

Accurate Project Costs -- How to obtain them, improve estimates with them and dramatically increase your effectiveness by knowing them.

By Curt Finch, CEO, Journyx

Critical Business Issue (CBI) – Something that might get me fired or put in jail if I screw it up again.

For most IT managers, inaccurate estimates constitute a CBI. Why are accurate estimates so important?

1. **Inaccurate estimates cause over-commitment.** Have you ever worked in a place where ten percent of ten projects got done instead of 100 percent of one project? Ultimately, nothing is accomplished and everyone is totally stressed out. Inaccurate estimates cause over-commitment of time, yet not much gets done.
2. **Inaccurate estimates cause bad decisions.** “Inaccurate” usually means “too low.” When this happens, the return on investment (ROI) calculation shows the project as ‘worth it’ when it is not.

How can you stop wasting your company’s resources on projects that are not worth corporate time? HP gave us the answer over a decade ago in Robert Grady’s book, “Practical Software Metrics for Project Management and Process Improvement.” In it, Grady provides a few insightful rules of thumb that are backed up by real numbers, such as:

- 6-8 percent of software project time is spent in the requirements and specifications phase.

This is based on statistics from numerous software projects at HP, so the percentage at your company may be different. The good news is that for projects of a particular flavor, you will find that the percentage usually doesn’t vary wildly, so you can use it to predict all project effort required for completion.

Let’s take an example group of projects in an imaginary IT shop in a large company, XYZ corporation. XYZ’s IT shop likes to write special applications on a regular basis (rather than be slaves to SAP or Oracle) and they’ve discovered the following:

1. All projects have the following phases:
 - a. Requirements: What the internal client wishes XYZ would do
 - b. Specifications: What XYZ is actually going to do.
 - c. Design: How XYZ is going to do it.
 - d. Coding: XYZ writes the application.
 - e. Testing: XYZ tests that it works internally.
 - f. Rollout: XYZ puts it into production.
 - g. Debug: XYZ does the first 90 days of maintenance.

2. After tracking time accurately on seven projects of this flavor, XYZ's CIO finds that 6 to 15 percent of engineering hours are spent on phases 'a' and 'b,' taken together with an average of 10 percent.

Now a new project comes up, the wizbang application integration project. It has the same flavor as the seven projects in the above-mentioned study, and three engineers spent 100 man hours on the requirements and specifications phase.

How long will it take them to complete entire project? Somewhere between 667 hours and 1,667 hours with 1,000 hours being the most likely number. Why? Because six percent of 1,667 is 100, 15 percent of 667 is 100, and 10 percent of 1,000 is 100. In other words, you can predict overall project length from the time spent in the earliest phases.

Estimates of function point counts, lines of code, and number of connections to other IT systems can also be used as metrics to corroborate the estimate produced from timesheet data. For example, if you estimate this project at 10,000 lines of code and your IT shop produces 10 lines per hour on average, then this is a 1,000 hour project. Does that estimate agree with the timesheet estimate or not?

What are the benefits of having accurate estimates?

1. Over commitment is reduced. Instead of completing half of two projects, you don't even start one of them until you finish the other.
2. Your credibility as a manager increases with both employees and executives.
3. Your employees won't suffer from burnout.
4. You save your company money.

Don't underestimate how much money it saves to never start projects that don't get finished. If your IT shop has 100 engineers and they each cost the company \$100,000 per year, that's a 10,000,000 dollar annual budget. For many companies, that number is doubled. When IBM instituted the RS-Plan project in AIX development in the 1990's to reduce the start of 'unfinishable' projects, they found that 30 percent of projects they were starting required resources that they didn't have. Once they worked out a system to *just not start them in the first place*, they saved more than \$90 million...every year.

Look around, and you'll probably find you've got a similar kind of waste going on in your company. Tracking employee time is the first step to fixing it.

Curt Finch is the CEO of Journyx (<http://pr.journyx.com>), a provider of Web-based software located in Austin, Texas, that automates billing, payroll & project management by tracking time, expenses and mileage. Finch is a software industry veteran. In 1997, Curt created the world's first Internet-based timesheet application and the foundation for the current Journyx product offering. Curt has managed development teams creating enterprise-level software solutions since 1985, with a focus on distributed workforce management.