

## Lessons from the (latest) internet revolution

Once more with publishing some notes that I'd left lying around. I was thinking about the new Internet – the one that's emerged over the last year and is becoming more real-time, reactive and uses more data analysis, maps and geography. And I was wondering what lessons we could learn from this shift.

### 1. Stop observing your users and start interacting with them

[Huge](#) just published a [book](#) on the shift from customers to users, i.e. from people who buy one-off purchases from a firm, and people who are engaged in a conversation with that firm, whether they're purchasers, potential purchasers, bloggers, employees or fans.

That resonates. It's a shift already happening in the commercial world, and it's a shift going on in the development world too. I've blogged before about the shift in organisational focus from vulnerability to resilience and the work of people like [CDAC](#) and the [World Bank](#) in this. In its purest form, this is just another way to say "start listening to and working with the people you're trying to help". It's not rocket science, but it does come with all sorts of culture changes, some of which (shock, horror) might just be hastened by tech.

### 2. Keep it clean

No, no, I'm not talking about the rise and fall of pornography markets here – I'm talking about data and the information it contains. If you ask many seasoned development professionals about their data collection methods, you might get quite a cynical response (see un blog). Okay, so Amazon has had a couple of [high-profile data fails](#), but on the whole commercial data is accurate and consistent because it has to be.

### 3. Make real-time internet easier

When you buy a book or a dress online, you don't have to wait for days to know what the price is or whether you've bought it (okay, with some sites you still do, but that's getting rarer now) – it comes up, the stock comes up, you buy and the thing is delivered to your house / office / significant other because you forgot their birthday.

Real-time development data (by which I mean occurs in months, days, hours rather than per year) is still, largely, manual. Teams of people are out there entering data, formatting it, cleaning it and transferring it from place to place because we don't have the systems in place to make this easy yet. Other teams are working on those systems piecemeal (and a big hand to people like [okfn](#), [data without borders](#) and [crisismappers](#) for making this possible), but there is still (to my knowledge – please correct me if I'm wrong) an easy template or data standards to allow development data to be discovered, uploaded, cleaned and transferred between systems without

pain. And when we're talking about common human systems like market prices, water systems, sewage disposal, traffic (yes, yes, I know there are [standards for that](#)), aid (yes, yes, [IATI](#)), that's a need that's calling out for coordination. When I can dial up and compare the water point systems for Uganda and Brasil, I'll be a lot happier bunny.

4. Keep it simple. But not **too** simple.

Websites and mobile apps rule today. In New York, Ruby coders are at a premium over C++ (remember that?) because the driver now is user interaction and simplicity rather than algorithms and depth. Which is great if you're building a consumer site, but not so great if you want to fairly handle complex uncertain data.

Well, that's as far as I got with that thought. More soon.