



 Print Article  Close Window

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

## How to Keep Cloud Projects Agile, Simple

– David Taber, CIO

**June 04, 2012**

In the previous articles in this series, we've examined persistent problems that can kill Agile projects. You don't want your next cloud project to be wrought with [too many specifications](#), managed as a [Big Bang, slash-and-burn deployment](#), or procured with [bid-to-win](#) behaviors.

Now it's time to work on an issue that underlies all of those problematic behaviors: saying "Yes" too often.



Saying "No" will keep a project on track and your credibility intact. (Image courtesy of [Zazzle.com](http://Zazzle.com).)

Of course this is heresy. User-centered design is at the core of good UI. It would be political suicide for IT to say "No" to powerful users. IT engineering exists to serve the business, not order it around.

All those statements are true. But so is this: Saying "Yes" when the answer really needs to be "No" is an invitation to a project failure. Improper expectations don't just hurt budgets and schedules. They

damage credibility—and, in the end, they force you to expend resources when you shouldn't have to.

## Saying "No" Part of a Painful Conversation

Unfortunately, the longer the project takes, the greater the likelihood of the scope creep that is the bane of [project management](#). It's hard to know which form of scope creep is worse—explicit (adding features to compensate for being late) or implicit (when users cry, "But I thought the system would just work that way").

Of course, you can try to resolve this by having an air-tight spec. Have fun with that—it's rarely effective, and it usually mires you in microscopic thinking that guarantees you'll miss something important.

It's no accident that [Agile](#) advocates short projects and minimalist specifications. The goal is to have the team with a small number of deeply motivated and detail-knowledgeable people moving as fast as possible down the field. It's also no accident that an Agile increment is called a [scrum](#), from the root of the word *scrimmage*.

There's no market for bad news, so nobody wants to be the person saying no all the time. Let's take it one step further—in many cases, if you clearly say "No," you're asking for retaliation, often in the form of an even more firmly stated requirement. That's just how people act.

Keeping that in mind, here are three conversational tactics for saying "No" that I have seen succeed.

### 1. The Bureaucratic Dodge.

OK, this doesn't sound good. But it's not dishonest, and it's actually easy for some executives—note the immortal title of Seth Godin's book, [All Marketers are Liars](#).

This approach can be very effective. You never say "Yes," and you always leave doubt that Yes is even possible. In a meeting, when some stakeholder asks for a feature that isn't part of the optimal plan, you respond with phrases like the following:

- "We can talk about that."
- "That's a great idea."
- "We didn't know about that one."
- "Let's look into it."
- "We'll get back to you on that."
- "That's on the prioritization list as of now."
- "We'll have to see when that can be scheduled in."

The critical feature of this approach is to not commit to anything and move on to other topics within microseconds. The longer a feature area is discussed in a meeting, the more people will start to think the answer is "Yes." Of course, this leaves you vulnerable to [happy ears](#), in which the absence of "No" is optimistically converted to an affirmative.

The remedy is to propagate the "No" after the meeting is over, working with organizations in private discussions rather than a confrontational meeting environment.

### 2. The Resource Gambit.

Part of the problem with a project agenda covered with the word "Yes" is economics. Nobody thinks about the real cost of his requests, so he over-consume resources that he believe to be free.

While you certainly don't want to come across as a bean-counter, you should still apply the power of their logic. These five questions will help.

- "Can you commit one of your people to working on the project team?"
- "Have you developed a set of success criteria to validate when the feature is complete?"
- "Do we know what the payoff of this feature is?" Use words that are appropriate for your organization—cost/benefit, ROI, [IRR](#), [hurdle rate](#), etc.
- "If we put this feature in, how much more will you be able to sell?" That's for a sales or marketing request. For other departments use more generic words such as "produce" or

"deliver."

- "If we can do this, where will the savings be?"

Try to enlist the CFO's help in these gambits. He or she can then ask the bare-fisted versions of this line of questioning:

- "Whose budget is this going to come out of?"
- "How much are you going to increase your sales quota to cover this?"
- "Where is the headcount reduction enabled by this increased investment in automation?"

### 3. Speaking to Users in Their Own Language.

IT people tend to be analytical and precise. Different areas of the business have very different thought patterns and communication styles, so effective persuasion means more than just different vocabulary.

Roger Fisher and William Ury's famous book tells about [Getting to Yes](#) in a negotiation. Here's how to improve your chances of "Getting to" No with your colleagues in other departments:

Members of the **sales staff** tend to work every problem as a negotiation. Since every decision for them is an exchange of value, frame the discussion in terms of what they'll give as well as what they'll get. Focus the conversation on incremental revenue (what they're measured on), not incremental profits or ROI (which they're likely deaf to).

**Engineering, manufacturing and operations** tend to work every problem as something to be solved. Since their decisions are all about optimization, frame the discussion in terms of efficiency and business processes. Most talk will be about costs, but when revenue is involved, focus the conversation on the profitability of incremental revenue.

**Customer support**, customer service, professional services and training all tend to think in terms of cost reduction and streamlining the time to execute. Since their metrics focus more on customer interactions than on revenue, frame the discussion around customer satisfaction.

Finally, the **CFO** wears twin hats—cost reduction and improvement of the return on capital invested. CFOs aren't blind to upside. They just don't believe it until they see it. With the CFO, focus the discussion on efficiency of capital deployed, reduction in fixed costs and payback interval, no matter the size of the investment.

Learning to say "No" is difficult, and it may not make you any friends during the software development lifecycle. On the other hand, when you deliver an easy-to-use cloud application on time, on budget and consistent with your corporate mission, you may find your popularity skyrocketing after all.

*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. 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.