Leading SoftwAre MaNiAcs™

Have You Forgotten About Cost Management?

Date: Jan 31, 2012

Title: Have You Forgotten About Cost Management?

Code: AR-COS

Abstract: You're in hell, knee-deep managing risk and schedules on your project and even though you're tracking the team's progress and knocking down work visualized in the burndown chart ever so slowly, you may have forgotten to consider the impact your project has had your original project cost plan.

The cry "I'm leading the team to deliver, what more do THEY want?" sure seems appropriate right about now. This article highlights a few, simple best practices that if introduced at the beginning of your project might help you easily control costs along the way.

Version: 1.0 (2012-01-31) Original

Value Engineering Comes First

Everyone has probably heard about ***earned value*** and has been exposed to cost and expense calculations that are used to evaluate if your project is on track. Needless to say, budgets and cost controls are key responsibilities for your role as project manager or ScrumMaster. In fact, some management only wants to see project progress communicated only in terms of cost.

Value engineering refers to getting more out of a project without sacrificing scope. Notable examples include increasing the bottom line (profits), decreasing costs, shortening the schedule, and hopefully improving quality while you're at it. If costs spiral out of control, the value may be diminished.

Types and Usage of Project Costs

Speaking of costs, there are two types of project costs you should be primarily concerned about:

Cost Type	Description	Example
Direct	Exclusively for your project	Customized UI code development
Indirect	Shared among projects	Enhancing server system throughput

Direct and indirect costs can also be characterized based on usage:

Cost Usage	Description	Example
Fixed	Pay for it once, use as much as you want	Fixed price contract to develop product logo to be used for your product being developed.
Variable	Pay only when you use it	Contract testing effort for as long as it takes for the project to be completed

Fixed indirect costs might include office rent, since the costs are shared and the price is fixed regardless of what project you and your team are working on. (No wonder some companies, like Real Software, are close to 100% virtual.) An example of ***direct variable cost*** could be the cost of goods (COG) associated with producing DVDs for demonstration software. The *more* you ship, the *more* the cost and, of course, hopefully the *more* the return!

Surprisingly, finance will typically consider full-time staff as a variable cost, although most of us think of staff as a fixed cost. It is a good idea to quantify your estimated costs based on cost type and usage.

What Does PMI Say About This?

PMI's *PMBOK® Guide* has plenty to say about cost management in its cost management knowledge area. In fact, with only three well-defined processes you'd think cost management would be fairly straight forward:

- Estimate costs process
- Determine budget process
- Control costs process

But these process definitions are far from easy to understand. They highly overlap and quite possibly require updates as the project unfolds (exactly in keeping with an Agile approach to iterative development). These processes include a huge assortment of inputs, outputs, tools, and techniques with, in my opinion, the ***cost management plan*** being the "must have" deliverable of the bunch.

Basic Cost Estimating Techniques

There are three possible cost techniques that needs to be identified in your cost management plan:

1. **Analogous estimating:** As its name implies, this technique takes the actual costs of a previous, similar project and applies it to your current project. This approach can provide a realistic result, but it is probably the least exact of all cost-estimating techniques since projects may appear to be related, but rarely are.

2. **Parametric estimating:** This uses a combination of historical data and other statistical models to provide a cost estimate. Remember the old "lines of source code" thing? Well, "guessing" the total lines of code is a unique form of parametric estimating for the software industry. There are some pretty big "gotchas" with this approach. To come up with a qualified model, however, you'll need to factor the skill level of the developer which needs to factor the10X multiplier that states the best engineers may be as much as ten times more effective than another engineer. So, who you assign to do the work has dramatic impact to the level of effort and, consequently, the cost. Quality is yet another issue among others. If you can consistently come up with a way to estimate that works, this approach may be satisfying to your engineering sense.

3. **Bottom-up estimate:** Taking each component of work and analyzing the level of effort can be used to derive an aggregated project cost. This can take quite a long time, but if your team has experience working well together on other projects, this might be the best approach. One slight variation that I've employed comes from lean-Agile where

level of effort and cost is based on assigning a shirt size to the work. Sizes I use are: small (1-2 days), medium (about a week or two), large (over two weeks), and extra large (over a month). This approach can be a great team building role play and is actually quite fun once everyone "gets the hang of it."

Some Rules of Thumb

By far, there's a couple of cost management "killers" we all need to keep in mind while leading a project to completion.

- Controlling ***scope creep*** is by far the best way to control cost creep. How do you handle the situation when the product owner requests unplanned features midway through a project life cycle? The answer I like to use is, "Our dog house can hold five dogs (features), a new dog (new feature) needs to go in, a dog (existing feature) goes out." Works every time (and upsets the product owner every time)!

- If project work takes longer than expected to complete (hey, it happens!), in order to control costs your team must focus on the most critical feature development that matters most to the Customer (yes, customer with a capital "C"). If you need to release even without all of the features implemented, at least you prioritized completed features that represent those "must have" items that users expect to use.

A Final Word

Covered in another article "Scheduling Agile Delivery Dates," it can be extremely difficult to provide reasonable and believable cost estimates in your cost management plan. I'd seriously propose using the ***Rough Order of Magnitude*** (***ROM***) technique based on a minimum to maximum schedule range. This will give your team the leeway they need to feel like they have a pragmatic schedule to deliver by and it should enable your business partners to plan effectively.

Bibliography

Project Management Institute, Inc. *A Guide to the Project Management Body of Knowledge: PMBOK® Guide, 4th Edition*. Newton Square, PA: Project Management Institute, 2008.

Whitaker, Ken. “The Agile Schedule.” (http://www.gantthead.com/content/articles/253920.cfm).

Whitaker, Ken. *Principles of Software Development Leadership: Applying Project Management Principles to Agile Software Development*. Boston: Course Technology PTR, 2009.

Bio

Ken Whitaker of Leading Software Maniacs™ (LSM) has more than twenty-five years of software development executive leadership and training experience in a variety of technology roles and industries. He has led commercial software teams at Software Publishing (remember Harvard Graphics?), Data General, embedded systems software companies, and enterprise software suppliers. Ken is an active PMI® member, Project Management Professional (PMP)® certified, and a Certified ScrumMaster (CSM). Sources for LSM’s presentations come from case studies, personal leadership experience, the PMI *Project Management Book of Knowledge* (*PMBOK® Guide*), and Ken’s leadership books: *Managing Software Maniacs*, *Principles of Software Development Leadership*, and *I’m Not God, I’m Just a Project Manager*. He’s also the creator of PM Chalkboard, www.pmchalkboard.com, the fastest way to learn basic project management principles with entertaining tutorial videos.

Leading Software Maniacs is proudly associated with:

PMI Registered Education Provider
Project Management Institute