

3. Saying Yes



Did you know that I invented voice mail? It's true. Actually there were three of us who held the patent for voice mail. Ken Finder, Jerry Fitzpatrick, and I. It was in the very early 80s, and we worked for a company named Teradyne. Our CEO had commissioned us to come up with a new kind of product, and we invented "The Electronic Receptionist," or ER for short.

You all know what ER is. ER is one of those horrible machines that answers the phone at companies and asks you all kinds of brain-dead questions that you need to answer by pressing buttons. ("For English, press 1.")

Our ER would answer the phone for a company and ask you to dial the name of the person you wanted. It would ask you to pronounce your name, and then it would call the person in question. It would announce the call and ask whether it should be accepted. If so, it would connect the call and drop off.

You could tell ER where you were. You could give it several phone numbers to try. So if you were in someone else's office, ER could find you. If you were at home, ER could find you. If you were in a different city, ER

could find you. And, in the end, if ER could not find you, it would take a message. That's where the voice mail came in.

Oddly enough, Teradyne could not figure out how to sell ER. The project ran out of budget and was morphed into something we knew how to sell—CDS, The Craft Dispatch System, for dispatching telephone repairmen to their next job. And Teradyne also dropped the patent without telling us. (!) The current patent holder filed three months after we did. (!!)¹

Long after the morphing of ER into CDS, but long before I found out that the patent had been dropped. I waited in a tree for the CEO of the company. We had a big oak tree outside the front of the building. I climbed it and waited for his Jaguar to pull in. I met him at the door and asked for a few minutes. He obliged.

I told him we really needed to start up the ER project again. I told him I was sure it could make money. He surprised me by saying, “OK Bob, work up a plan. Show me how I can make money. If you do, and I believe it, I'll start up ER again.”

I hadn't expected that. I had expected him to say, “You're right Bob. I'm going to start that project up again, and I'm going to figure out how to make money at it.” But no. He put the burden back on me. And it was a burden I was ambivalent about. After all, I was a software guy, not a money guy. I wanted to work on the ER project, not be responsible for profit and loss. But I didn't want to show my ambivalence. So I thanked him and left his office with these words:

“Thanks Russ. I'm committed . . . I guess.”

With that, let me introduce you to Roy Osherove, who will tell you just how pathetic that statement was.

A Language of Commitment

By Roy Osherove

Say. Mean. Do.

There are three parts to making a commitment.

1. You *say* you'll do it.
2. You *mean* it.
3. You *actually do* it.

But how often do we encounter other people (not ourselves, of course!) who never go all the way with these three stages?

- **You ask the IT guy** why the network is so slow and he says “Yeah. We really need to get some new routers.” And you *know* nothing will ever happen in that category.
- **You ask a team member** to run some manual tests before checking in the source code, and he replies, “Sure. I hope to get to it by the end of the day.” And somehow you *feel* that you'll need to ask tomorrow if any testing really took place before check-in.
- **Your boss** wanders into the room and mumbles, “we have to move faster.” And you *know* he really means YOU have to move faster. *He's* not going to do anything about it.

There are very few people who, when they say something, they mean it and then actually get it done. There are some who will say things and *mean* them, but they never get it done. And there are far more people who promise things and don't even *mean* to do them. Ever heard someone say, “Man, I really need to lose some weight,” and you knew they are not going to do anything about it? It happens all the time.

Why do we keep getting that strange feeling that, most of the time, people aren't really committed to getting something done?

Worse, often our intuition can fail us. Sometimes we'd *like* to believe someone really means what they say when they really don't. We'd *like* to believe a developer when they say, pressed to the corner, that they can finish that two-week task in one week instead, but we shouldn't.

Instead of trusting our guts, we can use some language-related tricks to try and figure out if people really mean what they say. And by changing what we say, we can start taking care of steps 1 and 2 of the previous list on our own. When we *say* we will commit to something, and we need to *mean* it.

Recognizing Lack of Commitment

We should look at the language we use when we *commit* to doing something, as the tell-tale sign of things to come. Actually, it's more a matter of looking for specific *words* in what we say. If you can't find those little magic words, chances are we don't mean what we say, or we may not believe it to be feasible.

Here are some examples of words and phrases to look for that are telltale signs of noncommitment:

- **Need\should.** “We need to get this done.” “I need to lose weight.” “Someone should make that happen.”
- **Hope\wish.** “I hope to get this done by tomorrow.” “I hope we can meet again some day.” “I wish I had time for that.” “I wish this computer was faster.”
- **Let's.** (not followed by “I . . .”) “Let's meet sometime.” “Let's finish this thing.”

As you start to look for these words you'll see that you start spotting them almost everywhere around you, and even in things you say to others.

You'll find we tend to be very busy not taking responsibility for things.

And that's *not* okay when you or someone else relies on those promises as part of the job. You've taken the first step, though—start recognizing lack of commitment around you, and in you.

We heard what noncommitment sounds like. How do we recognize real commitment?

What Does Commitment Sound Like?

What's common in the phrases of the previous section is that they either assume things are out of “my” hands or they don't take personal responsibility. In each of these cases, people behave as if they were *victims* of a situation instead of in control of it.

The real truth is that *you, personally*, ALWAYS have something that's under *your* control, so there is always *something* you can fully commit to doing.

The secret ingredient to recognizing real commitment is to look for sentences that sound like this: I will . . . by . . . (example: I will finish this by Tuesday.)

What's important about this sentence? *You're stating a fact about something YOU will do with a clear end time.* You're *not* talking about anyone else but yourself. You're talking about an *action* that you *will* take. You won't "*possibly*" take it, or "*might get to it*"; you *will* achieve it.

There is (technically) no way out of this verbal commitment. You said you'll do it and now only a binary result is possible—you either get it done, or you don't. If you don't get it done, people can hold you up to your promises. You will feel *bad* about not doing it. You will feel *awkward* telling someone about not having done it (if that someone heard you promise you will).

Scary, isn't it?

You're taking full responsibility for something, in front of an audience of at least one person. It's not just you standing in front of the mirror, or the computer screen. It's you, facing another human being, and saying you'll do it. That's the start of commitment. Putting yourself in the situation that forces you to do something.

You've changed the language you use to a language of commitment, and that will help you get through the next two stages: meaning it, and following through.

Here are a number of reasons you might not *mean* it, or follow through, with some solutions.

It wouldn't work because I rely on person X to get this done

You can only commit to things that you have *full control* of. For example, if your goal is to finish a module that also depends on another team, you can't commit to finish the module with full integration with the other team. But you *can* commit to specific actions that will bring you to your target. You could:

- Sit down for an hour with Gary from the infrastructure team to understand your dependencies.
- Create an interface that abstracts your module's dependency from the other team's infrastructure.
- Meet at least three times this week with the build guy to make sure your changes work well in the company's build system.

- Create your own personal build that runs your integration tests for the module.

See the difference?

If the end goal depends on someone else, you should commit to specific actions that bring you closer to the end goal.

It wouldn't work because I don't really know if it can be done

If it can't be done, you can still commit to actions that will bring you closer to the target. Finding out if it can be done can be one of the actions to commit to!

Instead of committing to fix all 25 remaining bugs before the release (which may not be possible), you can commit to these specific actions that bring you closer to that goal:

- Go through all 25 bugs and try to recreate them.
- Sit down with the QA who found each bug to see a repro of that bug.
- Spend all the time you have this week trying to fix each bug.

It wouldn't work because sometimes I just won't make it

That happens. Something unexpected might happen, and that's life. But you still want to live up to expectations. In that case, it's time to change the expectations, *as soon as possible*.

If you can't make your commitment, the most important thing is to raise a red flag as soon as possible to whoever you committed to.

The earlier you raise the flag to all stakeholders, the more likely there will be time for the team to stop, reassess the current actions being taken, and decide if something can be done or changed (in terms of priorities, for example). By doing this, your commitment can still be fulfilled, or you can change to a different commitment.

Some examples are:

- If you set a meeting for noon at a cafe downtown with a colleague and you get stuck in traffic, you doubt you'll be able to follow through on your commitment to be there on time. You can call your colleague as soon as you realize you might be late,

and let them know. Maybe you can find a closer place to meet, or perhaps postpone the meeting.

- If you committed to solving a bug you thought was solvable and you realize at some point the bug is much more hideous than previously thought, you can raise the flag. The team can then decide on a course of action to make that commitment (pairing, spiking on potential solutions, brainstorming) or change the priority and move you over to another simpler bug.

One important point here is: If you don't tell anyone about the potential problem as soon as possible, you're not giving anyone a chance to help you follow through on your commitment.

Summary

Creating a language of commitment may sound a bit scary, but it can help solve many of the communication problems programmers face today—estimations, deadlines, and face-to-face communication mishaps. You'll be taken as a serious developer who lives up to their word, and that's one of the best things you can hope for in our industry.

Learning How to Say “Yes”

I asked Roy to contribute that article because it struck a chord within me. I've been preaching about learning how to say no for some time. But it is just as important to learn how to say yes.

The Other Side of “Try”

Let's imagine that Peter is responsible for some modifications to the rating engine. He's privately estimated that these modifications will take him five or six days. He also thinks that writing the documentation for the modifications will take a few hours. On Monday morning his manager, Marge, asks him for status.

Marge: “Peter, will you have the rating engine mods done by Friday?”

Peter: “I think that's doable.”

Marge: “Will that include the documentation?”

Peter: “I'll try to get that done as well.”

Perhaps Marge can't hear the dithering in Peter's statements, but he's certainly not making much of a commitment. Marge is asking questions that demand boolean answers but Peter's boolean responses are fuzzy.

Notice the abuse of the word try. In the last chapter we used the "extra effort" definition of try. Here, Peter is using the "maybe, maybe not" definition.

Peter would be better off responding like this:

Marge: "Peter, will you have the rating engine mods done by Friday?"

Peter: "Probably, but it might be Monday."

Marge: "Will that include the documentation?"

Peter: "The documentation will take me another few hours, so Monday is possible, but it might be as late as Tuesday."

In this case Peter's language is more honest. He is describing his own uncertainty to Marge. Marge may be able to deal with that uncertainty. On the other hand, she might not.

Committing with Discipline

Marge: "Peter, I need a definite yes or no. Will you have the rating engine finished and documented by Friday?"

This is a perfectly fair question for Marge to ask. She's got a schedule to maintain, and she needs a binary answer about Friday. How should Peter respond?

Peter: "In that case, Marge, I'll have to say no. The soonest I can be *sure* that I'll be done with the mods and the docs is Tuesday."

Marge: "You are committing to Tuesday?"

Peter: "Yes, I will have it all ready on Tuesday."

But what if Marge really needs the modifications and documentation done by Friday?

Marge: "Peter, Tuesday gives me a real problem. Willy, our tech writer, will be available on Monday. He's got five days to finish up the user guide. If I don't have the rating engine docs by

Monday morning, he'll never get the manual done on time. Can you do the docs first?"

Peter: "No, the mods have to come first, because we generate the docs from the output of the test runs."

Marge: "Well, isn't there some way you can finish up the mods and the docs before Monday morning?"

Now Peter has a decision to make. There is a good chance he'll be done with the rate engine modifications on Friday, and he might even be able to finish up the docs before he goes home for the weekend. He *could* do a few hours of work on Saturday too if things take longer than he hopes. So what should he tell Marge?

Peter: "Look Marge, there's a good chance that I can get everything done by Monday morning if I put in a few extra hours on Saturday."

Does that solve Marge's problem? No, it simply changes the odds, and that's what Peter needs to tell her.

Marge: "Can I count on Monday morning then?"

Peter: "Probably, but not definitely."

That might not be good enough for Marge.

Marge: "Look, Peter, I really need a definite on this. Is there *any* way you can commit to get this done before Monday morning?"

Peter might be tempted to break discipline at this point. He might be able to get done faster if he doesn't write his tests. He might be able to get done faster if he doesn't refactor. He might be able to get done faster if he doesn't run the full regression suite.

This is where the professional draws the line. First of all, Peter is just wrong about his suppositions. He *won't* get done faster if he doesn't write his tests. He *won't* get done faster if he doesn't refactor. He *won't* get done faster if he omits the full regression suite. Years of experience have taught us that breaking disciplines only slows us down.

But secondly, as a professional he has a responsibility to maintain certain standards. His code needs to be tested, and needs to have tests. His

code needs to be clean. And he has to be sure he hasn't broken anything else in the system.

Peter, as a professional, has already made a commitment to maintain these standards. All other commitments he makes should be subordinate to that. So this whole line of reasoning needs to be aborted.

Peter: "No, Marge, there's really no way I can be certain about any date before Tuesday. I'm sorry if that messes up your schedule, but it's just the reality we're faced with."

Marge: "Damn. I was really counting on bringing this one in sooner. You're sure?"

Peter: "I'm sure that it might be as late as Tuesday, yes."

Marge: "OK, I guess I'll go talk to Willy to see if he can rearrange his schedule."

In this case Marge accepted Peter's answer and started hunting for other options. But what if all Marge's options have been exhausted? What if Peter were the last hope?

Marge: "Peter, look, I know this is a huge imposition, but I really need you to find a way to get this all done by Monday morning. It's really critical. Isn't there something you can do?"

So now Peter starts to think about working some significant overtime, and probably most of the weekend. He needs to be very honest with himself about his stamina and reserves. It's easy to *say* you'll get a lot done on the weekends, it's a lot harder to actually muster enough energy to do high-quality work.

Professionals know their limits. They know how much overtime they can effectively apply, and they know what the cost will be.

In this case Peter feels pretty confident that a few extra hours during the week and some time on the weekend will be sufficient.

Peter: "OK, Marge, I'll tell you what. I'll call home and clear some overtime with my family. If they are OK with it, then I'll get this task done by Monday morning. I'll even come in on Monday morning to make sure everything goes smoothly with Willy. But then I'll go home and won't be back until Wednesday. Deal?"

This is perfectly fair. Peter knows that he can get the modifications and documents done if he works the overtime. He also knows he'll be useless for a couple of days after that.

Conclusion

Professionals are not required to say yes to everything that is asked of them. However, they should work hard to find creative ways to make “yes” possible. When professionals say yes, they use the language of commitment so that there is no doubt about what they've promised.