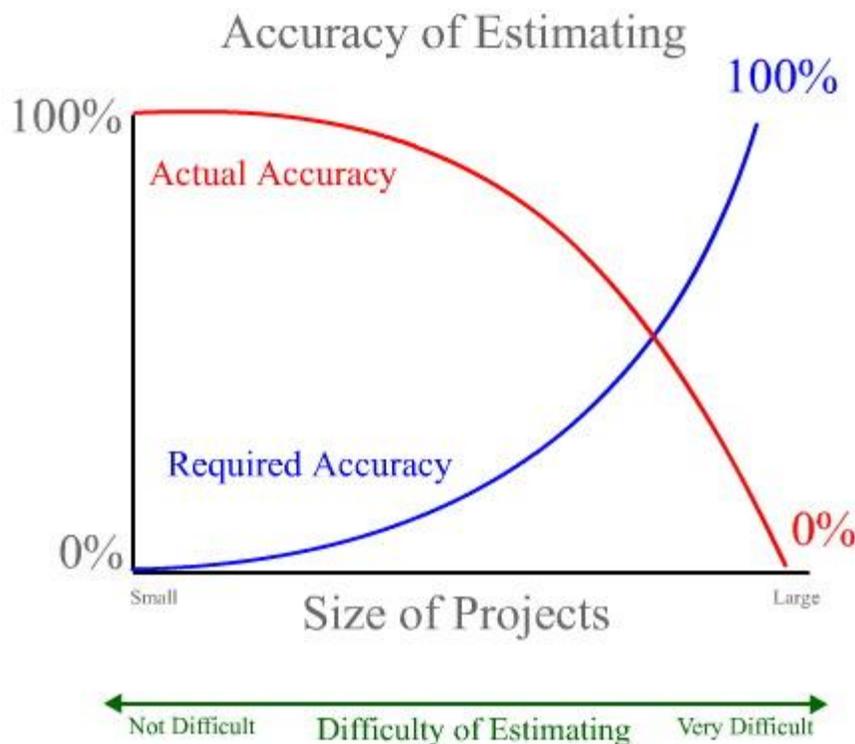


BEST PRACTICE Estimating Software Development

Introduction

Many software managers struggle with estimating projects. As can be seen from the following chart, the inherent problem with estimating is that small projects can be very easy to estimate, but the required accuracy is not important. On the other hand, large projects are very difficult to estimate, but the required accuracy is very important.



Characteristics of a Good Estimate

Explainable - Often I ask people how they came up with their estimates. To my surprise many people can not explain how the estimate was derived. A good estimate should have a sufficient amount of granularity so it can be explained. A project manager should ask:



What steps were used to derive the estimate?

If a project is going to be a replacement project for an existing application then a logical first step would be understand the size (amount of functionality) of the existing application. This is the minimal amount of acceptable functionality that needs to be delivered.

Another logical step is to analyze any requirements for the new application or project.

If historical data is used, then what was the source of the data?

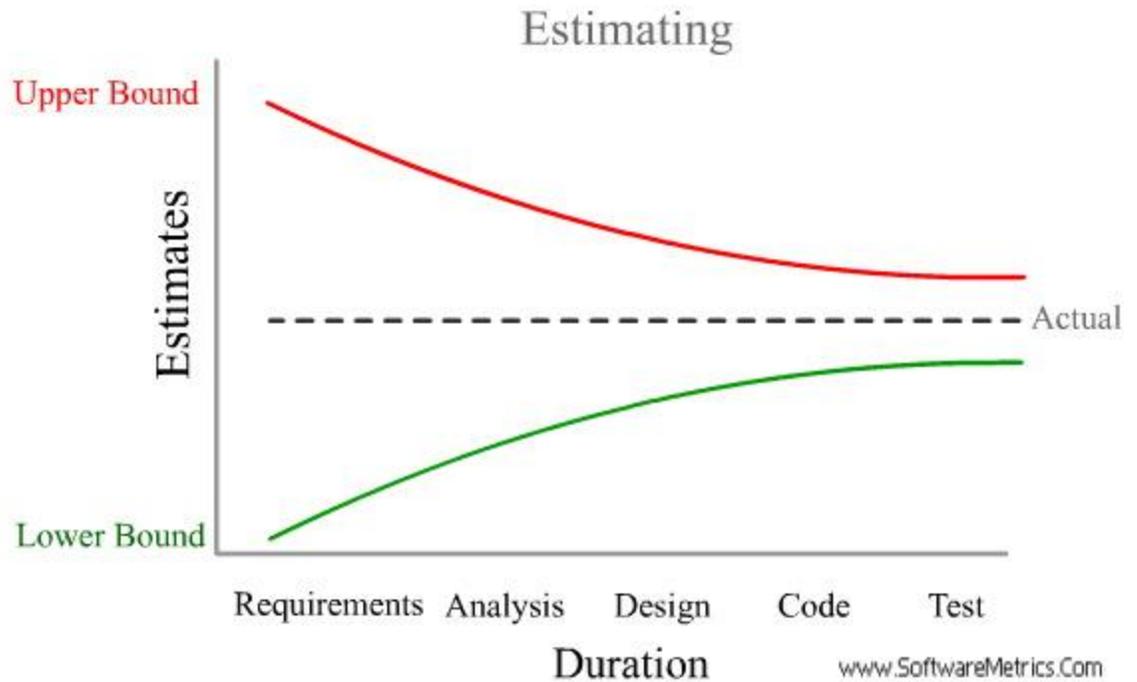
In the absence of any historical data you can use some [industry wide data.](#), but the best data is the data that you collect and glean from your own environment.

There are many ways to collect data from your own organization. Perhaps the easiest way is to examine a few past projects. It is not important to examine every project, but take a sample of several past projects. It is wise to examine the productivity (and what impacted the project) on a project that did not go well. Additionally, it is a good idea to examine a project that was on time and on budget and understand why.

What assumptions were made? And what is the potential impact of the assumption being incorrect

When I create an estimate for an organization I may have a list of assumption. For example, I recently worked with an organization that did not specify inquiries, drop down lists, but select data tables were included in data model. I made the assumptions that there would be at least one simple inquiry (EQ), plus some sort of ability to maintain the information in the select tables.

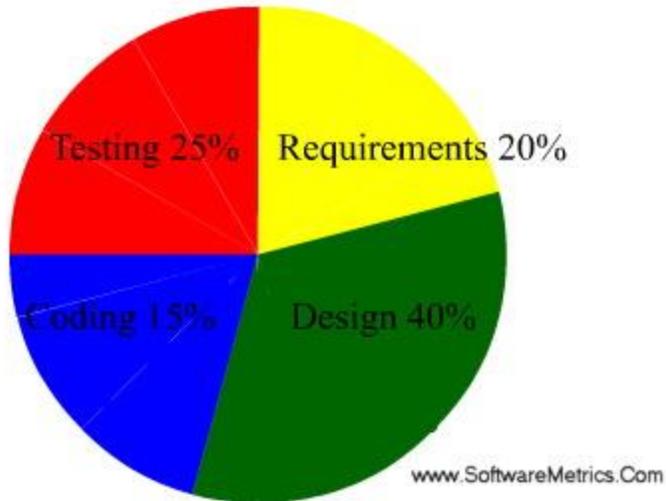
Revisable - as the project goes on the project should be revisable. That is, as more and more information is known about a project the estimate should be revised.



It is important to develop upper and lower bounds for a project estimates. The chart above represents how estimates need to converge on the actual as the project goes on.

Historical Ratios

The following pie chart represents the typical breakdown of effort by major phase. Again, this chart represents a typical project. It would be important to develop ratios for your own organization.



Understanding past historical ratios one can better understand how to revise project estimates. For example, assume a project was estimated at 200 hours. Then it would be expected to spend 40 hours ($200 * 20\%$) on requirements. But if the actual time spent in requirements was 50 hours then the estimate would need to be revised upward.

It would be incorrect to assume that the project is only late 10 hours (50 actual hours - 40 estimated hours). In fact the project would be over budget. In fact, this project would be over budget by 25% (percent change from 40 to 50).

The total number of hours for the project has gone up 25%. That means that the original estimate needs to be revised and 25% added or 50 hours ($200 * 25\%$). Therefore the new estimate would be 250 hours.

Major Phase	Estimated	Actual	Revised	Difference
Requirements	40	50		+10
Design	80		100	+20
Coding	30		38	+18
Testing	50		63	+13
Totals	200		250	+50



Estimating User Documentation

Function Points multiplied by 2.5 - 3.5 estimates the number of pages of user documentation. As the application gets larger the number of pages (or Pages/FP rise). The reason for this is not only is the functionality of a particular part of the application described, but also how it relates to other functionality.

What seems to impact productivity?

One of the keys to creating good estimates is understanding what impacts productivity.

Size

The greatest impact on productivity is the size of the project. As size [increases productivity falls](#) project becomes larger and larger marginal cost begins to rise.

Individual contributor skill sets

- 1/2 to 5 hours per function point for small projects (less than 100 function points).
- 20 - 60 hours per function for large project (greater than 7,000 function points).

This is a major reason why small software projects can have such wide swings in productivity rates. Hence, productivity of inexperienced to experienced software developer can be 1,000 percent (10 times) for small projects and about 300 percent on larger projects (3 times).

Management Skills

For small projects management skill sets have little or no impact, but on larger projects Management skill sets become more important.

Management skill flattens the productivity curve.

Technology

In the 90's it seemed that Java and internet software projects had very high productivity. The reason for this is that most of these projects were relatively small (as compared to mainline or mission critical software applications). As internet applications began to solve more and more business problems and become larger productivity rates (productivity curves) fell in line with more



traditional software projects. ***The choice of technology will shift the entire productivity curve upward or downward.***

JAVA and internet applications impact productivity about 20% for small projects (less than 50 function points) and 3% on large projects (greater than 15,000 function points).

The business of estimating software. This video is part of the online function point training course. You can sign up for the course by visiting <http://www.MetricsTraining.Com>

There are two other articles that may be of interest to you [Estimating Data](#) and [Estimating Defects and Test Cases](#).

Source of Data

Over the past seven years of consulting I have gathered data from over a 100 different companies and nearly 1,000 different projects. Additionally, I have collected data and information from surveys and other studies.

To see other full articles click [here](#). We often post new articles and information. Please send an [email](#) and we will send notifications when new information is posted.

References

Website – Estimating Software Development, www.softwaremetrics.com