EXPERT SUPPORT
Understanding technology and writing about it since 1990

Schedule Chicken: Knowing When to Commit to the Impossible

Douglas C. Shaker

Remember the movie “Rebel Without a Cause”? Two teenagers in stolen cars drive toward a cliff. Both intend to jump out of the cars before they go off the cliff, but the first to do so is “chicken.”

This kind of game is frequently played with project deadlines. The project team signs up for an impossible schedule, a schedule that seems to make the Marketing department or management happier than a realistic schedule.

Everyone is sure someone else is going to bail on the schedule first. When a group misses the big deadline, the other parties pretend they could have delivered on time. “Yeah, Software Development really blew it this time. We would have been ready for the launch, but now we have to wait for the software.” And that’s how the game is played.

Luckily, technical writers have an advantage in this game. As hard as it is to write about a tessellated widget, it’s always easier than it is to design it, create it, and get it to work properly.

As a documentation manager, you can start playing this game when the product manager asks you to produce 150 pages of product documentation in four weeks. A normal heuristic for estimating the time to write a technical document is about 20 pages a week. If the doc is for end users, isn’t very technical, and there’s a really fast writer on the project, that might go up to 30 pages a week. If it’s a very technical document like an API, the estimate drops to 10 pages a week. So with four weeks and one writer, you know your writer can get about 80 pages done, and that’s only if everyone is organized and engineering has an articulate and accurate technical expert at the writer’s disposal.

Realistically, it’s not possible to do this in four weeks. Do you decline the job? Of course not! Rev your engines...it’s time to play Schedule Chicken!

Before committing to the impossible schedule, ask some questions to make sure you aren’t getting into serious trouble:

- Are you planning a real Beta test? (The answer should be “Yes.”)
- Is the software in Beta yet? (It should either be “No” or there should be multiple Beta tests with one or more to go.)
- How much of the software is implemented and working? (You can feel comfortable if it is 50% or less.)
- Are there up-to-date design documents available? (“No” usually means engineering is still messing with the design.)

If it’s not in Beta and it’s not even working, you know this product isn’t going out the door in four weeks. If the software isn’t working, then the Beta is at least two weeks away and more likely four. The Beta will take at least four more weeks, since the software to be installed, put in use, and meaningfully tested. Add another two weeks to fix the bugs and two more to engineer the final release. That’s 10-12 weeks to get the software done, plenty of time to write those 150 pages.

You usually can’t tell the product team what you think their real schedule is, as that’ll just put them on the defensive. However, you can say, “We’re sure our writers can keep up with your engineering team and we’ll have documents ready for your release.” Voila! You are now playing Schedule Chicken.

The only thing left to do is to make sure you can avoid The Beta Trap. The Beta Trap occurs when you’ve committed to deliver docs for the official customer release, but suddenly find out you also have to write a complete set of docs for Beta.

Getting documents done for Beta is always tough. Why? Beta can occur any time after the software starts working, but way before it does anything correctly. You might only have a week or two from the time the software starts to work before it gets shipped out the door to the first Beta customers. At best, the documentation at this point is mostly fiction and woefully incomplete (a “colander version” you might say, because it has so many holes). It might be pretty good fiction, but there certainly hasn’t been time to test it or take screenshots; and at least 30% of it is going to change by the general release.

Luckily, the first Beta customers are usually experienced, long-time users who requested the new features in the first place. They might only need installation instructions and an engineer’s email address to test what they need. So, typically you don’t have to sweat the Beta, as long as everyone’s expectations are properly set.

You can set those expectations and avoid the Beta Trap by saying something like “Well, with a schedule this aggressive, we’ll have to stage the Beta documentation. We can do installation docs and a conceptual overview first, and add more comprehensive documentation as we get it done.” And, having no reasonable alternative in a universe without time travel, the team usually signs on. So does the VP of Engineering and the VP of Marketing.

That’s Schedule Chicken. If you are a technology professional, you’ve probably already played this game. And now you know what to call it.

Douglas C. Shaker was Expert Support’s Vice President emeritus.

This article is part of a series of articles on technical writing that Expert Support hosts on its website, expertsupport.com.
Expert Support is located in Los Altos, California and supplies contract technical writers to the computer and software industry.