## Estimating the Size of the Software

Estimation of the size of the software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time that will be needed to build the project. Here are some of the measures that are used in project size estimation:

### **1. Lines of Code (LOC)**

As the name suggests, LOC counts the total number of lines of source code in a project. The units of LOC are:

1. **KLOC:**Thousand lines of code
2. **NLOC:** Non-comment lines of code
3. **KDSI:** Thousands of delivered source instruction

* The size is estimated by comparing it with the existing systems of the same kind. The experts use it to predict the required size of various components of software and then add them to get the total size.
* It’s tough to estimate LOC by analyzing the problem definition. Only after the whole code has been developed can accurate LOC be estimated. This statistic is of little utility to project managers because project planning must be completed before development activity can begin.
* Two separate source files having a similar number of lines may not require the same effort. A file with complicated logic would take longer to create than one with simple logic. Proper estimation may not be attainable based on LOC.
* The length of time it takes to solve an issue is measured in LOC. This statistic will differ greatly from one programmer to the next. A seasoned programmer can write the same logic in fewer lines than a newbie coder.

**Advantages:**

1. Universally accepted and is used in many models like COCOMO.
2. Estimation is closer to the developer’s perspective.
3. Both people throughout the world utilize and accept it.
4. At project completion, LOC is easily quantified.
5. It has a specific connection to the result.
6. Simple to use.

**Disadvantages:**

1. Different programming languages contain a different number of lines.
2. No proper industry standard exists for this technique.
3. It is difficult to estimate the size using this technique in the early stages of the project.
4. When platforms and languages are different, LOC cannot be used to normalize.

### Function Point Analysis

In this method, the number and type of functions supported by the software are utilized to find FPC(function point count). The steps in function point analysis are:

1. Count the number of functions of each proposed type.
2. Compute the Unadjusted Function Points(UFP).
3. Find the Total Degree of Influence(TDI).
4. Compute Value Adjustment Factor(VAF).
5. Find the Function Point Count(FPC).

The explanation of the above points is given below:

**1. Count the number of functions of each proposed type:**

Find the number of functions belonging to the following types:

* External Inputs: Functions related to data entering the system.
* External outputs: Functions related to data exiting the system.
* External Inquiries: They lead to data retrieval from the system but don’t change the system.
* Internal Files: Logical files maintained within the system. Log files are not included here.
* External interface Files: These are logical files for other applications which are used by our system.

**2. Compute the Unadjusted Function Points(UFP):**

Categorize each of the five function types as simple, average, or complex based on their complexity. Multiply the count of each function type with its weighting factor and find the weighted sum. The weighting factors for each type based on their complexity are as follows:

| **Function type** | **Simple** | **Average** | **Complex** |
| --- | --- | --- | --- |
| **External Inputs** | 3 | 4 | 6 |
| **External Output** | 4 | 5 | 7 |
| **External Inquiries** | 3 | 4 | 6 |
| **Internal Logical Files** | 7 | 10 | 15 |
| **External Interface Files** | 5 | 7 | 10 |

**3. Find the Total Degree of Influence:**

Use the ’14 general characteristics of a system to find the degree of influence of each of them. The sum of all 14 degrees of influence will give the TDI. The range of TDI is 0 to 70. The 14 general characteristics are: Data Communications, Distributed Data Processing, Performance, Heavily Used Configuration, Transaction Rate, On-Line Data Entry, End-user Efficiency, Online Update, Complex Processing Reusability, Installation Ease, Operational Ease, Multiple Sites and Facilitate Change.   
Each of the above characteristics is evaluated on a scale of 0-5.

**4. Compute Value Adjustment Factor(VAF):**

Use the following formula to calculate VAF:

*VAF = (TDI \* 0.01) + 0.65*

**5. Find the Function Point Count:**

Use the following formula to calculate FPC:

*FPC = UFP \* VAF*

**Advantages:**

1. It can be easily used in the early stages of project planning.
2. It is independent of the programming language.
3. It can be used to compare different projects even if they use different technologies(database, language, etc).

**Disadvantages:**

1. It is not good for real-time systems and embedded systems.
2. Many cost estimation models like COCOMO use LOC and hence FPC must be converted to LOC.

COCOMO Model

Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.COCOMO is one of the most generally used software estimation models in the world. COCOMO predicts the efforts and schedule of a software product based on the size of the software.

**The necessary steps in this model are:**

1. Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
2. Determine a set of 15 multiplying factors from various attributes of the project.
3. Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

The initial estimate (also called nominal estimate) is determined by an equation of the form used in the static single variable models, using KDLOC as the measure of the size. To determine the initial effort Ei in person-months the equation used is of the type is shown below

**Ei=a\*(KDLOC)b**

**In COCOMO, projects are categorized into three types:**

1. Organic
2. Semidetached
3. Embedded

**1.Organic:** A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. **Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.**

**2. Semidetached:** A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. **Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management system.**

**3. Embedded:** A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. **For Example:** ATM, Air Traffic control.

ccording to Boehm, software cost estimation should be done through three stages:

1. Basic Model
2. Intermediate Model
3. Detailed Model

**1. Basic COCOMO Model:** The basic COCOMO model provide an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

**Effort=a1\*(KLOC) a2 PM**  
                **Tdev=b1\*(efforts)b2 Months**

Where

**KLOC** is the estimated size of the software product indicate in Kilo Lines of Code,

a1,a2,b1,b2 are constants for each group of software products

**Effort** is the total effort required to develop the software product, expressed in **person months (PMs)**.

**xample1:** Suppose a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three model i.e., organic, semi-detached & embedded.

**Solution:** The basic COCOMO equation takes the form:

                Effort=a1\*(KLOC) a2 PM  
                Tdev=b1\*(efforts)b2 Months  
                Estimated Size of project= 400 KLOC

**i)Organic Mode**

                E = 2.4 \* (400)1.05 = 1295.31 PM  
                D = 2.5 \* (1295.31)0.38=38.07 PM

**(ii)Semidetached Mode**

                E = 3.0 \* (400)1.12=2462.79 PM  
                D = 2.5 \* (2462.79)0.35=38.45 PM

**(iii) Embedded Mode**

                E = 3.6 \* (400)1.20 = 4772.81 PM  
                D = 2.5 \* (4772.8)0.32 = 38 PM

**2. Intermediate Model:** The basic Cocomo model considers that the effort is only a function of the number of lines of code and some constants calculated according to the various software systems. The intermediate COCOMO model recognizes these facts and refines the initial estimates obtained through the basic COCOMO model by using a set of 15 cost drivers based on various attributes of software engineering.

**Intermediate COCOMO equation:**

**E=ai (KLOC) bi\*EAF**  
                **D=ci (E)di**

Coefficients for intermediate COCOMO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project** | **ai** | **bi** | **ci** | **di** |
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semidetached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

**3. Detailed COCOMO Model:**Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost driver?s effect on each method of the software engineering process. The detailed model uses various effort multipliers for each cost driver property. In detailed cocomo, the whole software is differentiated into multiple modules, and then we apply COCOMO in various modules to estimate effort and then sum the effort.

The Six phases of detailed COCOMO are:

1. Planning and requirements
2. System structure
3. Complete structure
4. Module code and test
5. Integration and test
6. Cost Constructive mode

Project Scheduling

Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software project manager wants to do the following:

1. Identify all the functions required to complete the project.
2. Break down large functions into small activities.
3. Determine the dependency among various activities.
4. Establish the most likely size for the time duration required to complete the activities.
5. Allocate resources to activities.
6. Plan the beginning and ending dates for different activities.
7. Determine the critical path. A critical way is the group of activities that decide the duration of the project.

The first method in scheduling a software plan involves identifying all the functions required to complete the project. A good judgment of the intricacies of the project and the development process helps the supervisor to identify the critical role of the project effectively. Next, the large functions are broken down into a valid set of small activities which would be assigned to various engineers. The work breakdown structure formalism supports the manager to breakdown the function systematically after the project manager has broken down the purpose and constructs the work breakdown structure; he has to find the dependency among the activities. Dependency among the various activities determines the order in which the various events would be carried out. If an activity A necessary the results of another activity B, then activity A must be scheduled after activity B. In general, the function dependencies describe a partial ordering among functions, i.e., each service may precede a subset of other functions, but some functions might not have any precedence ordering describe between them (called concurrent function). The dependency among the activities is defined in the pattern of an activity network.

Once the activity network representation has been processed out, resources are allocated to every activity. Resource allocation is usually done using a Gantt chart. After resource allocation is completed, a PERT chart representation is developed. The PERT chart representation is useful for program monitoring and control. For task scheduling, the project plan needs to decompose the project functions into a set of activities. The time frame when every activity is to be performed is to be determined. The end of every action is called a milestone. The project manager tracks the function of a project by audit the timely completion of the milestones. If he examines that the milestones start getting delayed, then he has to handle the activities carefully so that the complete deadline can still be met.

**Earned Value Analysis**

In this analysis, a “value” is assigned to each track or work package based on the expenditure forecast. The value assigned is known as the **“planned value (PV)”**. The work that has not yet begun is given a value known as the **“earned value of zero”.** The total value credited to a project is called **“earned value(EV)”**, which is also represented as “money value”.

## **Methods For Earned Value Analysis**

* **0/100 Technique:**The technique where a task is assigned a value of zero until such time that is completed when it is given a value of 100% of the budgeted value.
* **50/50 Technique:** The technique in which a task is assigned a 50% value as soon as it is started and then given a value.
* **75/25 Technique:**The technique where a task is assigned 75% on starting and 25% on completion.
* **Milestone Technique:**The techniquewhere a task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan.
* **Percentage Complete:**In some cases, there may be a way of objectively measuring the amount of work completed.

## **Stages in Earned Value Analysis**

* **Creating the baseline budget:** This is the first stage in setting up EVA. This budget is based on the project plan. It predicts the earned value through time. Normally, it is measured in person hours or workdays, for example: in a software development project.
* **Monitoring Earned Value:**The second stage is monitoring the earned value as the project progresses. This is achieved by monitoring the completion of each task. [Actual cost(](https://www.geeksforgeeks.org/accounting-method-amortized-analysis/)AC) is the actual cost of each task and it can be analyzed and collected.
* **Schedule Variance(SV):** This is the third stage which is measured in cost as EV-PV which is the deviation between planned work and completed work.  
  **Example**: Consider these values,  
  PV =40000  
  EV=35000  
  SV=35000-40000 = -5000

Here the calculated SV value is negative and hence we conclude that the project is behind the original schedule.

* **Time variance(TV):** The difference between the current time and the time when the achievement of the earned value was planned to occur.
* **Cost Variance(CV):** This value is the difference between the actual cost and the earned value. Using this value we can estimate the accuracy of the original cost scheduled for the project. If the [CV values](https://www.geeksforgeeks.org/software-engineering-cost-variance-cv-and-schedule-variance-sv/) are found to be negative, we conclude
* the project is over cost.

## Advantages

* **Project Performance Measurement**: EVA provides a comprehensive method for measuring and assessing the performance of a project. It helps [project managers](https://www.geeksforgeeks.org/6-steps-to-become-a-project-manager/) gain a clear understanding of how well a project is progressing in terms of cost and schedule.
* **Objective Performance Metrics**: EVA relies on objective metrics, making it less susceptible to subjective interpretations. This can lead to more accurate assessments of project performance.
* **Early Issue Identification**: EVA can highlight problems in [project execution](https://www.geeksforgeeks.org/introduction-to-execution-and-ko-meetings/) early, enabling project managers to take corrective actions promptly. This can prevent cost overruns and schedule delays.

## Disadvantages

* **Complexity**: EVA involves complex calculations and terminology, which can be challenging for project teams to understand, especially for smaller projects or teams with limited expertise.
* **Resource-Intensive**: Implementing EVA requires tracking detailed data and maintaining comprehensive records. This can be resource-intensive, and some organizations may lack the necessary tools or resources for effective EVA implementation.
* **Time-Consuming**: Calculating and updating EVA metrics can be time-consuming, which may not be suitable for projects that require quick decision-making and frequent changes.

## **Risk Management**

A software project can be concerned with a large variety of risks. In order to be adept to systematically identify the significant risks which might affect a software project, it is essential to classify risks into different classes. The project manager can then check which risks from each class are relevant to the project.

There are three main classifications of risks which can affect a software project:

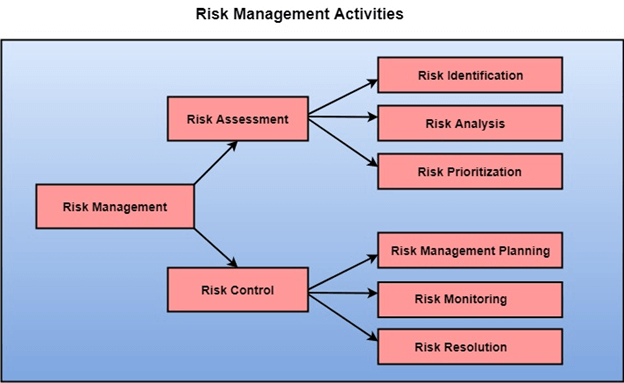
1. Project risks
2. Technical risks
3. Business risks

**1. Project risks:** Project risks concern differ forms of budgetary, schedule, personnel, resource, and customer-related problems. A vital project risk is schedule slippage. Since the software is intangible, it is very tough to monitor and control a software project. It is very tough to control something which cannot be identified. For any manufacturing program, such as the manufacturing of cars, the plan executive can recognize the product taking shape.

**2. Technical risks:** Technical risks concern potential method, implementation, interfacing, testing, and maintenance issue. It also consists of an ambiguous specification, incomplete specification, changing specification, technical uncertainty, and technical obsolescence. Most technical risks appear due to the development team's insufficient knowledge about the project.

**3. Business risks:** This type of risks contain risks of building an excellent product that no one need, losing budgetary or personnel commitments, etc.

## **Risk Management Principle of Risk Management**



## **Risk Assessment**

The objective of risk assessment is to division the risks in the condition of their loss, causing potential. For risk assessment, first, every risk should be rated in two methods:

* The possibility of a risk coming true (denoted as r).
* The consequence of the issues relates to that risk (denoted as s).

Based on these two methods, the priority of each risk can be estimated:

                    p = r \* s

## Where p is the priority with which the risk must be controlled, r is the probability of the risk becoming true, and s is the severity of loss caused due to the risk becoming true. If all identified risks are set up, then the most likely and damaging risks can be controlled first, and more comprehensive risk abatement methods can be designed for these risks.

**. Risk Identification:** The project organizer needs to anticipate the risk in the project as early as possible so that the impact of risk can be reduced by making effective risk management planning.

A project can be of use by a large variety of risk. To identify the significant risk, this might affect a project. It is necessary to categories into the different risk of classes.

There are different types of risks which can affect a software project:

1. **Technology risks:** Risks that assume from the software or hardware technologies that are used to develop the system.
2. **People risks:** Risks that are connected with the person in the development team.
3. **Organizational risks:** Risks that assume from the organizational environment where the software is being developed.
4. **Tools risks:** Risks that assume from the software tools and other support software used to create the system.
5. **Requirement risks:** Risks that assume from the changes to the customer requirement and the process of managing the requirements change.
6. **Estimation risks:** Risks that assume from the management estimates of the resources required to build the system.

**2. Risk Analysis:** During the risk analysis process, you have to consider every identified risk and make a perception of the probability and seriousness of that risk.

It is not possible to make an exact, the numerical estimate of the probability and seriousness of each risk. Instead, you should authorize the risk to one of several bands:

1. The probability of the risk might be determined as very low (0-10%), low (10-25%), moderate (25-50%), high (50-75%) or very high (+75%).
2. The effect of the risk might be determined as catastrophic (threaten the survival of the plan), serious (would cause significant delays), tolerable (delays are within allowed contingency), or insignificant.

## **Risk Control**

It is the process of managing risks to achieve desired outcomes. After all, the identified risks of a plan are determined; the project must be made to include the most harmful and the most likely risks. Different risks need different containment methods. In fact, most risks need ingenuity on the part of the project manager in tackling the risk.

**There are three main methods to plan for risk management:**

1. **Avoid the risk:** This may take several ways such as discussing with the client to change the requirements to decrease the scope of the work, giving incentives to the engineers to avoid the risk of human resources turnover, etc.
2. **Transfer the risk:** This method involves getting the risky element developed by a third party, buying insurance cover, etc.
3. **Risk reduction:** This means planning method to include the loss due to risk. For instance, if there is a risk that some key personnel might leave, new recruitment can be planned.
4. **Risk Leverage:** To choose between the various methods of handling risk, the project plan must consider the amount of controlling the risk and the corresponding reduction of risk. For this, the risk leverage of the various risks can be estimated.
5. Risk leverage is the variation in risk exposure divided by the amount of reducing the risk.
6. **Risk leverage = (risk exposure before reduction - risk exposure after reduction) / (cost of reduction)**
7. **1. Risk planning:** The risk planning method considers each of the key risks that have been identified and develop ways to maintain these risks.
8. For each of the risks, you have to think of the behavior that you may take to minimize the disruption to the plan if the issue identified in the risk occurs.
9. You also should think about data that you might need to collect while monitoring the plan so that issues can be anticipated.
10. Again, there is no easy process that can be followed for contingency planning. It rely on the judgment and experience of the project manager.

**2. Risk Monitoring:** Risk monitoring is the method king that your assumption about the product, process, and business risks has not changed.