



 Print Article  Close Window

From: [www.cio.com](http://www.cio.com)

## Kicking Waterfall Project Management Habits Requires Agility

– David Taber, CIO

May 11, 2012

In last week's article, we covered the stuff of zombies, vampires and [antidotes for bid-to-win behaviors](#) that can suck the life out of cloud projects—you know, the sort of images that are the standard fare of CIO articles. This time, we're looking at one of the underlying causes of bid-to-win: the classic [waterfall project management](#) habits that come straight from, you guessed it, the construction industry.

When dealing with hardware and physical architecture, it's relatively obvious what the sequence of activities will be, where you are in the project and where the variances will be. As a result, a pre-determined schedule with [Program Evaluation and Review Technique \(PERT\) milestones](#) is good for both the client and the suppliers. Of course there will be contingencies, but you can make logistical decisions months or even quarters in advance of the move-in date. Typically, schedule and budget over-runs are less than 20 percent and can be handled with simple incentives and penalties.

Waterfall project management also works fine in computer and network hardware design projects, thanks to design automation and simulation tools that support fairly deterministic results. In a distributed asynchronous system such as cloud software, not so much. The bigger the project, the higher the range of over-runs—sometimes beyond 100 percent. As [Weinberg's second law](#) puts it, "If architects built buildings the way programmers build software, the first woodpecker to come along would destroy civilization."

### Two Major Challenges to Waterfall Project Management

There are two key reasons for [Big Bang](#) deployments, when there's a decisive go-live date at which point all users cut over to the new system all at once.

The first reason is that management likes to check off the box, declaring victory on that expensive project authorized all those months ago. This is just a bad habit that you really need to push back on, using the following arguments:

- Will a slash-cut really yield the best business value?
- Will it be the lowest risk approach?

- Will users—both employees and customers—be able to make the transition immediately?
- Will the data be ready all at once, and at the same time as the system?
- Will outside systems' integrations be ready to make the transition at the same time, or will some of them need to hang back for a while?

If the only reason for a Big Bang deployment is "management wants it that way," drive a stake in the heart of that monster as early in the project definition cycle as you can.

The second reason for sticking to waterfall project management is more of a challenge because it's real—a new system replacing an existing one. There are solid business reasons for not wanting to have two systems running in parallel, and to prevent a [Tower of Babel in the data sets](#). Sometimes, for logistical and customer reasons, there is no realistic alternative to a singular cutover.

But the likelihood of this transition actually succeeding in one weekend or one month just isn't that great. There are too many examples of hits to the stock price, and even shotgun-wedding corporate mergers, which were triggered by large systems projects that really couldn't go live all at once, even though they had every corporate plan to do so.

### The Antidote is Building in Waves

Using a metaphor from construction, it's safer to rebuild the house one room at a time. It's only as you open up a wall that you discover the dry rot and bad electrical connections that will mess up your plans. The goal is to discover the problems as early as possible, following the [fail fast](#) model to minimize cost and schedule impacts.

For system additions and integrations, the steps are relatively straightforward.

- Break the large project down into modular sections.
- Identify the areas of stability and the areas of change. Design the changed items to be as autonomous, transparent and safe as possible.
- Incrementally add new features and additions in sprints, testing for stability at every turn.
- Expose the user features last, after all infrastructure, integrations and data have been validated.
- Deliver small stuff fast and predictably to gain the trust of the organization.

Things get harder when you have either a replacement system or a merger of two running system into this "new world for users." What we recommend is staging the cutover to minimize the risk of business interruption.

- Keep the existing system(s) running as the system(s) of record throughout the construction and validation of the new system.
- Involve a pilot team of users throughout the construction project. If possible, these users should have a clean subset of the data they work with—for example, a geographic sales territory or a well-defined range of accounts that only they "own" and work with.
- Bring these *avant garde* users into pilot usage of the system toward the end of the validation phase. During this phase, they need to be using both the old system and the new.
- Only after they have validated the new system's functionality and data do you expand the deployment to other teams. Tightly manage the definition and sequencing of these other teams, and keep [squeaky wheel politics](#) at bay as much as you can. Be willing to slow down the deployment to new groups if you hit problems; by inching along, you can typically avoid back-tracking.
- Use the pilot users' success to bolster the image of the project with the groups that aren't on board yet.
- Following this "wave" approach to deployment to reduce risk, the last group of users should move to the new system months after the pilot go-live.

To support this, you'll need robust data synchronization and configuration-control infrastructure that spans the old systems and the new one. Lowering business risk does come at a cost, but it increases the likelihood of good project ROI.

But you'll need one more thing to make this work— which next week's piece will cover.

*David Taber is the author of the new Prentice Hall book, "[Salesforce.com Secrets of Success](#)" and is the CEO of [SalesLogistix](#), a certified Salesforce.com consultancy focused on business process*

*improvement through use of CRM systems. SalesLogistix clients are in North America, Europe, Israel and India, and David has over 25 years of experience in high tech, including 10 years at the VP level or above.*

*Follow everything from CIO.com on Twitter [@CIOonline](#), on [Facebook](#) and on [Google +](#).*

© 2012 CXO Media Inc.