

# Software Estimation Techniques - Common Test Estimation Techniques used in SDLC

In order to successful software project & proper execution of task, the Estimation Techniques plays vital role in software development life cycle. The technique which is used to calculate the time required to accomplish a particular task is called *Estimation Techniques*. To estimate a task different effective **Software Estimation Techniques** can be used to get the better estimation.

Before moving forward let's ask some basic questions like What is use of this? or Why this is needed? or Who will do this? So in this article I am discussing all your queries regarding **ESTIMATION**.

## What is Estimation?

"**Estimation** is the process of finding an **estimate**, or approximation, which is a value that is usable for some purpose even if input data may be incomplete, uncertain, or unstable."

[Wiki Definition]



The Estimate is prediction or a rough idea to determine how much effort would take to complete a defined task. Here the effort could be time or cost. An estimate is a forecast or prediction and approximate of what it would Cost. A rough idea how long a task would take to complete. An estimate is especially an approximate computation of the probable cost of a piece of work.

The calculation of **test estimation techniques** is based on:

- Past Data/Past experience
- Available documents/Knowledge
- Assumptions

- Calculated risks

Before starting one common question arises in the testers mind is that **“Why do we estimate?”** The answer to this question is pretty simple, it is to avoid the exceeding timescales and overshooting budgets for testing activities we estimate the task.

**Few points need to be considered before estimating testing activities:**

- Check if all requirements are finalized or not.
- If it not then how frequently they are going to be changed.
- All responsibilities and dependencies are clear.
- Check if required infrastructure is ready for testing or not.
- Check if before estimating task all assumptions and risks are documented.

## Software Estimation Techniques

There are different Software **Testing Estimation** Techniques which can be used for estimating a task.

- 1) Delphi Technique**
- 2) Work Breakdown Structure (WBS)**
- 3) Three Point Estimation**
- 4) Functional Point Method**

### **1) Delphi Technique:**

Delphi technique – This is one of the widely used software testing estimation technique.

In the Delphi Method is based on surveys and basically collects the information from participants who are experts. In this estimation technique each task is assigned to each team member & over multiple rounds surveys are conducted unless & until a final estimation of task is not finalized. In each round the thought about task are gathered & feedback is provided. By using this method, you can get quantitative and qualitative results.

In overall techniques this technique gives good confidence in the estimation. This technique can be used with the combination of the other techniques.

### **2) Work Breakdown Structure (WBS):**

A big project is made manageable by first breaking it down into individual components in a hierarchical structure, known as the **Work breakdown structure**, or the WBS.

The WBS helps to project manager and the team to create the task scheduling, detailed cost estimation of the project. By using the WBS motions, the project manager and team will have a pretty good idea whether or not they've captured all the necessary tasks, based on the project requirements, which are going to need to happen to get the job done.

In this technique the complex project is divided into smaller pieces. The modules are divided into smaller sub-modules. Each sub-modules are further divided into functionality. And each functionality can be divided into sub-functionalities. After breakdown the work all

functionality should review to check whether each & every functionality is covered in the WBS.

Using this you can easily figure out the what all task needs to completed & they are breakdown into details task so estimation to details task would be more easier than estimating overall Complex project at one shot.

Work Breakdown Structure has four key benefits:

- **Work Breakdown Structure forces the team to create detailed steps:**  
In The WBS all steps required to build or deliver the service are divided into detailed task by Project manager, Team and customer. It helps to raise the critical issues early on, narrow down the scope of the project and create a dialogue which will help make clear bring out assumptions, ambiguities, narrow the scope of the project, and raise critical issues early on.
- **Work Breakdown Structure help to improve the schedule and budget.**  
WBS enables you to make an effective schedule and good budget plans. As all tasks are already available so it helps in generating a meaningful schedule and makes scheming a reliable budget easier.
- **Work Breakdown Structure creates accountability**  
The level of details task breakdown helps to assign particular module task to individual, which makes easier to hold person accountable to complete the task. Also the detailed task in WBS, people cannot allow hiding under the "cover of broadness."
- **Work Breakdown Structure creation breeds commitment**  
The process of developing and completing a WBS breed excitement and commitment. Although the project manager will often develop the high-level WBS, he will seek the participation of his core team to flesh out the extreme detail of the WBS. This participation will spark involvement in the project.

### 3) Three Point Estimation:

Three point estimation is the estimation method is based on statistical data. It is very much similar to WBS technique, task are broken down into subtasks & three types of estimation are done on this sub pieces.

Optimistic Estimate (Best case scenario in which nothing goes wrong and all conditions are optimal.) = A

Most Likely Estimate (most likely duration and there may be some problem but most of the things will go right.) = M

Pessimistic Estimate (worst case scenario which everything goes wrong.) = B

Formula to find Value for Estimate (E) =  $A + (4 * M) + B / 6$

Standard Deviation (SD) =  $(B - A) / 6$

Now a days, planning poker and Delphi estimates are most popular *testingtest estimation techniques*.

### 4) Functional Point Method:

Functional Point is measured from a functional, or user, point of view.

It is independent of computer language, capability, technology or development methodology of the team. It is based on available documents like SRS, Design etc.

In this FP technique we have to give weightage to each functional point. Prior to start actual estimating tasks functional points are divided into three groups like Complex, Medium & Simple. Based on similar projects & Organization standards we have to define estimate per function points.

Total Effort Estimate = Total Function Points \* Estimate defined per Functional Point

Let's take a simple example to get clearer:

	<b>Weightage</b>	<b>Function Points</b>	<b>Total</b>
<b>Complex</b>	5	5	25
<b>Medium</b>	3	20	60
<b>Simple</b>	1	35	35
<b>Function Total Points</b>			<b>120</b>
<b>Estimate defined per point</b>			<b>4.15</b>
<b>Total Estimated Effort (Person Hours):</b>			<b>498</b>

Advantages of the Functional Point Method:

- In pre-project stage the estimates can be prepared.
- Based on requirement specification documents the method's reliability is relatively high.

Disadvantages of **Software Estimation Techniques**:

- Due to hidden factors can be over or under estimated
- Not really accurate
- It is based on thinking
- Involved Risk
- May give false result

- Bare to losing
- Sometimes cannot trust in estimate

### **Software Estimation Techniques Conclusion:**

There may be different other methods also which can be effectively used for the project test estimation techniques, in this article we have seen most popular **Software Estimation Techniques** used in project estimation. There can't be a sole hard and fast rule for estimating the testing effort for a project. It is recommended to add on to the possible knowledge base of test estimation methods and estimation templates constantly revised based upon new findings.

Source:

<http://www.softwaretestingclass.com/software-estimation-techniques/>