4. Unambiguousness: SRS is unambiguous when every fixed requirement has only one interpretation. This suggests that each element is uniquely interpreted. In case there is a method used with multiple definitions, the requirements report should determine the implications in the SRS so that it is clear and simple to understand.

5. Ranking for importance and stability: The SRS is ranked for importance and stability if each requirement in it has an identifier to indicate either the significance or stability of that particular requirement.

Typically, all requirements are not equally important. Some prerequisites may be essential, especially for life-critical applications, while others may be desirable. Each element should be identified to make these differences clear and explicit. Another way to rank requirements is to distinguish classes of items as essential, conditional, and optional.

6. Modifiability: SRS should be made as modifiable as likely and should be capable of quickly obtain changes to the system to some extent. Modifications should be perfectly indexed and cross-referenced.

7. Verifiability: SRS is correct when the specified requirements can be verified with a cost-effective system to check whether the final software meets those requirements. The requirements are verified with the help of reviews.

8. Traceability: The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each condition in future development or enhancement documentation.

There are two types of Traceability:

1. Backward Traceability: This depends upon each requirement explicitly referencing its source in earlier documents.

2. Forward Traceability: This depends upon each element in the SRS having a unique name or reference number.

The forward traceability of the SRS is especially crucial when the software product enters the operation and maintenance phase. As code and design document is modified, it is necessary to be able to ascertain the complete set of requirements that may be concerned by those modifications.

9. Design Independence: There should be an option to select from multiple design alternatives for the final system. More specifically, the SRS should not contain any implementation details.

10. Testability: An SRS should be written in such a method that it is simple to generate test cases and test plans from the report.

11. Understandable by the customer: An end user may be an expert in his/her explicit domain but might not be trained in computer science. Hence, the purpose of formal notations and symbols should be avoided too as much extent as possible. The language should be kept simple and clear.

12. The right level of abstraction: If the SRS is written for the requirements stage, the details should be explained explicitly. Whereas,for a feasibility study, fewer analysis can be used. Hence, the level of abstraction modifies according to the objective of the SRS.

Properties of a good SRS document

The essential properties of a good SRS document are the following:

Concise: The SRS report should be concise and at the same time, unambiguous, consistent, and complete. Verbose and irrelevant descriptions decrease readability and also increase error possibilities.

Structured: It should be well-structured. A well-structured document is simple to understand and modify. In practice, the SRS document undergoes several revisions to cope up with the user requirements. Often, user requirements evolve over a period of time. Therefore, to make the modifications to the SRS document easy, it is vital to make the report well-structured.

Black-box view: It should only define what the system should do and refrain from stating how to do these. This means that the SRS document should define the external behavior of the system and not discuss the implementation issues. The SRS report should view the system to be developed as a black box and should define the externally visible behavior of the system. For this reason, the SRS report is also known as the black-box specification of a system.

Conceptual integrity: It should show conceptual integrity so that the reader can merely understand it. Response to undesired events: It should characterize acceptable responses to unwanted events. These are called system response to exceptional conditions.

Verifiable: All requirements of the system, as documented in the SRS document, should be correct. This means that it should be possible to decide whether or not requirements have been met in an implementation.

Information and Control in Organization:

With small projects, the project leaders are likely to be working very closely w ith the other team members and might even be carrying out many non-managerial tasks themselves. Therefore they should have a pretty good idea of what is going on. When projects are larger, many separate teams will be working on different aspects of the project and the overall managers of the project are not going to have day-to-day direct contact w ith all aspects of the work.

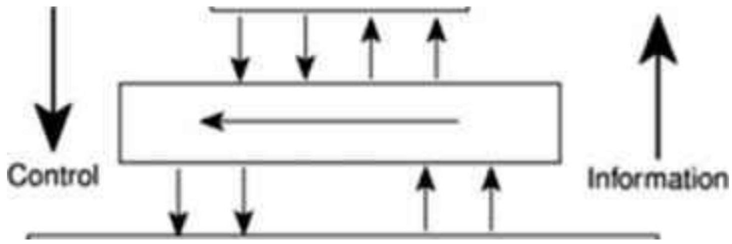
larger projects are likely to have a hierarchical management structure (Figure 1.3). Project team members will each have a group leader who allocates them work and to whom they report progress. In turn the group leader, along with several other group leaders, w ill report to a manager at the next higher level. That manager might have to report to another manager at a higher level, and so on.

There might be problems that cannot be resolved at a particular level. l;or example, additional resources might be needed for some task, or there might be a disagreement with another group. These will be referred to the next higher level of management.

At each higher level more information will be received by fewer people. There is thus a very real danger that managers at the higher levels might be overloaded w ith too much information. To avoid this, at each level the information will have to be summarized.

The larger the project, the bagger the communication problems.

The referral of disagreements to a higher level is sometimes known as escalation.


Control
Information