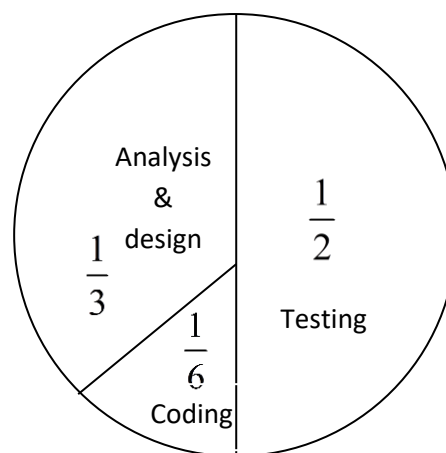


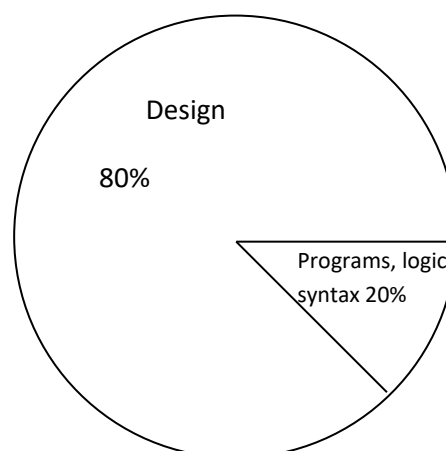
Software Engineering Goals:

1. Low cost of production (SW Cost).
2. High performance efficiency- (meeting people’s requirements).
3. Portability.
4. Low cost of maintenance (Correctness, Adaptive and Enhancement).
5. High reliability (Depends on the SW type).
6. Delivery on time - Scheduling.

1. **Cost of production (SW cost):** The amount spends on SW in the organization very great. It is important to see which parts of SW development project cost most money. The following figure shows the cost of testing is great; coding is a small part of SW development



Error made during the earlier stages of a project tends to be more expensive. The following figure shows the relative costs of fixing mistakes. A syntax error in a program will be automatically detected on a first compilation and easily corrected. But, a design flaw may not be detected until late in system testing and it may involve a whole lot of rework.



Software is the most expensive component in many computer– based systems, whereas only several years ago H/W was the most expensive component. Estimating the cost of SW is not exact sciences since there are too many variables. These variables related to human, technical, environment and political factors. A liable approach to SW cost estimation is important for the continued success of a SW development organization.

2. Performance (efficiency):

In those days the cost and speed of hardware meant that every effort was made to use the hardware memory and CPU as carefully as possible. More recently and due to the increasing speed of computers and the fall in their cost, the emphasis is on meeting people’s requirements.

The trouble with fast run-time and small memory usage is that they are incompatible. Generally, it is necessary to make a judgment about what are the particular performance requirements of SW.

3. Portability:

It means to transfer SW from one type of computer tp another with a minimum cost and efforts. It is very important that programs should be written in such a manner that they may be implemented under more than one computer type and under different operating system. Using high level languages and the establishment of international standards, the prospects increased for the complete portability of the programs (SW).

4. Maintenance:

There are three types of maintenance:

Corrective Maintenance: is fixing bugs in SW that has gone into use.

Adaptive Maintenance: is altering SW because the computer or operating system has changed.

Enhancement Maintenance: is adding new function to the SW due to the user’s need.

The maintenance of existing SW can account for over 60 percent of all efforts expended by a development organization. However software should be maintained with a minimum cost and efforts.

5. Reliability:

Different applications have different requirements for reliability. For example, SW for air traffic should not fail, or human life may be lost. An inventory control system or word processor SW should not fail also, but the impact of failure is less dramatic. We can use the nature of the project to aid in estimating effort & cost to assure reliability.

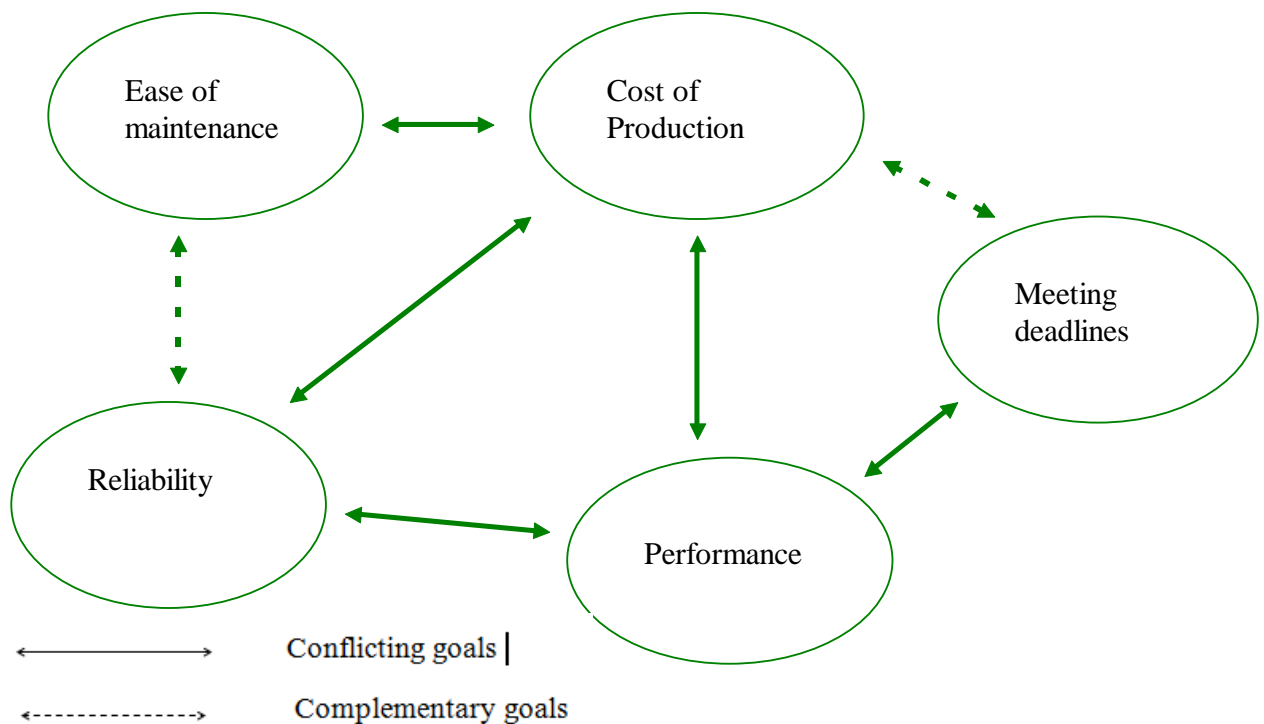
6. Delivery on time:

The major problem faced by SW manager is meeting deadlines (scheduling). The problem is related to the difficulty of predicting how long it will take to develop a SW.

Accuracy in scheduling is very important. A missed schedule creates dissatisfied customers and raise internal costs.

Complementary and Conflicting Goals in SWE:

The trouble with the goals is that very often they conflict with each other. For example, low cost of production and high reliability conflict. A gain high performance and portability with each other e.g. low cost of maintenance and high reliability are complementary. In order to make judgments about any SW development technique, we need to know what aims is tries to meet.



What are the key challenges facing software engineering?

- Coping with legacy systems, coping with increasing diversity and coping with demands for reduced delivery times.
- Legacy systems/ Old, valuable systems must be maintained and updated.
- Heterogeneity/ Systems are distributed and include a mix of hardware and software.
- Delivery/ There is increasing pressure for faster delivery of software.