**Software Project Estimation**

Estimation is a process to predict the time and the cost that a project requires to be finished appropriately. But in term of software development, it also means of the consideration of the experience of the [software development company](https://www.tpptechnology.com/); the technique they employ; the process they may go through to finish the task. This whole process requires the use of complex tools and good mathematical background knowledge. It is in some cases is the accomplishment of hard work of a whole team. The error margin is, consequently, guaranteed possibly around 5-10%.

The whole process of estimation would cost the company rather considerable cost and time at the very first stage of [building an app](https://www.tpptechnology.com/services/mobile/). But this will make the final result more credible, realistic, and customer-satisfying. Projects especially big ones are advisable to employ this crucial step to avoid unpredictable failure.

**1. Determining the goals and commitments:**

Communicating clearly the goals or the needs of someone with the estimation team would well guide the whole process to the right place it is expected to be. Communication of the goals clearly also allows both the estimation team and the software development company to vision the reality of all the requirements. They, therefore, would plan an appropriate timeline, personnel and employ the suitable technologies to achieve the goals in the most fruitful way. Basing on those mentioned, terms and agreements between the customers and the app developing team would be set.

## ****2. Comprehending the functional scope of the project:****

The most essential factor for sizing a project is to determine what someone want to develop and what would it reach. In some cases, there would be breakdowns in functions according to reference documentation. This lets someone know in-depth about the possible extent and elements of the system which will be estimated.

However, in some cases, the estimation task become much more uncertain and complex. Then, it is necessary that the estimator should follow the following steps to get out.

* Foster communication to enhance comprehension.
* Learn from the existing similar system.
* Spend more effort in studying the industry and the type of project.
* Review the validation and prioritize created functional requirements created.

**3. Paying attention to non-functional needs:**

These functions focus on the “how” the project might work to meet the requirements of the customers which are usually left un-addressed when determining the functional requirements.

Followings are some of typical non-function requirements software estimation process usually left unattended:

* Maintainability
* Scalability
* Security
* Interoperability
* Performance
* High availability
* Usability

These non-functional requirements may account for different importance depending on the type of project or industry. Also, the complexity and the context of different projects could vary the role of the about mentioned functions

## ****4. Setting clear priorities:****

The estimations and limits of the project as money, time, or personnel make obtaining the planned goals much more complex. In these situations, it is important to clarify the utmost importance of each functional priority in order to focus on the most essential ones.

Estimating clearly the priorities from the beginning would also keep the project more on track. Not many changes and additional requirements occur during the time of making the software, which would save a lot of time and effort.

**Software Project Estimation Approaches**

The evidence on differences in estimation accuracy of different estimation approaches and models suggest that there is no “best approach” and that the relative accuracy of one approach or model in comparison to another depends strongly on the context. This implies that different organizations benefit from different estimation approaches. Software project estimation can be done in many ways. Here we have applied four estimation process for CARS Cafeteria Management System.

**Lines of Code Based Estimation Approach**

Lines of code (LOC) is a software metric used to measure the size of a computer program by counting the number of lines in the text of the program's source code. LOC is the metric which measure the size of project by counting the number of instructions in the developed program. Obviously while counting the number of source instructions, lines used for commenting the code and the header lines are ignored. In order to estimate the LOC count at the beginning of a project, one would have to make a systematic guess. So sometimes project divided into modules and sub modules until sizes of different modules can be approximately predicted. Lines of code (LOC) only measures the volume of code, one can only use it to compare or estimate projects or programs that use the same language, and is coded using the same coding standards. To change one is to change the volume of code. A better method to compare without regard to direct volume is to measure the complexity of the software.

**Function Point Based Estimation Approach**

[Function Point (FP)](https://www.geeksforgeeks.org/software-engineering-functional-point-fp-analysis/) is an element of software development which helps to approximate the cost of development early in the process. It may measures functionality from user’s point of view. FPs of an application is found out by counting the number and types of functions used in the applications. Various functions used in an application can be put under five types.

* Number of External Inputs (EI)
* Number of External Outputs (EO)
* Number of external inquiries (EQ)
* Number of internal files (ILF)
* Number of external interfaces (EIF)

FP characterizes the complexity of the software system and hence can be used to depict the project time and the manpower requirement. The effort required to develop the project depends on what the software does. FP is programming language independent. FP method is used for data processing systems, business systems like information systems. The five parameters mentioned above are also known as information domain characteristics.

**Use Case Based Estimation Approach**

A Use-Case is a series of related interactions between a user and a system that enables the user to achieve a goal. Use-Cases are a way to capture functional requirements of a system. The user of the system is referred to as an ‘Actor’. Use case based estimation technique used to forecast the software size for software development projects. It is used when the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) and methodologies are being used for the software design and development. The software size is calculated based on elements of the system use cases with factoring to account for technical and environmental considerations. The use case based estimation for a project can then be used to calculate the estimated effort for a project.

**Class Based Estimation Approach**

Class based estimation method has been designed explicitly for Object Oriented based software. Using the requirements model use cases are developed. The number of use cases may change as the project progresses. From the requirements model number of key classes are determined. The type of interface for the application are categorized and a multiplier is developed for support classes, where the multipliers for no GUI, a text-based user interface, a conventional GUI, and a complex GUI are: 2.0, 2.25, 2.5, and 3.0, respectively. The number of key classes are multiplied by the multiplier to obtain an estimate for the number of support classes. The total number of classes (key + support) are multiplied by the average number of work units per class. Lorenz and Kidd suggest 15 to 20 person-days per class.

**Cars Cafeteria Management System**

CARS Cafeteria is a Cafeteria for teacher and Officers located at Mokarram Bhaban, University of Dhaka. Teaches and Officers from different departments have their lunch in this Cafeteria and different types of events or parties are also held here by booking the whole cafeteria in the evening hours. Several employees work under one manager in this Cafeteria.

The SRS contains the functional, non-functional and the supporting requirements and establishes a requirement’s baseline for the development of the system. The document contains different requirement features for the management system like use case diagram, activity diagram, database modeling, class cards, CRC diagram and behavioral modeling. Based on this requirement baseline the estimation is generated using different approaches.

**Requirements for the CCMS**

**Normal requirements:**

Normal requirements are generally the objectives and goals that are stated for a product or system during meetings with the customer. The presence of these requirements fulfills customers’ satisfaction. These are the normal requirements for our project.

* Users will create an account by providing their credentials.
* A predefined account will be given to system admin.
* System will verify the user’s credentials from database.
* Users can update his/her profile.
* Users can recover their password if forgotten.
* Users must be logged in before doing any operation.
* Users can order food online.
* A memo will be provided after order confirmation.
* Order details will be added into virtual cart.
* Users can reserve table(s) for lunch through online.
* Users can book whole cafeteria after lunch through online.
* If any user reserves cafe which is already booked, he will be assigned into queue. First queued user will be notified after every booking cancellation.
* Users can pay bills through “SSL Commerz” and cash on delivery.
* If any users cancel order or reservation, he will be refunded.
* Estimated time of food delivery will be prompted in the user’s display.
* Staff info, user info, every transaction details will be stored into admin database.
* A replica of user info, order memo and transaction history will be stored in user database.
* If a user updates his/her info, it will be updated in user database first and then the same update will be replaced into Admin Database automatically.
* Admin can manage everyday menu.
* Inventory management.
* If a food is out of stock it will be shown as stocked out.

**Expected requirements**

These requirements are intrinsic to the product or system and may be so elementary that the customer does not explicitly state them. Their absence will be a cause for significant dissatisfaction. Below the expected requirements for our project are briefly described-

* The system will be secured.
* Delivery time for every delivery will be estimated by GPS
* Several users can request for tables and cafeteria at the same time.
* Responsiveness of the System will be expeditious.
* Transaction history of Non Registered users will also be recorded.
* Interactive and attractive graphical user interface.

**Exciting requirements**

These requirements are for features that go beyond the customer’s expectations and prove to be very satisfying when present. Following are some exciting requirements of our project:-

* If a user order certain items frequently, it will be prompted in the user’s homepage.
* All relevant food items according to user’s taste will be suggested.
* After analyzing the orders of previous one month from the users, the sys- tem will show the mostly ordered food items in the admin’s homepage.

**Features**

The features from the above requirements are given below:

* Account creation
* Account verification
* User profile update
* Password recovery through email
* Password recovery through mobile
* Order food
* Virtual card for Order details
* Order memo generate
* Table Reservation for lunch
* Booking of whole cafeteria
* Payment through SSL Commerz and cash on delivery
* Food delivery
* Admin profile
* Staff profile
* Staff availability for delivery food
* Transaction details
* Menu create, update, delete
* Inventory management

For the above feature the estimated line of code given below:

|  |  |
| --- | --- |
| Features | Estimated LOC |
| Account creation  Account verification  User profile update  Password recovery through email  Password recovery through mobile  Order food  Virtual card for Order details  Order memo generate  Table Reservation for lunch  Booking of whole cafeteria  Payment through SSL Commerz and cash on delivery  Food delivery  Admin profile  Staff profile  Staff availability for delivery food  Transaction details  Menu create, update, delete  Inventory management | 3980  2970  3730  2730  2523  9763  2520  6540  6354  5756  11036  7860  13689  4056  2670  3578  10759  12536 |
| Estimated lines of code | 113050 |

**LOC based estimation for CCMS (Duration, Cost and Human Resource)**

Here, Average productivity for systems of this type = 620 LOC/pm. Burdened labor rate =$8000 per month, the cost per line of code is approximately $13.

Based on the LOC estimation the total estimated project cost is 113050 \* $13 = $1469650 and estimated effort is (113050 / 620) pm = 183 person-months

**Estimation Using Function Point Approach**

After analyzing all the features, use case diagram, class diagram and requirements the information’s of the domain value is given below:

**Number of External Inputs Application:**

* User Create
* User Login