The Most Important Points Nobody Told You Before You Started Building That App

Tomer Dicturel Follow

Jan 2 · 4 min read ★

For nearly 50 years—ever since Frederick Brooks published the classic "The Mythical Man-Month"—software development teams have struggled with how to build a project on time and according to spec. It's no easy task. Here's what they are forgetting to tell you before you started building that new app…

The final product will look nothing like the original specifications

Building an app should be simple enough. You sit down a few people in a room, agree on a few specifications, and then let the smartest people in the room go to work coding what you just finished discussing. Easy enough, right? Wrong. There is a very high probability that the final product will look nothing like the original specifications. There are a

number of very good reasons for this, and it has nothing to do with the "incompetence" of the software development team. Deadlines change. Plans change. In some cases, even the original problem you were trying to solve changes. In fact, it's very much a miracle anything actually gets built in the end.

The more stakeholders you have on a project, the messier the final result

On the surface, this would seem to make perfect sense to limit the number of chefs in the kitchen, yet you'd be surprised at just how many perfectly sensible people ignore this. Instead, there's a rush to involve not just the development team, but also the sales team, the marketing team, and maybe even the guy down the hall who has a funky, made-up title on his business card. And what happens next is like the old-fashioned game of telephone, where each person who hears a conversation repeats it slightly differently to the next person in the chain. According to what is now known as Brooks' Law (in honor of

Frederick Brooks), "adding manpower to a late software project makes it later."

There will always be one part of the finished product that nobody knows exactly what it does

In a best-case scenario, there will always be a direct one-to-one mapping between all the features originally drawn up by the software development team, and the final features that appear in the app or software. But here's the problem—most software development teams are under so much pressure to get the project out the door that they will skimp on the documentation of what each line of code is actually supposed to do. Repeat this enough times, and it inevitably leads to a "feature" that nobody really knows what it does, or how it even appeared in the first place. (And whatever you do, don't call it a "bug" —it's always a "feature"!)

One person on your team will be in charge of moving the goalposts

"HOW TO TORTURE A BUSINESS ANALYST # 3"

As much as people like to talk about "being in alignment" (or whatever the latest MBA 101 jargon happens to be), people are rarely in alignment. That's what makes us people, not machines. And one of those people will (unofficially, of course) appoint himself or herself as the person in charge of moving the goalposts. You know, the person who shows up at the Monday morning meeting and announces out of nowhere that the project deadline date has been moved up a few weeks, or that a long-forgotten feature is now "mission-critical" and must be added immediately.

**

Conclusion

So the next time you sit down with your team and start to hammer out the deadlines and specifications for your next software project, keep these points in mind. It might just save you a lot of blood, sweat, and tears.

Hey! I'm Tomer, an entrepreneur and maker. You might know me from Mevee, Crane, and Shots, among other products I've launched! This article is a part in a larger series I'm writing mostly based on my experiences, and is largely made of my and my team's opinions.

I hope this helps you to avoid making the same mistakes I did, and remember to keep shipping!

The Startup 🔍

This story is published in The Startup, Medium's largest entrepreneurship publication followed by +406,714 people.

Subscribe to receive our top stories here.