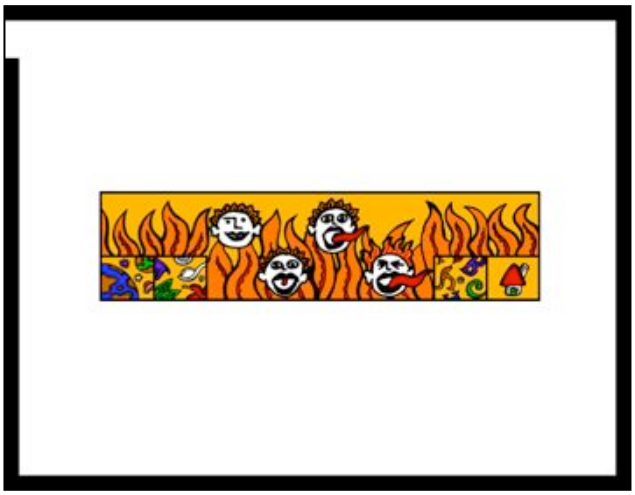
Good evening. It's a pleasure to be here. I've been to several of these and it's always a pleasure to participate. Today I'm going to be presenting some ideas about how I believe the market for starting and running digital hardware companies is changing and how new tools have enabled us to start thinking about creating electronics using Lean Startup methods. The content in this presentation is speculative, and I'm looking for this to be the beginning of a conversation, so I'm looking forward to chatting with everyone afterward.



First, let me tell you a bit about my background. I’m a user experience designer. I was one of the first professional Web designers in 1993, where I was lucky enough to be present for the birth of such things as the online shopping cart and the search engine. This is the navigation for a hot sauce shopping site I designed in 1994.

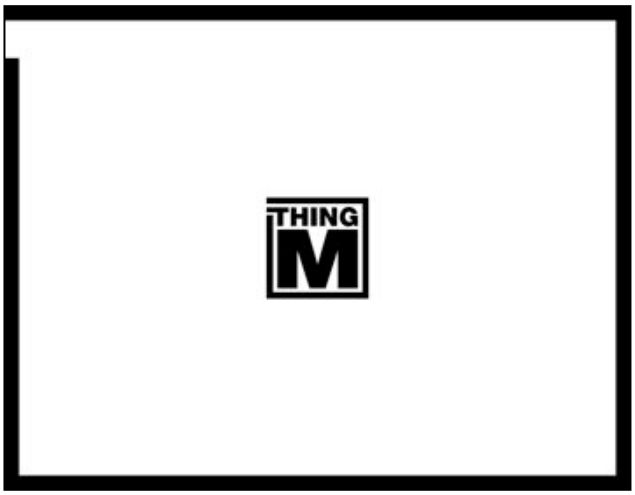
I'm proud of the fact that 16 years later they were still using the same visual identity.

Here's one of my UI designs for the advanced search for HotBot, an early search engine, from 1997. If you're wondering why Google's front page was so stripped down, I think it was because we did this.

Since then I've consulted on the user experience design of dozens, maybe hundreds of web sites. Here's one for credit.com, who were fantastic clients a couple of years ago.

I sat out the first dotcom crash writing a book based on the work I had been doing. It's a cookbook of user research methods.

And 2001 I co-founded a design and consulting company called Adaptive Path.



I left the Web behind in 2004 and founded a company with Tod E. Kurt called ThingM in 2006.

We're a micro-OEM. We design and manufactures a range of smart LEDs for architects, industrial designers and hackers. Most Makerbots that have lights on them use BlinkMs, as does Lady Gaga's stage show, Italian theater productions and probably 300 Burning Man costumes every year. However, as a startup, it's mostly a hobby. It's no one's full-time project and makes enough money to pay for our employees and to have some left over for a couple of R&D projects every year. This is an RFID wine rack that we did about four years ago. The different light colors represent different facets of information that's pulled down from a cloud-based service, such as current market price.

In 2010 I wrote a book on the user experience design of ubiquitous computing devices, which I define as things that do information processing and networking, but are not experienced as general purpose computing or communication devices.

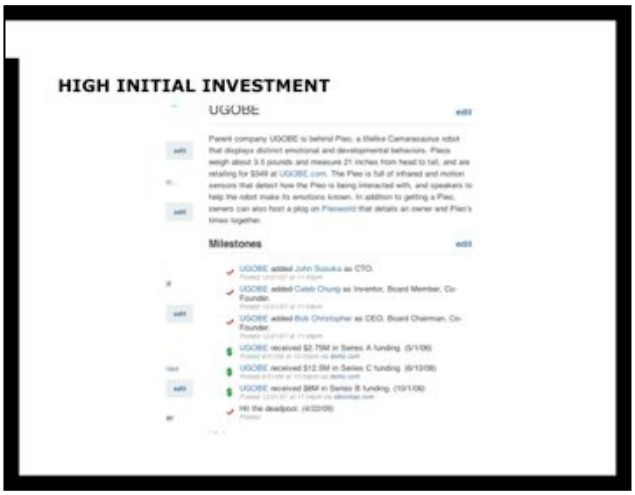
I also organize an annual summit of people developing hardware design tools for non-engineers.

However, ThingM, books and conferences are not my day job. They're entertaining sidelines. My primary day job is as an innovation and user experience design consultant focusing on the design of digital consumer products. Here are some I've worked on for Yamaha, Whirlpool and Qualcomm.

The last couple of years my clients have been large consumer electronics companies. I've helped them design new innovative products and services and to create more user centered company cultures. I can't give you any details.

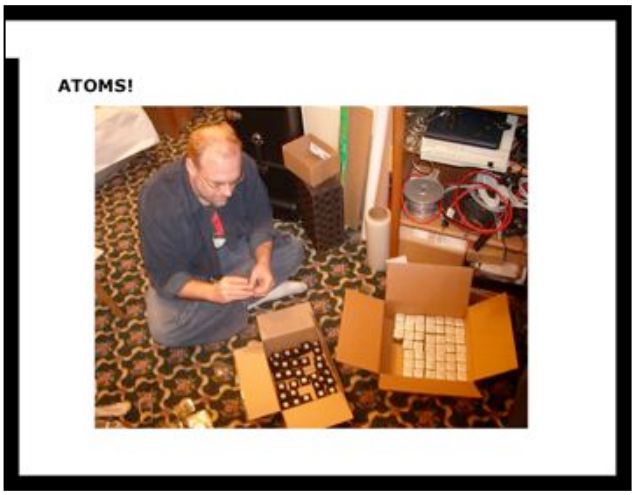
About a year and a half ago I started a new startup called Crowdlight, based on some technology we had developed at ThingM. We were not successful at raising money or getting enough traction, so we mothballed it, but the difficulties we faced led me to thinking a lot about how hardware product companies are started, funded and, based on conversations I had with Joel Truher, who's here tonight, how we can import ideas from lightweight internet startups into the manufacturing space. How can we be more like Github and less like General Motors?

Let's first review the traditional challenges to hardware.



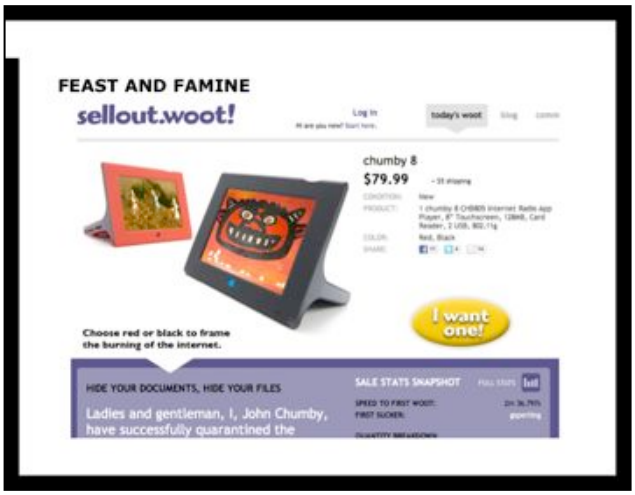
Traditionally, hardware takes a high initial investment because the philosophy is that you have to capture economies of scale, which means that you have to make 10,000 units or more of a product to justify making it at all. This creates a cascade of assumptions: well, if you're going to make 10K units, you'd better make the right thing and you'd better make them bulletproof because you likely can't afford to make two rounds the first time out. This means extensive market research and industrial design and engineering. If you look at the initial investment of most recent consumer electronics startups, you see pretty high numbers because of that very line of reasoning. Zeo, $2.5M initial investment; Fitbit, $2M Series A; Sifteo, $1M Series A after a $500K grant. Compare that to web/mobile companies and you see that they typically have much smaller initial investments. Here are the investments that Ugobe, who made the Pleo dinosaur got. One of the founders had earlier invented the Furby, so they were able to secure $11M their first year to make it to their first Christmas on the market in 2007. That's astronomical and if investors hadn't thought "Furby" the whole time, they would never have gotten such money.

The second big challenge boils down to logistics of moving atoms around. With atoms you have to assemble them, you have to sell them, you have to ship them. Let me use the example of ThingM. This is our first batch of BlinkMs from four years ago. We don't package them like this anymore…



because we had to do all of the packaging ourselves. You multiply any activity by 1000 and it sucks. But this was the least of our problems. To make our product we had to set up a supply chain. To make one of our LinkM USB dongles takes something like seven separate shipping events, and one part crosses the Pacific three times. To get to end consumers we had to build sales and distribution channels, which required collecting the names and contact information for every possible distributor of our product, That's more than 500 vendors we had to cold call. Then we have to ship them to the vendors. That requires finding a fulfillment service that has a good API and reasonable rates, then tying that into an ecommerce front end so that when someone from Germany wants to buy 150 BlinkMs and have them shipped three day air using their own DHL number, you don't have to figure that out by yourself, because—again it's not hard, when you multiply it by 20 orders, suddenly you're talking about two days spent at the post office filling out international shipping manifests. I know I may sound like I'm complaining, and to some extent I am, but only because these were the things that were really painful for us, and which are the hidden costs in creating a hardware company.

The bottom line is that because atoms are not bits, actions on them do not scale the same way as they in software. When you're working with bits, you can solve a problem once, and then your costs in solving that problem in the future are a tiny margin of the cost of solving it the first time. With atoms in the current system, when you solve it once, your costs of solving it subsequent times are actually a significant proportion of that first time. This really adds up, and this is what requires so much investment when the assumption is that you have to make thousands of a given item to make it worthwhile.



The pattern that physical products sells is different than online products. Because it's so much harder to scale their distribution dynamically, finances come in bursts. In consumer electronics, for example, it's a couple of giant bursts—Black Friday is the shopping that puts retailers in the black in November after ten months of losing money. As an OEM we see similar cycles, with a big burst at the beginning of the year when everyone has fresh budgets and then several smaller bumps throughout the year until another big one in September in preparation for Christmas. The issue with this is that baseline expenses are not bursty and they add up. if you miss a key burst of interest, say not having your products finished until November, you may not have a chance to make ANY money that season, and you're sitting on a bunch of inventory, which quickly becomes stale, and your employees, who need paychecks regularly. The traditional thing is that companies then sell their inventory at a giant discount after the big season is over so that they can pay their bills and hope they can use that make up their expenses to make it to the next burst. That's how stuff ends up on Woot at 1/3 the price nine months after its big launch. What this means is that the primary source of customer validation comes when its too late to change course. Steve Blank, the patron saint of the lean startup, says, "You don't know if you're wrong until you're out of business/money." He said it about traditional software startups, but it's even more true about products. (http://www.slideshare.net/venturehacks/customer-development-methodology-presentation).

Investors are in the business of making money by managing investment risk, so when they look at traditional hardware startups they see a lot of extra risk. It's expensive to design and manufacture, you have to get the logistics right when moving and selling atoms, and the "time to traction" is long, so there are only a couple possibilities for customer validation every year.

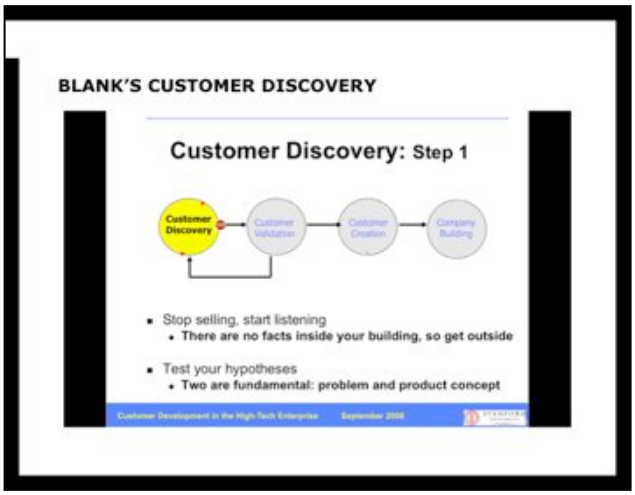
Investors calculate how much money they would have to spend to reduce that risk, to hire experienced people, build infrastructure, get sales channels, and compare that to web/app startups, where expertise is high, the infrastructures are robust and cheap, and sales channels are pretty well established and have a low barrier to entry. They come to the conclusion then, that it just costs too much money to run a hardware startup given the risks. In other words, hardware startups are not capital efficient.

If you look at many of the hardware startups that have been funded lately, you'll see that they're actually Web services that have physical components. They're primarily structured as web services that use a very simple device to get a data stream into or out of the cloud. Most of the value comes from the service that's in the cloud. I don't know if these startups essentially presented themselves to investors as "Web sites with hardware benefits" or if they presented themselves as a new form of consumer electronic device that's hard to counterfeit/duplicate because the functionality is not in the device, but in the online service, but regardless the form of these devices is one that's effectively designed to mitigate the risk of making hardware.

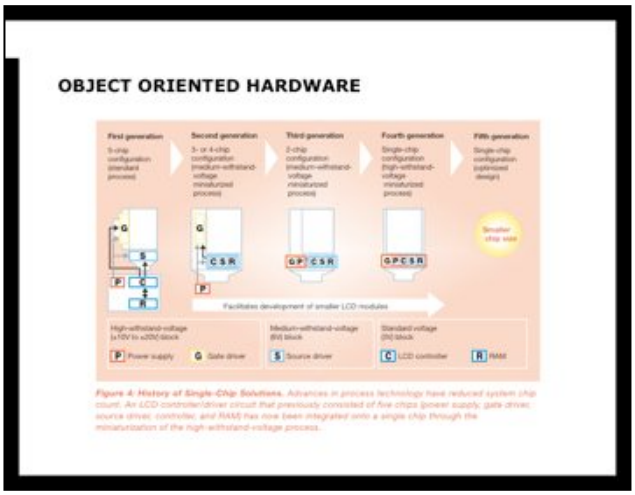
I think that there's another way. I think that it's time to move the design and manufacture of physical products out of the Industrial Revolution cycle of design-make-sell. I think that it's possible to use the modern tools of digital fabrication, online collaboration and ecommerce to create a new way to make things, one that's much more responsive to customer needs, much more manageable for company founders, much less risky for investors, and one that does not create the waste that traditional make-first manufacturing creates.

In the same way that Ruby on Rails, server virtualization and web analytics technologies enabled the design and funding model that's at the heart of lean web ecosystem I believe there are a handful of trends that combine to create an innovation ecosystem for designing, manufacturing and selling digital products.

As with software-based startups, the key is to enable groups to rapidly iterate product ideas and designs based on measurable customer validation.



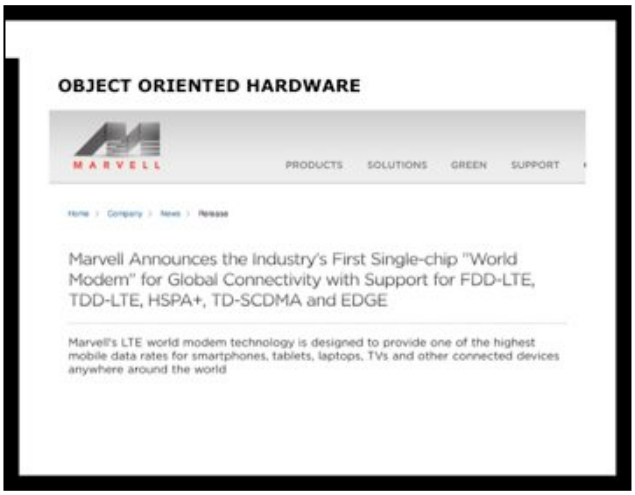
Let me me give a quick overview of how lean startups think about development. It's based on a mix of ideas from agile software development and user-centered design. The basic ideas is that you make minimal products, put them out in the world, watch whether and how people use them, and amplify the aspects that are getting traction. This is a slide from Steve Blank's 2008 on customer development that illustrates this basic idea.



First, let's start with tools that allow you to iterate designs quickly. There are a couple of technology changes that accommodating this.

Semiconductor manufacturers are putting increasingly more functionality on chips. Things that used to take five chips, as this diagram from Renesas Electronics shows, can now be done on one chip. This has all kinds of benefits from an assembly standpoint, but it also has an additional benefit. It creates an abstraction layer around a unit of functionality, in this case an LCD driver, to creates a single building block that's meaningful in human terms, rather than just electronic terms.

This is the start of object-oriented hardware. Each block is an atom of functionality that communicates with other blocks over a local network.

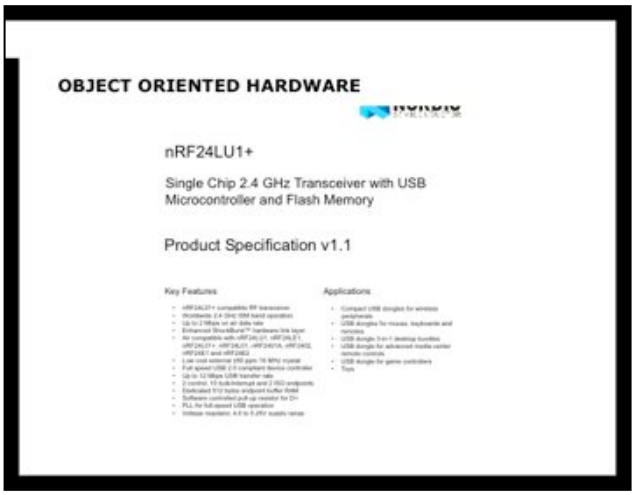


One block can do all of the work to connect to any phone network in the world.

Another is a complete GPS system.

Yet another is a multiaxis accelerometer that does the necessary math to clean up the signal.

This abstraction of knowledge into silicon means that rather than starting from basic principles of electronics, designers can focus on what they're trying to create, rather than which capacitor to use or how to tell the signal from the noise.



Still another is a single chip from Nordic that has both an 8051 microcontroller core, a wireless communication stack and a USB core. It's the core of lightweight wireless devices.

Once you have the breadboard design, you need to be able to get products out there, so you need assembly.

Fortunately, assembling electronics at small scales has gotten cheap. It's not just that it's cheap to ship stuff to Asian factories and to get PCBs made in small runs, but it's surprisingly inexpensive to assemble hardware in medium sized runs yourself. Not ten units, which you can do by hand, and not a million, which requires a serious setup, but, say 100, 1,000, or 5,000. This puts the idea of making small run electronics into cottage industry magnitude and brings it back closer to the hands of designers.

This is one of Sparkfun Electronics pick and place machines.

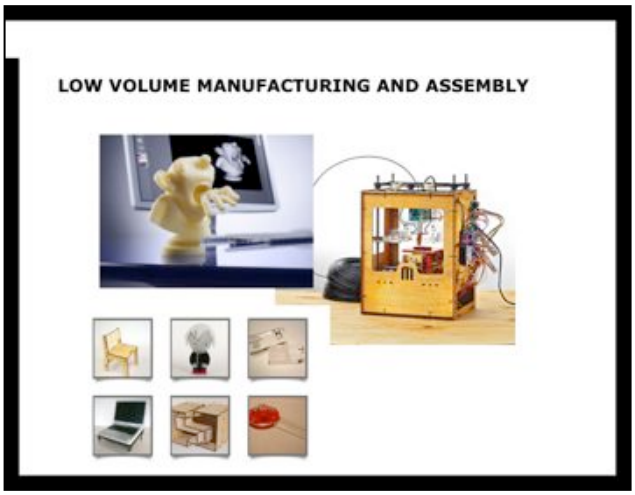
Source: Sparkfun

This is Adafruit's, who work out of a loft in New York.

Source: Adafruit

This is DIYDrones'. These are small companies that are nevertheless big enough that they decided to make their own electronics, because it's now a reasonable business decision.

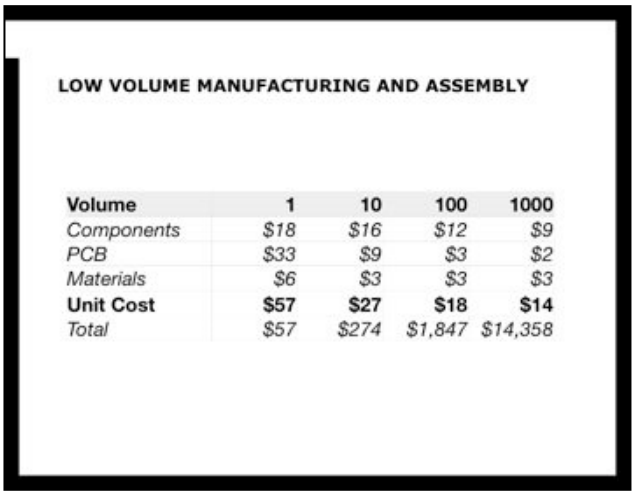
Source: Chris Anderson, DIYDrones



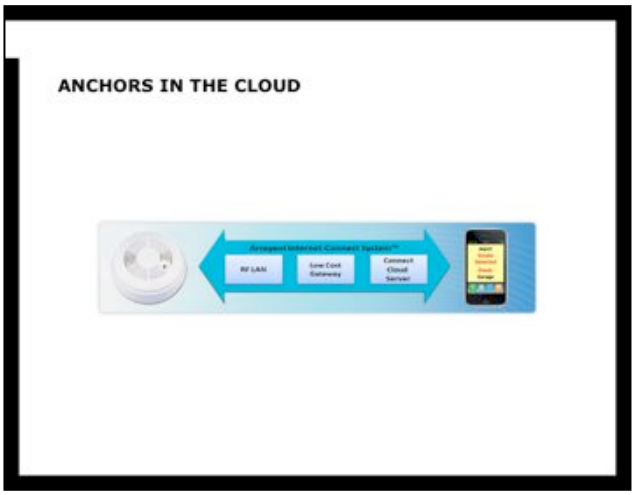
Of course only nerds will buy a plain circuit board, but if you put the circuit board in a case, it becomes a product. That's where all of the 3D printing and lightweight manufacturing come in. The price of 3D printing and laser cutting are coming down quickly and allow you to iterate rapidly on designs, then produce them in small quantities. It's not a great technology for making thousands of things, but for making 10, then making another slightly different 10 a day later, it's great.

This is a digital radio that David Mellis did as part of his MIT Media Lab master's thesis last year. It's designed to be a base design that can be quickly iterated on and to use cheap materials. Dave is one of the developers of the Arduino, and he intentionally made a very technologically simple product to focus on the core ideas of lightweight manufacturing and rapid iteration.

Here's the inside of another design based on the same idea. He ran workshops with people who had never designed a product and they would iterate on this idea and create their own versions in an afternoon.



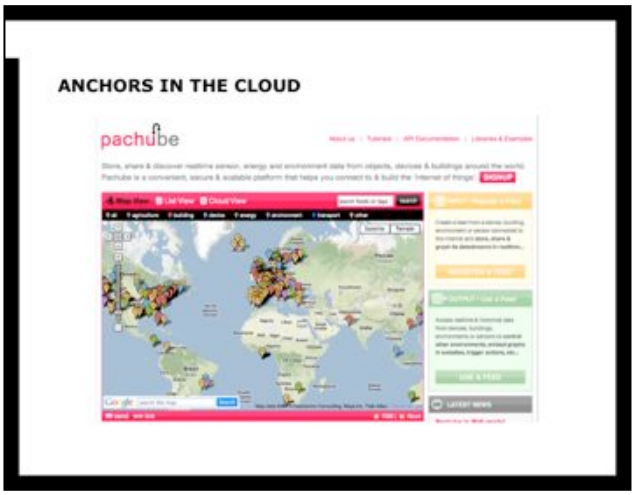
Here are Mellis' costs. You can see that the unit price for the materials drops very quickly. With a device like this you can imagine making small numbers of a product quickly.



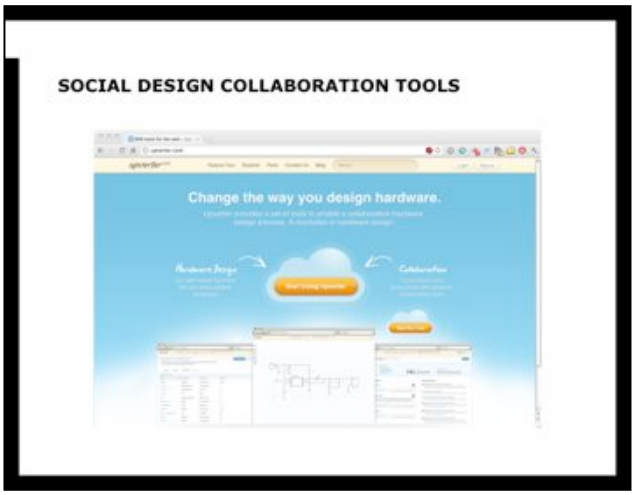
Connecting devices to the cloud, much like Fitbit, Nest and Zeo do allows you to both iterate quickly on features, since most of the functionality of those devices lies in the cloud-based service that they're attached to and to collect usage analytics.

Moving functionality to the Web is a pretty natural step. After twenty years of the Web, there's a lot of familiarity with it. Designers and developers are immersed in Web-like ideas. We increasingly think of digital technology as inherently anchored to the cloud and intuitively understand the possibilities that networked connections provide. There are embedded hardware products, hardware objects, that will do all of the provisioning of a service in the cloud once a connection is made. I grabbed this image from Arrayent, who is a company that makes a little hardware blob that connects virtually anything, in this case a smoke detector to their cloud service.

Source: Arrayent

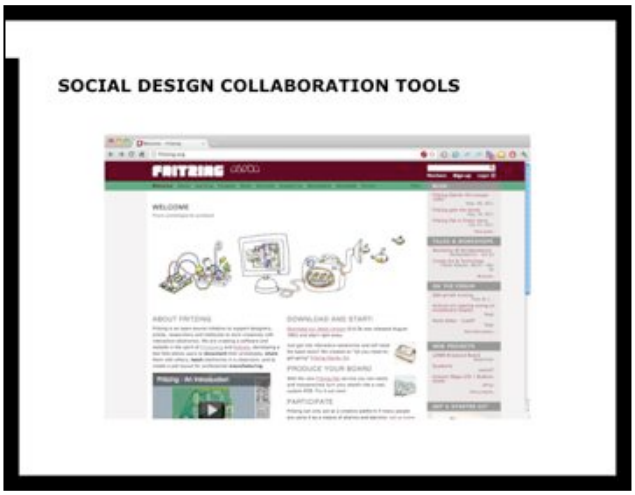


Moreover, there are now services such as Pachube, which was recently acquired, that allow an arbitrary data stream from any net connected device to share that stream with any other device. Pachube will do the buffering, the protocol translation, the analytics, everything. One device publishes an output stream, another device then subscribes to it it. It's a system that has its roots in Web mashups, now mapped to hardware



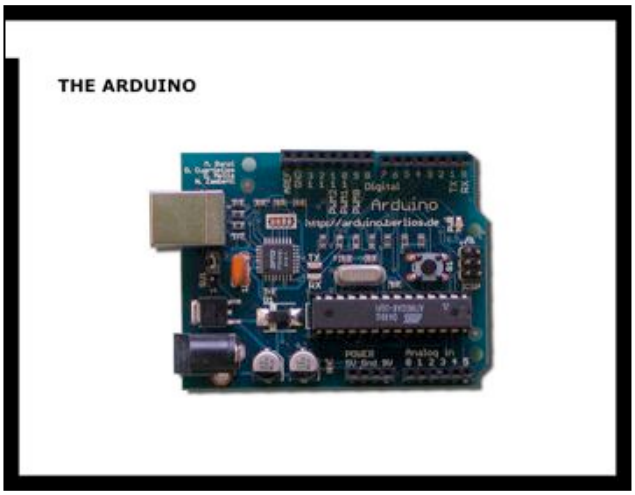
One of the most exciting changes is the movement of hardware development tools online. Hardware development used to be a solitary activity done in a lab with an oscilloscope and a soldering iron. Now it's becoming increasingly a social activity thanks to a new generation of online tools. The more easily people can share the design of a product, especially across discipline boundaries, the more quickly that product can be iterated.

Upverter, a Y Combinator-funded startup that just launched their beta, is a product that integrates electronic design with social collaboration. It's sort of like SourceForge, or GitHub for hardware.

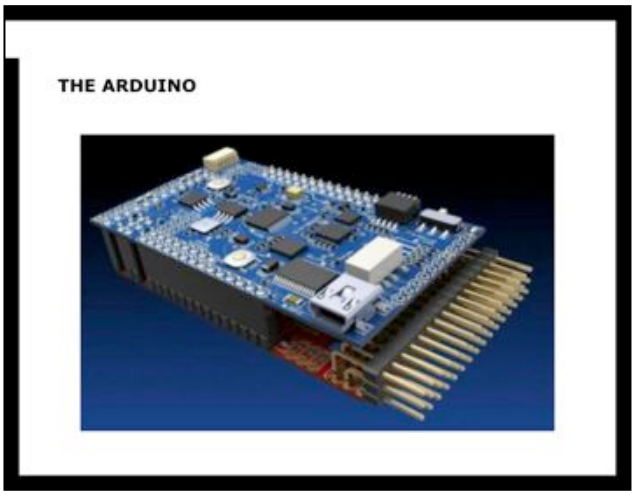


This is Fritzing, a open source project for online social hardware design. They will even print the circuit board for you and mail it to you.

Once you have social collaboration and the publishing and subscription of designs, schematics and code, you have the equivalent of View Source for hardware design. That, in turn, means that designers no longer have to start from scratch or from electronic textbooks or worry about asking noob questions on discussion boards. It's a model taken directly from how the Web grew.



The Arduino platform is probably the most mature and successful product to have come out of this type of collaborative technology environment. I'm sure everyone here is familiar with it, but I'd like to revisit it and talk about why it's especially important.

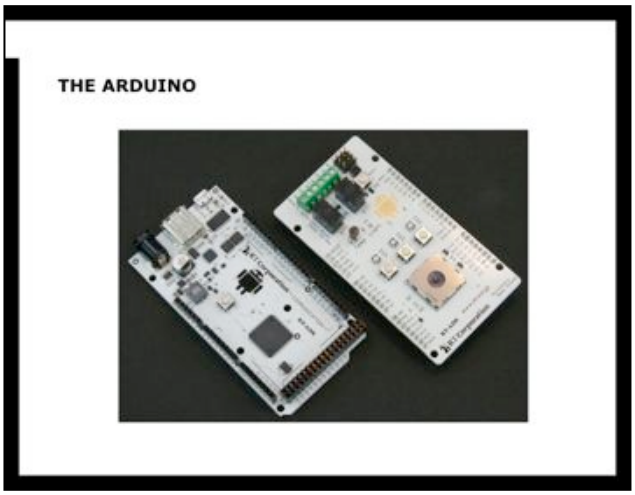


It's important because it's has become the reference platform that people extend to accomplish specific things. Here's the Ardupilot drone controller from the DIYDrones folks.

Source: DIYDrones

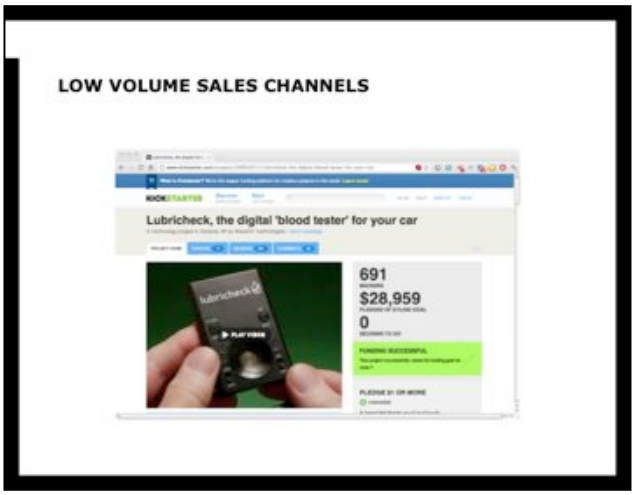
Here's one from All Power Labs that's used for precisely controlling an alternative energy gasifier unit.

Source: AllPower Labs



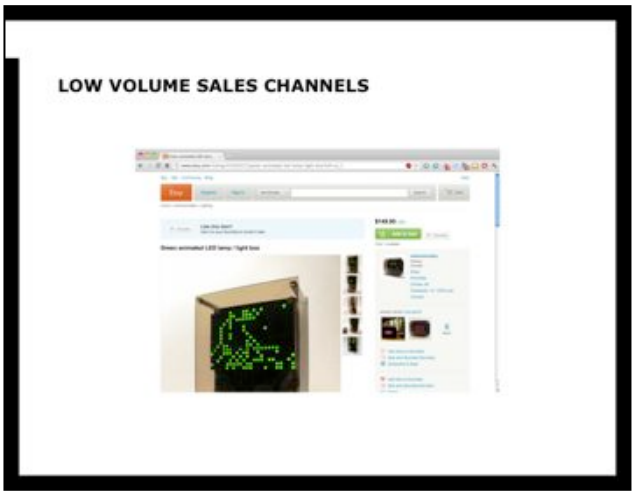
Here's Google's Open Accessory development platform. It's also based on the Arduino.

There were microcontroller platforms before, but the Arduino's popularity and flexibility makes it the Linux of Internet of Things hardware. Odds are that if you want to do something, there's a free design for it out there that runs on the Arduino. Thus, what makes it important has little to do with the actual hardware, but the community that's formed around it, much like the success of Linux wasn't because it was a superior Unix variant from the start, but that it was made that way by a community. The Arduino is therefore not the killer hardware app, but it forms the bedrock on which applications are built.



The final component of the ecosystem is probably the most important and least developed. It's a marketing and distribution mechanism that allows people to sell hardware in low volumes so that they can close the lop of customer validation, and perhaps cover their costs and generate operating income in the process.

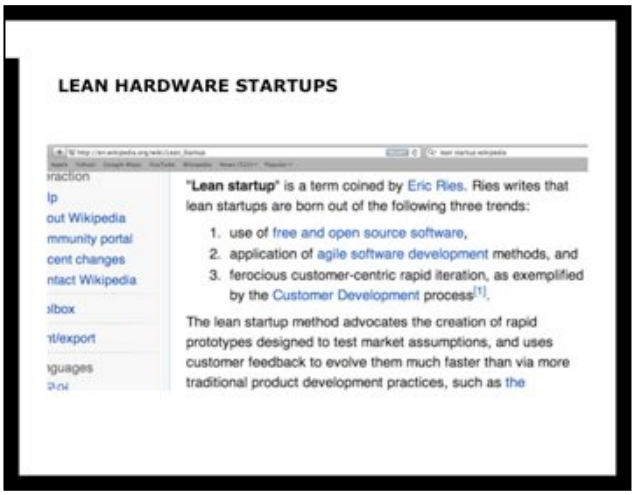
Kickstarter, in this instance, acts like a group buying site for products that don't exist yet, giving developers feedback about the popularity of their idea and teaching them how to position it for a market before they've made a single product.



Etsy allows very small run electronic products.

Even fab.com, which sells limited edition high design products like rugs and backpacks sells low run electronics.

These channels are immature, but they're becoming increasingly popular. In effect, they're doing an end run around the traditional consumer electronic sales channels to address the long tail of electronics buyers. That also happens to be where much of the greatest innovation happens.



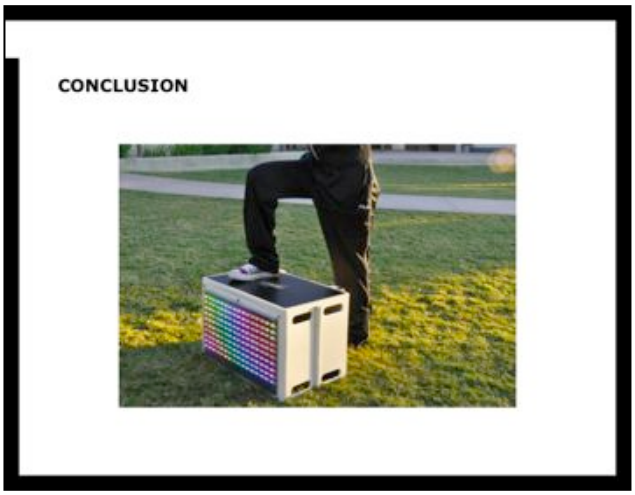
If we look at Eric Ries' definition of what makes a lean startup, here taken from Wikipedia, we can see all the pieces in this new ecosystem.

The tools are free and open. The costs for testing and assembly are low.

Object oriented hardware and social tools enable rapid iterative design and development, while cloud computing allows for rapid deployment of associated services.

Although they're immature, we're getting increasingly more low volume sales channels to test out ideas. I've singled out Kickstarter because in addition to sales, it provides feedback even before there are any sales, which is even more in line with the lean startup philosophy.

In the end what I am describing here is not the Internet of Things, or ubiquitous computing, but it is the innovation ecosystem that will lead to the Internet of Things.



When you put these approaches together, you start to see a new way of making things, one that's more like creating software than industrial manufacturing. By allowing small batches of products to be made, sold and distributed, you reduce the logistics required, create a channel for near-immediate feedback, and shrink iteration cycles to from years to months. Taken together, these practices enable an ecosystem where people can be more creative, creates a method where designers can be more responsive to niche markets, reduces the risk of investing in product companies, and reduces the amount of investment required.

There are of course downsides to working like this, but I believe that they're outweighed by the enormous potential of creating lean hardware startups that focus on the challenges of designing digital products that make people's lives better, not how you ship five thousand things from point A to point B.

I plan on pursuing these ideas in a practical way this year. ThingM is going do a couple of Kickstarter projects for various ideas we have, and I'm looking for collaborators who are interested in working this way to create a new generation of products and ventures. If you're interested, please talk to me.

Image: Legend Performance Cheer box prototype by Abraham Peters


THING
M
Mike Kuniavsky
mikek@thingm.com
SMART THINGS
Ubiquitous Computing User Experience Design
MK
MIKE KUNIAVSKY

Thank you.