



Estimation of effort using Function Points

-Vijay Shankar

Vijay Shankar

© Cognizant Technology Solutions, Feb 2003.

All rights reserved. You may make one attributed copy of this material for your own personal use.

For additional information contact Vijay Shankar at vjathma@yahoo.com

Prologue

This paper attempts to portray Function Point Estimation methodology and also provides a tool that could be used for the estimation. The tool is **easy-to-use** XL spread sheet that can be downloaded/edited/modified/verified and used too!

Why estimate?

An omelet, promised in two minutes, may appear to be progressing nicely. But when it has not set in two minutes, the customer has two choices

Either wait & eat (or) eat it raw!

If the customer does not want to eat it raw as well as he does not want to wait, then the cook has too has a choice - he can turn the heat on and

Get the omelet burnt in one part and raw in another!

In software industry too customers & vendors have the same choices. One most important factor that requires consideration is setting the expectations right upfront. This can be done first by getting the estimation right.

Characteristics of a Good Estimate

A good estimating methodology should be :-

1. Explainable
2. Sufficient amount of granularity exists for the estimates

Often, in practice the estimates between estimators don't necessarily concur and the point of contention is unexplainable!. The case exists, if estimation is not done using a scientific methodology. One could use any methodology as preferred but the key consideration is that the estimates should be "explainable". FP is one such method to help you to solve this problem.

What is Function Points?

Function Point is an estimation technique that could be used to estimate the effort for a software development engagement. This technique could be used at any point in the life cycle of the project.

Function points is a scientific method to quantify functionality provided to the user, with the following characteristics

- Measures in terms of functionality that the user requests and receives
- Measures software development and maintenance rates and size independently of the technology used for implementation;
- Provides a normalizing measure across projects.

This being a scientific method provides almost near accurate effort when used properly.

Again don't be paranoid on the word scientific; it does use complex functions and calculations to estimate the effort. The whole effort of this paper is to make sure that FP is understood and could be used; hence I have abstracted all the calculations. An XL (no macros) is attached with this document. This XL tool I have constantly used for estimation.

Folks with academic interest are welcome to check out and validate what the sheet has! or check out with IFPUG.

To the best of my knowledge the formulas adheres to what IFPUG dictates!. Don't hold me responsible for any errors or omissions. You test it out before you start using it!

Benefits of using FP

Using FP invariably has lot of benefits, I have attempted to list a few, and the list is not encyclopedic!

- Better and reasonable time to market!
- Improves Customer satisfaction
- Improves quality
- Increases profitability
- Reduces resource burnt out
- No finger pointing, "knowing looks" and high priority memos!

The methodology

The basis of FP uses **Data functions** and **Transactional Functions** for estimating the effort of an engagement.

Data functions talk about the storage of the data and the transactions functions talk about the processing complexity.

The methodology – Data functions

Let's start with Data functions; you need to understand a few definitions before we divulge into the process of counting using FP. The process kicks off identifying 2 major categories as

- Internal Logical Files
- External Logical Files

Internal Logical Files (ILF) are defined as logical files, in which user identifiable data is maintained and/or managed by an elementary process.

In short this could be perceived as the logical files that the application maintains within its scope.

External Interface Files (EIF) are defined as logical files that the application does not maintain or is at out of scope of the system considered, but the application reads data from these files.

In short this could be understood as files aren't maintained by the application, but data is being read from them.

Data Element Type (DET) is defined as user recognizable, non-recursive field. A data element is information that either is stored on an Internal Logical File or that is used to invoke a transaction. *Example: - Fields in an input screen is articulated as DETs.*

Record Element Type (RET) is defined as user recognizable subgroup of data. This could be visualized as subgroups with Parent-Child relationship.

Consider an Order processing system, you need to specify customer name, number, date, amount and following this, the child records would contain order details (individual order entries).It has to be maintained together. Hence it is considered as 2 RETs

The methodology – Transactional Functions

The next step is to identify and count the Transactional Functions. Transactional functions are classified as:-

- External Inputs
- External Outputs
- External Queries

External Inputs (EI) are defined as processes that feed data into the system from outside the system boundary and EI maintains one/more ILF

It is conceived as a unique process that maintains ILFs (or) Data elements used by the EI functions are unique (or) ILF/EIFs referenced are unique.

External Outputs (EO) are defined as processes that send data outside the system boundary; creates derived data; maintains one or more ILF; uses processing logic.

It is conceived as unique process that maintains ILFs (or) Data elements used by the functions are unique (or) ILF/EIFs referenced are unique.

External Inquiries (EQ) are defined, as processes that has primary intent to present data to the user or as a process that sends data out of the system boundary.

The EQ does not create derived data or does not maintain one/more ILFs or does not use processing logic.

It is conceived as unique process (or) Data elements used by the EQ functions are unique (or) ILF/EIFs referenced are unique

File Type referenced (FTR) is defined as the file type referenced by the transaction functions.

Case study

Consider a use case that has the scope of recording what an employee does as a part of his work during a typical day. The use-case assumes that employee opens up a web page or an application and types his number, the system displays the employee name in the screen and then the user enters the date, duration and chooses the activity from pre-defined list.

We assume that the employee name & necessary details and the pre-defined activity list are maintained by another application and are not considered to be in the scope of the application that is considered at this moment. The following is the input screen using which the input is captured.

The screenshot shows a web form with the following elements:

- Two stacked text input fields: "Employee No" and "Employee Name".
- Three input fields in a row: "Date", "Duration", and "Activity" (a dropdown menu).
- A "Submit" button located below the "Date" and "Duration" fields.

In the input screen, we assume that the user types in the employee number – when the user types in the employee number, the employee name is queried from the database and displayed on the screen. Next the employee types in the date, duration of the activity and then selects the activity from the combo which is pre-defined in the database. After doing this the user clicks **Submit** to store the data in the database/file.

Case study - Estimating Data Functions

Now, let's start our counting process. As a first step, we have to determine data functions.

Let's put the following questions

1. Does the application maintain any files?
2. Does the application reads data from the files but does not maintain them?

For the first question – you would identify a file something like a **Time Recorder file** being maintained by the application, this file is used to store the information of the Activity log. This is counted as 1 ILF. Then we have to count the number of data elements that is stored. In this case it is employee number, date, duration and activity. Hence the FP count for this ILF would be the following:-

Time Recorder ILF:-

DETs : - 4 (employee number, duration, activity & date)
RET – 1 (since there is no PCR between the data stored)

For the second question – we assumed that the employee information (name) and pre-defined activity list exists. Since these files are not maintained by the application, we count them as 2 separate EIFs . We would have **Employee details** and **Activity** as the EIFs.

Employee EIF:-

DETS – 2 (Employee name and Employee number)
RET – 1

Activity EIF:-

DETS – 1 (Activity name)
RETS – 1

That's it!. Just enter this information in the FP Estimator XL / **Data functions** sheet. This will estimate the data functions.

Do it your self!

[Fill in the **Data functions** in the XL sheet]

Case study - Estimating the transactional functions

Continue the process to identify and count transactional functions. We have to classify the functions as External Inputs, External Outputs and External Queries as defined earlier. In our use-case, we have one data input screen and 2 queries, that fills data into the Employee name and Activity UI components. There are no reports/outputs considered in the current scope.

Counting this, we have

EI – Data capture Screen:-

EI – 1

DET – 5, FTR – 1 (We store the employee number, activity information, data, duration and count 1 for Submit button – since it invokes a transaction; we reference one file – hence one FTR)

EQ- Employee information and Activity information:-

EQ – 2 – we are considering this as EQ, since it does not use any complex logic or does it create any derived data.

So for each EQ we have,

DET – 1, FTR – 1

EO- NONE

In the use case considered we don't have any EO – 0

Do it your self!

[Fill in the **Transactional functions** the XL sheet]

FP Estimator tool



FP Estimator.xls

Note: - If you are reading this out of PDF file, do mind to check out the FP estimator XL file also with this document. This could be a handy tool!

If you have completed the **–do-it-yourself–** section you should see the following values in the summary table of the XL.

| <i>Project Details</i> | |
|------------------------|--|
| Project | |
| Project Code | |
| Project Manager | |
| Estimated by | |
| Date | |

| <i>Estimate</i> | |
|----------------------------------------------------------|-------|
| Data functions | 17.00 |
| Transactional functions | 9.00 |
| Total FP Unadjusted | 26.00 |
| Total FP Adjusted | 18.20 |
| Value Adjustment Factor | 0.70 |
| Productivity Base Line - Function points/week per person | 4.00 |
| Effort in Person Weeks Unadjusted | 6.50 |
| Effort in Person Days - Unadjusted | 32.50 |
| Effort in Person Weeks - Adjusted | 4.55 |
| Effort in Person Days - Adjusted | 22.75 |

At this stage you got the hold of **unadjusted function point count**. After this stage, you have to identify the system characteristics and the impact on the estimates based on the system characteristics.

System characteristics are represented under various heads as discussed below and the sum of the influence is considered as the Value Adjustment Factor to the estimate. This can be calculated by selecting **Value Adjustment Factor Sheet** and recording the system characteristics in the XL sheet.

As described, FP uses the following system characteristics in any system. The characteristics are listed as below.

- ✚ Data communications
- ✚ Distributed Data Processing
- ✚ Performance
- ✚ Heavily Used Configuration
- ✚ Transaction Rate
- ✚ Online Data Entry
- ✚ End User Efficiency
- ✚ Online Update
- ✚ Complex Processing
- ✚ Reusability
- ✚ Installation Ease
- ✚ Operational Ease
- ✚ Multiple Sites
- ✚ Facilitate Change

Once you complete the entries in the Value Adjustment factor sheet, you would get the **adjusted FP count**.

If you know the Productivity baseline in terms of FP per week, you can arrive at the effort required for the engagement. The productivity baseline would vary from organization to organization and from technology to technology. In the XL, that I have used, I have assumed an arbitrary

productivity baseline factor of 4 FP per week, you can record-in what your organization has!. If this is done, you have calculated the effort for the system under consideration.

Once the effort is estimated, we could distribute it amongst different phases of the project and hence re-sourcing & scheduling can also be arrived at.

I have discussed a simple use-case here in this paper; in practice you may get at counting complicated use-case too, don't get rattled off just keep following the process. I sign off wishing you a Happy counting...

-- End of document --