EstimatorPal Approach to Software Estimation

Murali Chemuturi, Chief Consultant

Background

Software Estimation has been a research topic for some time now and it continuous to be so. This is so as the size measurement has been defying efforts to evolve a universally acceptable software size measure. Yes, there are many popular software size measures but there is no software measure that has been recognized by all international standards bodies have agreed upon. And – new measures are still sprouting.

Suffice it to say here that the understanding of software size is not as well understood as the measures for, say, mass, weight, length, distance, heat, temperature and so on.

There are many well-documented software estimation methodologies and sizable followings. There are a plethora of tools too for software estimation. All of these have software size and effort estimation and some have cost estimation too.

Some of these tools take the minimum input and give a single-figure estimate. Some take size input and give multi-value estimates. The variety is infinite.

This, as I see, is the software estimation scenario.

EstimatorPal approach to aspects of software estimation

EstimatorPal perceives that a software estimation tool is not complete unless it provides for all the four aspects of software estimation. Thus, EstimatorPal provides for estimation of software size, development-effort, development-cost and development-schedule.

EstimatorPal approach to measures of software size

Given the scenario that the universal agreement on software measure is non-existent, EstimatorPal perceives that as many popular software sizing techniques as possible needs to be included in the software estimation tool. The following software size measures are included in the EstimatorPal –

1. Function Points
2. FPA mark II
3. Object Points
4. Use Case Points
5. SSU (Software Size Units)
6. Test Points
7. LOC
8. Intermediate COCOMO
9. Task Based Estimation

Software Classification

I have not come across an accepted classification of software being developed in the world. Here is how I classify –

1. System software – which facilitate others using a computer. I put in this class - operating systems, software development environments including programming languages, DBMS, middleware, software tools
2. Real-time software – which control machines where the response time is critical. I put in this class – all machine control software that is embedded in machines – including CNC machines, aero planes, rockets, weapon systems, robots and so on
3. Commercial software – that is used in businesses. This software has significant emphasis on user interface and is likely to be used by persons who are not very well versed in computers and needs to be tailored to suit the personal and organizational preferences. I put all software that is used in business organizations for the purpose of recording and analyzing business transactions.

A software estimation tool needs to cater to all the above-mentioned software classes.

EstimatorPal approach to classes of software

System software and Real-time software focus mostly on programming. LOC (Lines of Code) is, in our opinion, he most suited measure for measuring the software size. EstimatorPal provides for this technique.

Commercial software focuses more on user interface. EstimatorPal provides for this class of software thru Function Points, FPA Mark II, Object Points, Use Case Points, LOC and Intermediate COCOMO.

Of late testing projects are increasing for IV&V (Independent Verification and Validation). Therefore, EstimatorPal provides for measuring Test Points too.

EstimatorPal approach to Location of Software Development

Gone are the days when software development was completely in-house. Now software development takes place both in-house as well as out side the organization. The software development may take place across continents.

Therefore, software estimation cannot be restricted for in-house use alone. It needs to be agreed upon by persons internal to the organization and by the persons outside the organization – may be the client or the sub-contractor.

Now, when we make software estimate, we are aware that there is one more software estimate in existence (or would be in existence shortly) somewhere and that our estimate

would be compared and contrasted against that other independent estimate. This makes it necessary that our estimate is auditable and reconcilable.

EstimatorPal provides for auditable estimates and hence reconcilable.

EstimatorPal approach to estimation inputs

There was a dilemma of whether take only a few inputs and build the estimate or allow the estimator to provide more inputs.

Take a few parameters and produce an estimate presupposes that the products are well standardized and the modifications to products depend on the customer preferences. Say in cars, a model comes with 95% or more standard features and the customer is allowed to add a few more features. This is an example of a standard product and estimation can be automatically produced using parameters.

In software development such scenarios are rare. Therefore, taking a few parameters as input and producing an automatic estimate would be too restrictive and the accuracy of such an estimate becomes suspect.

EstimatorPal takes the approach to facilitate the estimator to produce an estimate from the scratch in an auditable manner. The estimator would be facilitated to enter all the aspects, components and factors needed to build the estimate. The tedium of entry would be reduced to the minimum thru click-and-select facility in as many places as possible; provide for programmed decision facility as much as feasible and make the interface as intuitive as possible.

EstimatorPal approach to productivity

The discipline of Industrial Engineering itself accepts that productivity varies from facility to facility. What is more, the productivity keeps on improving with introduction of more and better tools, better methodologies, better environment, training etc. Therefore, it is best to leave it to the estimator to specify the productivity - may be it is Person Hours (or Days) per unit (Function Point etc.).

EstimatorPal provides for selecting the appropriate productivity to the estimator.

EstimatorPal approach to cost estimation

Software development effort in Person Hours (or Days) is the single biggest contributor to software development cost. However, there will be other costs including travel cost, lodging and boarding, transport, computer and network time and so on. These costs will vary from place to place and organization to organization. It is also possible that all of these are applicable to all projects – but it is certain that the applicability will vary from project to project.

EstimatorPal retrieves the software effort from the estimate and facilitates selection of additional cost items as applicable to project at hand. Effort is converted to Person Days and allows input of cost of a Person Day. All computations are automatic. The estimator focuses only on ensuring that all relevant cost items are included and their quantity is appropriately entered.

Thus EstimatorPal allows freedom and takes away the tedium.

EstimatorPal approach to development schedule

It is well recognized that project scheduling calls for human ingenuity owing to the following aspects -

i. The granularity for the activities to be performed in order to complete the project varies from project to project, even in similar projects
ii. Resource availability and their allocation has a significant impact on the schedule
iii. The variety of human resources that can be employed on the project has been increasing in recent times.
iv. The working environment, tools usage and the process-orientation do have a significant impact on the schedule

Therefore, there is no substitute to professionals in the preparation of the project schedule.

What is attempted in EstimatorPal is to assist the professional in preparing the schedule and to the extent possible, reduce the time he needs to spend on the schedule preparation activity. There are many tools dedicated to project scheduling activities that are built on techniques like PERT/CPM - EstimatorPal tries to complement those tools than supplant them. EstimatorPal provides a facility to export the schedule to an Excel sheet as an intermediate file for export to any of these tools. Therefore, please note that the schedule produced by this software is a rudimentary schedule and may need improvement by you.

Schedule preparation in EstimatorPal proceeds in three steps, namely,

i. Prepare an initial schedule
ii. Prepare a detailed schedule
iii. Export the schedule

EstimatorPal provides for manpower in two classes, namely Analysts and Programmers. Analysts carry out activities such as requirements analysis, design, test planning etc. Programmers carry out activities such as coding, and testing.

EstimatorPal takes effort days and computes the duration based on the number of persons allocated for the activity.

EstimatorPal provides for eight phases in the preliminary schedule, which is made by the estimator.

It is recognized that scheduling increases the person days to be spent due to fragmentation of days between activities. Therefore, EstimatorPal also allows the scheduled person days to be more than the estimated person days.

No intervention is necessary in building the detailed schedule – EstimatorPal automatically adjusts the preliminary schedule into detailed schedule comprising of fifty-six activities.

Export is possible to MS-Excel – why? – it is possible to export from excel to many scheduling software packages – that is the reason, to select MS-Excel as vehicle for schedule export.

Overall Approach of EstimatorPal to software estimation

On the whole, EstimatorPal approaches software estimation in an integrated manner combining size, effort, cost and schedule estimation into a seamless thread and making it as simple and intuitive as possible while at the same time making all estimates auditable.

The overriding objective is to ensure accurate and auditable estimates yet make it easier for the estimators.

Please feel free to send your feedback to Murali Chemuturi – murali@chemuturi.com