Sweating the Small Stuff

September 16, 2007

Original link

So Jottit has launched, only five months after I suggested to my friend Simon that we create a website that was just a big text box people could type stuff into. And there are two ways I look at it. One is: It took us five months to do *that*? And the other is: We did that in only five months?

When you look at what the site does, it seems pretty simple. It has few features, no complex algorithms, little gee-whiz gadgetry. It just takes your text and puts it on the Web. And considering how often I do that every day, it seems a bit odd that it took so long to create yet another way. And then I check the todo list.

As I've said, this is a site I wanted to get every little detail right on. And when you start sweating the small stuff, it's frankly incredible just how much of it there is. Even our trivial site is made up of over two dozen different screens. Each one of those screens has to be designed to look and work just right on a wide variety of browsers, with a wide variety of text in them.

And that's just making things look good — making them work right is much harder. Each screen does, on average, five or six different things. And each of those things can be done under three or four different modes. Now we're up to over 500 different things to do, each of which can have bugs in lots of unthought-of ways. And then, many of these pieces are exposed to users, who can do whatever they want with them — and do. If you're building a site that accepts text from users, you need to think about something that lets some people just paste stuff from emails, others write HTML, others play YouTube videos, while others try to insert malicious text to break things for people.

There are lots of features we want to add to Jottit, but before we do any of that we want to make what we have work perfectly. And, at the moment, that means tasks as varied as reporting a bug in a piece of software we use to its developer, configuring the web server to display a nicer error message under certain odd conditions, having another computer monitor the first computer to see if it goes down, figuring out how to tweak the UI to make certain unclear things clearer to people, rewriting some of the text on the site to be nicer, creating a new site to inform our users of updates, making some stuff from our project open source, fixing stuff in other open source projects, testing the site on phones and weird browsers, examining an algorithm we use to see if it needs improvement, and fixing a bug that was just reported by a user. And those are just the things on *my* todo list!

When you look at it that way, it's amazing anyone ever builds a website.