

O'REILLY®

Creating Functional Teams for the IoT

Hardware, Software, and
Manufacturing Collaboration



Mike Barlow

O'REILLY®

Solid

C O N F E R E N C E

THE NEW HARDWARE MOVEMENT

“The future has a funny way of sneaking up on you. You don’t notice it until you’re soaking in it. That was the feeling at O’Reilly’s Solid Conference.”

—Wired



Connected, intelligent hardware is the next great technology opportunity—one that promises to revolutionize every industry. It’s getting easier to design, engineer, prototype, manufacture, and market physical products, putting innovation within reach of startups and giant enterprises alike.

Solid is O’Reilly’s conference about creating innovative connected products, and it’s for everyone involved in the hardware creation process: executives, entrepreneurs, engineers, developers, designers, researchers, and investors. It’s the only conference that covers the entire product-creation stack from conceptualization through shipping, and it offers the perspectives of leaders from fields as diverse as consumer electronics, agriculture, healthcare, and heavy industry, as well as a glimpse at the cutting-edge technologies that are about to remake the world.

Find out more at solidcon.com

Creating Functional Teams for the IoT

*Hardware, Software, and
Manufacturing Collaboration*

Mike Barlow

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Creating Functional Teams for the IoT

by Mike Barlow

Copyright © 2015 O'Reilly Media. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://safaribooksonline.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editor: Susan Conant

Interior Designer: David Futato

Cover Designer: Randy Comer

June 2015: First Edition

Revision History for the First Edition

2015-06-01: First Release

2015-09-23: Second Release

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Creating Functional Teams for the IoT*, the cover image, and related trade dress are trademarks of O'Reilly Media, Inc.

While the publisher and the author have used good faith efforts to ensure that the information and instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-491-93274-2

[LSI]

Table of Contents

Building Functional Teams for the IoT Economy.....	1
A More Fluid Approach to Team Building	3
Raising the Bar on Collaboration	5
Worlds Within Worlds	6
Supply Chain to Mars	8
Rethinking Manufacturing from the Ground Up	10
Viva La Revolución?	11

Building Functional Teams for the IoT Economy

You don't have to be a hardcore dystopian to imagine the problems that can unfold when worlds of software collide with worlds of hardware.

When you consider the emerging economics of the Internet of Things (IoT), the challenges grow exponentially, and the complexities are daunting.

For many people, the IoT suggests a marriage of software and hardware. But the economics of the IoT involves more than a simple binary coupling. In addition to software development and hardware design, a viable end-to-end IoT process would include prototyping, sourcing raw materials, manufacturing, operations, marketing, sales, distribution, customer support, finance, legal, and HR.

It's reasonable to assume that an IoT process would be part of a larger commercial business strategy that also involves multiple sales and distribution channels (e.g., wholesale, retail, value-added reseller, direct), warehousing, operations, and interactions with a host of regulatory agencies at various levels of government, depending on the city, state, region, and country.

A diagram of a truly functional IoT would look less like a traditional linear supply chain and more like an ecosystem or a network. The most basic supply chain looks something like this:

Supplier → Producer → Consumer

If we took a similarly linear approach and mapped it to a hypothetical IoT scenario, it might look something like this:

Entrepreneurs → Capital Markets and Investors → Sales → Software Development → Hardware Design → Prototyping → Manufacture → Warehousing → Distribution/Logistics → Wholesale → Retail → Consumer Markets → Customer Service → Finance → Legal → HR → Regulators

Even an abbreviated description of a hypothetical IoT “supply chain” reveals the futility of attempting to map a simple linear sequential model onto a multi-dimensional framework of interconnected high-velocity processes requiring near-real-time feedback. A practical IoT ecosystem model would look something like this:

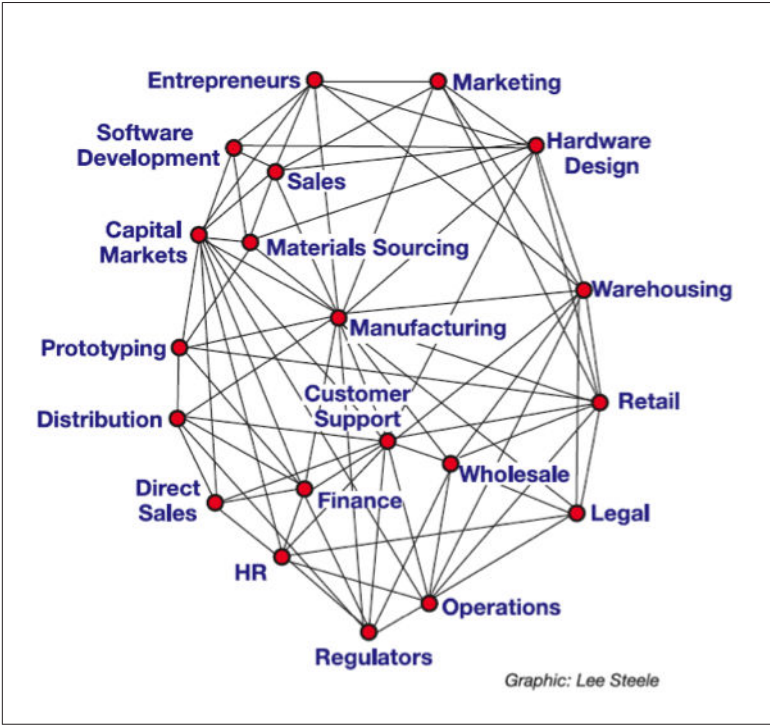


Figure 1-1. Sketch of a hypothetical IoT ecosystem

In all likelihood, the IoT economy will resemble a network of inter-related functions, responsibilities, and stakeholders. The sketch only gives a rough idea of the complexities involved—each node would have its own galaxy of interconnected processes, technologies, organizations, and stakeholders. In any case, it’s a far cry from sim-

ple “upstream/downstream” charts that make supply chains seem like smoothly flowing rivers of pure commercial activity.

Unlike traditional software development scenarios, IoT projects are undertaken with the “real world” clearly in mind. It’s understood that IoT services and products will “touch” the physical universe we actually live in. If the assignment is developing new parts for jet engines used by passenger airlines, there will be real consequences if the parts don’t work as promised.

A More Fluid Approach to Team Building

Working relationships between software developers, hardware designers, and manufacturers within IoT ecosystems are still taking shape and evolving. The structure, staffing, and organization of a team would depend largely on the project at hand and the depth of available talent.

“There isn’t an ideal team. You have to start with the problem and work backwards to discover what is available and commoditized, and what really needs to be created to make a solution real,” says Andrew Clay Shafer, senior director of technology at **Pivotal**, a company that provides a cloud platform for IoT developers. “The dream team would have unlimited resources, expertise, and imagination. Realistically, you won’t have a dream team, so you are better off examining the problem, looking at your available resources, and filling in gaps.”

Mike Estee, CTO and co-founder of **Other Machine Co.**, a company that builds machines and creates software for desktop manufacturing, says the firm looks for software developers with experience in “building consumer-facing applications that integrate directly with hardware designed in tandem” and hardware designers who can bring products to market “on time, on budget, and with a high degree of polish.”

Ayah Bdeir is the founder and CEO of **littleBits**, a company that creates “libraries” of electronic modules that snap together with magnets. An advocate of the open source hardware movement, she is a co-founder of the **Open Hardware Summit**, a **TED Senior Fellow**, and an **alumna of the MIT Media Lab**. From her perspective, the “ideal” development team includes hardware, firmware, software,

and web experts. Tightly integrating the design function is absolutely critical, she says.

“Design is crucial at the very beginning. Our second hire in the company was an industrial designer, and it has transformed our business and our product,” Bdeir says. “Specific skills and languages don’t always mean as much as the ability to adapt. The team needs to be creative, agile, and quick to learn.”

While software development is considered a relatively “mature” field, the need to combine software development and hardware design has spawned a host of new problems. In an email, Bdeir shared her view of the current situation:

“For software developers, a lot of the barriers have been removed. Today, you can prototype and launch a software product with very little time, money, and most importantly, few external dependencies. When you see there is traction, you can scale your offering gradually, and not overextend yourself financially.

For hardware, the barriers are much more defined and difficult to deal with. You can prototype a lot faster, and get initial interest/traction through Kickstarter, press, or showing off your product at events. But there is a pretty big step to getting first-run manufacturing done.

Finding manufacturing partners who will help you make your product manufacturable is hard when you are at a small scale, and it takes a lot of time. When we were starting out, we had to manufacture our magnetic connector with a factory that made plastic insoles and knee guards because I couldn’t find any manufacturer that knew electronics or connectors that would take me seriously.”

Estee, who spent nine years as a software engineer at Apple, has a similarly chastened view of the challenges. Here’s a brief excerpt from an [essay](#) he posted recently on Medium:

“If you’ve never started a hardware company before, it’s very hard to understand the vast differences between a prototype, a product, and a company which supports it. A product is so much more than just the machine. It’s also the packaging, the quality assurance plan, documentation, the factory, the supply chain, vendor relations, testing, certification, distribution, sales, accounting, human resources, and a health plan. It goes on and on. All things you probably weren’t thinking about in the rush to get the Kickstarter launched.”

Indeed, the complexities seem nearly inexhaustible. Shafer describes the challenges from an industry-wide perspective:

“As we move into a world where all things are integrated, the stack is getting wider and deeper. The choice is between building teams that have expertise at each level ... or choosing platforms to do the heavy lifting and act as force multipliers for teams with more focused skills. The challenge is deciding which problems to solve yourself and which to outsource.

People get excited about creating ‘things,’ but the Internet is the other half of the equation, and there are still hard problems to solve there. Even moderate IoT build-outs could quickly create sensor networks that have latency, scaling, and analytics problems that only a handful of the big web companies have had to solve until now.

Today, there is no equivalent to the LAMP¹ stack for the IoT. It will be several years before a common way of solving IoT problems begins to emerge.”

Raising the Bar on Collaboration

At least one evolutionary path seems fairly certain: developing products and services for the IoT will require significantly higher levels of collaboration and information sharing than those considered normal or acceptable in traditional development scenarios.

Harel Kodesh, vice president and CTO at **GE Software**, recalls when software development was a mostly solitary endeavor. “Now it’s a team sport to the extreme,” he says. “We have many more releases, but the prototypes are smaller and there are many more components. When I was at Microsoft in 1998, you sat at your workstation and the only thing you had to think about was the code you were writing. At the end of the day, or the end of the week, someone called the ‘build master’ would go around, collect the code everyone had written, compile it, and try to get it to work.”

Since testing was only partially automated, most of it was done manually. It wasn’t unusual for a company to spend five years developing and testing code for a new operating system. Today, development and testing move at greatly accelerated speeds. People generally expect software and related systems to work—especially when they’re developed for the IoT.

¹ LAMP stands for Linux, Apache, MySQL, and PHP.

The biggest difference between then and now, Kodesh says, is the cloud. In the past, developers ran code on workstations; now they run it in the cloud. For IoT development teams, the cloud is a blessing and a curse. The cloud's ability to deliver scale at a moment's notice makes it invaluable and indispensable for developing IoT products and services. GE believes in its cloud development model. But if you're a passenger riding in a high-speed train that's built with subcomponents that were designed in the cloud, should you be concerned?

Those kinds of concerns are leading GE and other organizations, such as NASA and the US Air Force, to explore the creation of “digital twins,” which are extremely accurate software abstractions of complicated physical objects and systems. A digital twin would be linked to its physical counterpart through the IoT. The steady flow of data from the real world means the digital twin can be updated continuously, enabling it to “mirror” the conditions of its physical counterpart in real time.

Ideally, digital twins would serve as “living” models for machines and systems in physical space. And since it's unlikely that software developers would have access to jet engines or diesel locomotives while they're in service, the digital twin concept is expected to play a major role in the evolution of IoT development and design processes. In case you don't believe the IoT will have any impact on you personally, doctors are already talking about using digital twins in healthcare scenarios. [A recent article in The Guardian](#) mentions the possibility of creating digital twins for soldiers before they deploy for combat missions, which would allow doctors to create replacement parts for wounded soldiers quickly and accurately.

Worlds Within Worlds

Looking under the hood of an IoT software development process reveals even more complexity. IoT applications are often built from communities of smaller independent processes called microservices. Facebook and Google are common examples of microservices architecture. Users are typically unaware of the granularity of the application until a particular component, such as your newsfeed on Facebook, stops working. Usually the glitch sorts itself out quickly, and you don't even realize that part of the application had crashed.

Microservices architecture is particularly useful in IoT scenarios because it enables developers to create highly usable applications that are extremely scalable and easy to replace. In a sense, microservices architecture is reminiscent of service-oriented architecture (SOA), except that SOA was conceived to integrate multiple applications using a service bus architecture; with microservices, however, the service endpoints interact with each other directly to deliver a user-level application.

In his role as head of software engineering for Predix at GE Software, Hima Mukkamala supervises teams of developers working to create the microservices required for real-world IoT applications. In one situation, an oil production operator needed to compare the output of two production lines. “They asked us to build an asset performance management (APM) application so they could compare the two production lines and determine what changes were necessary so the two lines would produce the same amounts of oil,” Mukkamala explains.

Traditional development techniques would have taken months to produce a useful set of software tools for the business problem. Normally, that type of project would have involved separate teams for software development, hardware design, testing, monitoring, and deployment. Using a DevOps-style approach cut the time required from six months to six weeks.

Mukkamala says he was surprised by the relative ease of coding, building, monitoring, and deploying viable applications created from microservices. “I thought that would be harder, but it was less of a challenge than I had imagined,” he recalls. “What I thought would be easy, however, was transitioning from a traditional software development model to a DevOps model. That turned out to be a much more stressful challenge. For developers accustomed to a traditional SRA (software requirements analysis) approach, writing and deploying code every day was a shock.”

As CIOs like to say, “It’s never just technology. It’s always people, process, and technology.” Mukkamala’s hands-on experience testifies to the importance of acknowledging cultural barriers and implementing a change management process.

“In times past, developers would write code and throw it over the wall. Now they have to work very closely with the operations people,” he says. “Collaboration between developers and ops people has

to start from day one. Developers have to think like people in ops, and people in ops have to think like developers. Now we act like one team, all the way from development to operations.”

Vineet Banga is a development manager at GE Software. Working with Kodesh and Mukkamala, he often serves as a “human bridge” connecting senior management, software development, operations, and “machine teams” that finalize the marriage of code and physical devices.

“We’re moving out of our normal comfort zones and constantly trying to broaden our horizons,” he says. “The result is that our efforts are now always overlapping, which is good. On the development side, we’re looking more at hardware/operations and on the operations side, they’re looking more at software.”

Teams that were highly specialized a few years ago are now much more cross-functional and multidisciplinary. Team members are expected to write code in a variety of programming languages and be comfortable working with different layers of the software stack

A typical team might consist of two Java engineers, a web UI developer, a security engineer, and two engineers with hands-on infrastructure experience. Depending on the project, a team might include a Java developer, a software architect, a Linux systems developer, and user experience developer.

In any event, the team would be responsible for making sure the code they write provides the levels of scale, security, interoperability, and compliance required for the project. “We try to delegate as many decisions as possible to the team,” Banga says. “When you have six or seven people from different backgrounds working together, sometimes the best decisions are made by the team itself, and not by managers or executives.”

Supply Chain to Mars

Michael Galluzzi is lead business strategist for additive manufacturing and supply chain management at the **National Aeronautics and Space Administration (NASA)** Swamp Works Lab at Kennedy Space Center, Florida. From his perspective, the challenges of creating an IoT economy aren’t confined to Earth—they extend into deep space.

Imagine this scenario: NASA sends a spacecraft to Mars. Everything goes fine. Ten years after the spacecraft lands, its rover transmits an alert indicating the imminent failure of a critical part.

In a perfect imaginary world based on traditional logistics and current technologies, a spare part for the rover would be sitting in a warehouse somewhere and it would be dispatched to Mars on the next available rocket. The reality is far more complicated.

Aside from the fact that we don't send rockets to Mars on a regular schedule, not to mention the time it takes to get there, it's unlikely in these days of just-in-time (JIT) manufacturing that a backup version of the part would be available from Earth-based sources. A new part would need to be manufactured using *in situ* resources or ideally using regolith-based 3D printing or additive manufacturing technology, which NASA Swamp Works Lab is currently developing.

But here's where it gets really tricky: spacecraft parts are built to rigid specifications. Companies that produce spacecraft parts are carefully vetted by the government and its various regulatory agencies. Those kinds of companies tend to be small, and they often go out of business. So there's a very strong likelihood that the company that made the original part no longer exists—it's simple macroeconomics (i.e., low or no demand, no supply). That raises more problems: who owns the IP for the part? Who has the expertise and experience required to manufacture the part to NASA's specifications?

If the part is manufactured from rare earth minerals (and many spacecraft parts are indeed made from exotic raw materials), are those minerals still available, and what kinds of permissions will be required to obtain them?

Ideally, an IoT economy would be capable of solving those kinds of problems. But the IoT still lacks critical chunks of infrastructure, such as a common data platform with neural networking capabilities for sharing information up and down the value chain. In addition to data integration challenges, there are data quality issues. Some suppliers have invested in bleeding-edge advanced analytics, but many others still rely on Excel.

NASA typically deals with thousands of suppliers, from very large to small—high-tech “mom and pop” firms that produce small batches of highly specialized parts—and there's no common data ontology or digital thread for sharing the kinds of highly detailed data

required for creating replacement parts and sending them to other planets.

“The supplier community can’t access the data they need because the legacy information systems are not connected,” Galluzzi says. “The challenge is integrating the information systems down to suppliers at the lowest tiers of the supply chain to obtain the supply chain visibility needed to understand the multi-functional relationships of suppliers or at minimum, to mitigate counterfeit parts or supply chain disruption.”

The future, he says, will be written by organizations that develop capabilities for sourcing and distributing relevant data content at every link and lifecycle of the value chain, from development of new products and services to delivery—even when the consumers are located on Mars.

Rethinking Manufacturing from the Ground Up

Even if your supply chain doesn’t extend to Mars or beyond, the IoT is forcing suppliers at all levels to rethink and re-envision the standard manufacturing models that have long served the world’s industrial societies.

“The supply chain is an old term that conjures up processes that are linked together at every step, when in fact, they are diversified networks of suppliers,” says Dennis Thompson, who leads the manufacturing and supply chain services group at **SCRA**, an applied research company working with federal and corporate clients. He also serves in leadership roles at two public-private partnerships, the **Digital Manufacturing and Design Innovation Institute (DMDII)** and the **National Digital Engineering & Manufacturing Consortium (NDEMC)**.

Henry Ford’s idea of supply chain management, for example, was a vertically integrated corporation that owned or operated suppliers at every level of the production process, from hauling iron ore to delivering finished automobiles to a dealer’s showroom. “Today, we need to find ways of creating networks of suppliers and managing them as if they were vertically integrated, without owning their assets,” Thompson says.

At the DMDII, efforts are underway to invent a new generation of flexible supplier networks. “The challenge is pulling together networks of suppliers around projects, and then disassembling those networks when the projects are finished and new projects come up,” Thompson says. One of the tools necessary for creating agile supplier networks is “a next-generation ERP (enterprise resource planning) system that would link suppliers on a common platform.”

DDMII is building an open source software platform called the Digital Manufacturing Commons (DMC) for suppliers in the digital manufacturing economy. Some have already described the DMC as an industrial version of SimCity, while others see it as a virtual hub enabling communities of developers and suppliers to share information more effectively. Thompson is optimistic the DMC project will bear fruit, and so are DDMII’s corporate backers, which include companies such as Boeing, Lockheed Martin, Rolls-Royce, GE, Microsoft, Siemens, and Caterpillar. “They’re anticipating getting a return on their investment,” Thompson says.

Viva La Revolución?

Technology revolutions are typically accompanied or followed by cultural revolutions, and the IoT revolution is unlikely to break the pattern. As suggested earlier in this report, the primary challenges facing IoT pioneers aren’t technical. The hard parts of the problem involve getting disparate players such as software developers, hardware designers, and mechanical engineers to share knowledge and work collaboratively toward achieving common goals. It seems likely that new forms of management will be required, as well as new approaches to concepts such as intellectual property and corporate secrecy.

“I think it comes back to getting the right people at the table and pulling together networks of designers, developers, manufacturers, and sales people,” says Thompson. “The voice of the customer should be represented. If you build it, will they buy it? That’s a question you’ve got to ask.”

As product development cycles become shorter, incorporating the “voice of the customer” might become less of an afterthought and more of a standard checklist item.

The idea of “failing fast” has also gained traction in digital manufacturing circles. “Agile manufacturing is possible,” says Shashank Samala, co-founder of **Tempo Automation**, whose specialty is automating the manufacture of electronics. “Agile is relative, not absolute. If we can make manufacturing 10 times more agile than it is today, that will be an achievement.”

It’s easy for software developers to be “agile” because there are already huge libraries of code to draw from, and developers rarely have to start from scratch. “With software, you can turn your ideas into actual products very quickly. Most of the knowledge has already been abstracted. It’s more difficult to create new hardware quickly, because we don’t have modules that we can build on,” says Samala. “That’s why hardware is harder to develop than software. But we are moving in the right direction.”

Still, it seems as though software development races along while hardware development moves at a glacial pace. Bdeir sees the apparent dichotomy as less of a problem and more of a potential business opportunity for companies such as littleBits:

“Making a good product always takes time, whether it’s hardware or software. The difference is that with hardware, it either works or it doesn’t. You can’t iterate when something is on the market. So your MVP (minimum viable product) barrier has to be a lot higher.

We have a ways to go to raise the bar of development in hardware. Every hardware developer is practically starting from scratch: figuring out power distribution, sensor conditioning, getting drivers of hardware to talk to each other. It’s the equivalent of coding in assembly. We can and should simplify hardware design by making the design process more open and more accessible and most importantly, more modular so you can raise the level of abstraction from the component level to the interaction level. That has been the guiding principle behind littleBits: Stop wasting your time figuring out what the right resistor for that motor is, and focus on the motion that you need to make a magical experience.”

About the Author

Mike Barlow is an award-winning journalist, author, and communications strategy consultant. Since launching his own firm, Cumulus Partners, he has represented major organizations in numerous industries.

Mike is coauthor of *The Executive's Guide to Enterprise Social Media Strategy* (Wiley, 2011) and *Partnering with the CIO: The Future of IT Sales Seen Through the Eyes of Key Decision Makers* (Wiley, 2007). He is also the writer of many articles, reports, and white papers on marketing strategy, marketing automation, customer intelligence, business performance management, collaborative social networking, cloud computing, and big data analytics.

Over the course of a long career, Mike was a reporter and editor at several respected suburban daily newspapers, including *The Journal News* and the *Stamford Advocate*. His feature stories and columns appeared regularly in *The Los Angeles Times*, *Chicago Tribune*, *Miami Herald*, *Newsday*, and other major US dailies.

Mike is a graduate of Hamilton College. He is a licensed private pilot, an avid reader, and an enthusiastic ice hockey fan. Mike lives in Fairfield, Connecticut, with his wife and their two children.
