WHAT

BUSINESSWEEK JUNE 11, 2015

CODE

Author: Paul Ford

A message from Josh Tyrangiel

Cumbie's Note:

This is an excerpt (pp 59-61) of Paul Ford's magazine-long essay from Bloomberg's Businessweek, What is Code?

TMitTB refers to "The Man in the Taupe Blazer", from page 2 of the article:

For your entire working memory, some Internet thing has come along every two years and suddenly hundreds of thousands of dollars (inevitably millions) must be poured into amorphous projects with variable deadlines.

Content management projects, customer relationship management integration projects, mobile apps, paperless office things, global enterprise resource planning initiatives—no matter how tightly you clutch the purse strings, software finds a way to pry open your fingers.
Here we go again. On the other side of your (well-organized) desk sits this guy in his mid-30s with a computer in his lap. He's wearing a taupe blazer. He's come to discuss spending large sums to create intangible abstractions on a "website re-architecture project." He needs money, support for his team, new hires, external resources. It's preordained that you'll give these things to him, because the CEO signed off on the initiative—and yet should it all go pear-shaped, you will be responsible. Coders are insanely expensive, and projects that start with uncomfortably large budgets have an ugly tendency to grow from there. You need to understand where the hours will go.

He says: "We're basically at the limits with WordPress."

5

The Time You Attended the E-mail

Address Validation Meeting

In the interest of understanding more about how all this works, and with an open invitation from TMitTB, you attend a meeting of the programmers.

Two of them are late, and bravely you ask the one already in attendance to explain what's going on. He quickly gathers the limits of your information through a series of questions, beginning with, "Do you know what a Web page is?"

Here's what he shows you: To gather an e-mail address and a name, you can make a Web page using HTML. ■ 23

On today's agenda: How to make sure that registration is a positive experience for users but also a secure experience for the company. The questions to be discussed, the programmer tells you, are along the lines of, "Where will you put this data? Will you put it in a text file? What will you do with it? How will you act upon it?"

Enter the remaining two programmers. Programmer A, who is senior, takes her place at the whiteboard. ...

Programmer A: "Let's just start with e-mail validation."

Programmer B: “Can you define valid?”

Programmer C: “A well-formed—”

—well-formed via a regular expression test?

—or well-formed according to RFC 5321 and RFC 5322?

Let’s assume a library, OK?

Do we send everyone who submits a form a confirmation message?

We should, right?

Is it in the spec?

No.

Then we shouldn’t do it.

Well, wait, maybe we should update the spec.

It’s best practice.

Can I say what worries me?

Wait–capture that. We need to check on confirmations.

Got it.

What really, really worries me are “temporary” e-mail services, right? Like Mailinator.com, which allows you to give out “disposable” e-mail addresses.

So don’t let anyone enter a Mailinator address.

[Typing.]

Right, but Mailinator.com doesn’t have just one address. It has, like, 300.

So get a list.

There’s no list actually. You have to go and reload the Mailinator page and make a list and cross your fingers.

Yeah, we looked at this. There’s no list.

Can we not rathole on Mailinator before we talk overall security?

I mean, if we're counting on the library validation, is that good enough? No one is going to enter a 500-page e-mail.

Russians will.

Very possibly.

But we can set upload limits, right?

Sort of, but we also allow people to upload large images for their profiles–up to 2 megabytes.

So?

No, he's right. Those all come as a lump, so someone could enter a 500-page-long e-mail, and we need to check it.

I'll take that if you want, because it's going to relate to the overall database schema.

Can we catch it at the database?

That's why I'll take it.

Someone take a picture of the whiteboard?

Sure.

And we should talk about account expiration?

Is that in the spec?

No, but it’s a dependency for the password-expiration admin tool.

Who owns that?

And on it goes, whiteboard after whiteboard, punctuated by the sound of a mobile phone’s fake camera shutter. “Do we need to keep track of how many times they’ve been e-mailed?” “How do we remove e-mails once they’re in the system?” “What if someone enters the same e-mail twice?”

Programmer A, the leader, seems very professional. She’s at the whiteboard, scribbling, erasing, scribbling, erasing. Lists, arrows, boxes, lines. She wrote RUSSIANS? on the board. But after an hour you realize: This is just e-mail. One field. One little bit of data. You haven’t even hit names yet. What if the user has one name? What if Bono or Cher signs up for an account? What if it’s the Chinese Bono? Do we want to allow sign-ups in Chinese? What browsers do we need to support? Do the call center people need to be able to manage accounts?

It’s hard not to think of barrels of cash burning.

Programmer B is entering things into a tracking system, creating issues, assigning tasks to people. A flurry of e-mailed assignments is emerging from this meeting. Programmer C is young and annoying and very programmery, but the others seem to like him well enough.

How do we ensure that credit cards are valid, that physical addresses are real? Will we perform financial transactions ourselves? Which external systems will integrate with our systems? Who will get the sales reports? We didn’t talk about the mailing list software. We didn’t talk about password length, the number of letters and symbols necessary for passwords to be secure, or whether our password strategy on this site will fit in with the overall security profile of the company, which is the responsibility of a different division.

So this is the work. It goes on for days.

It gets turned into specifications and user stories, then reviewed with TMitTB, who right now is away at a conference (but tells you he’s overjoyed you attended this meeting).

Not a line of code is written throughout this process.

5.1 ~~What Is the Relationship Between Code and Data?~~

~~Data comes from everywhere. Sometimes it comes from third parties—Spotify imports big piles of music files from record labels. Sometimes data is user-created, like e-mails and tweets and Facebook posts and Word documents. Sometimes the machines themselves create data, as with a Fitbit exercise tracker or a Nest thermostat. When you work as a coder, you talk about data all the time. When you create websites, you need to get data out of a database and put them into a Web page. If you’re Twitter, tweets are data. If you’re the IRS, tax returns are data, broken into fields.~~

~~Data management is the problem that programming is supposed to solve. But of course now that we have computers everywhere, we keep generating more data, which requires more programming, and so forth. It’s a hell of a problem with no end in sight. This is why people in technology make so~~