

# Beyond the Prototype: Understanding the Challenge of Scaling Hardware Device Production

Rushil Khurana

Carnegie Mellon University,  
Pittsburgh, PA, USA  
rushil@cmu.edu

Steve Hedges

Microsoft Research  
Cambridge, UK  
steve.hedges@microsoft.com

## ABSTRACT

The hardware research and development communities have invested heavily in tools and materials that facilitate the design and prototyping of electronic devices. Numerous easy-to-access and easy-to-use tools have streamlined the prototyping of interactive and embedded devices for experts and led to a remarkable growth in non-expert builders. However, there has been little exploration of challenges associated with moving *beyond a prototype* and creating hundreds or thousands of exact replicas – a process that is still challenging for many. We interviewed 25 individuals with experience taking prototype hardware devices into low volume production. We systematically investigated the common issues faced and mitigation strategies adopted. We present our findings in four main categories: (1) gaps in technical knowledge; (2) gaps in non-technical knowledge; (3) minimum viable rigor in manufacturing preparation; and (4) building relationships and a professional network. Our study unearthed several opportunities for new tools and processes to support the transition beyond a working prototype to cost effective low-volume manufacturing. These would complement the aforementioned tools and materials that support design and prototyping.

## Author Keywords

Hardware device realization; low volume electronics manufacturing; productization; long tail hardware.

## CSS Concepts

- **Hardware → Post-manufacture validation and debug;**
- **Hardware → Design for manufacturability; Applied computing → Industry and manufacturing**

## INTRODUCTION

At a time when machine learning and cloud computing are prominent topics in both research and industry, it is easy to overlook the physical hardware that underpins these technologies. However, hardware is as important as ever; it

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

*CHI '20, April 25–30, 2020, Honolulu, HI, USA*

© 2020 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-6708-0/20/04.

<https://doi.org/10.1145/3313831.3376761>

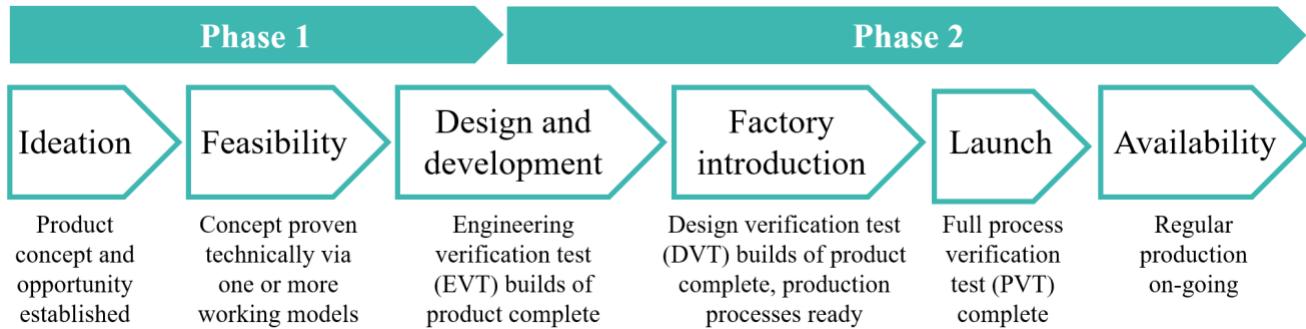
supports computation and storage in data centers and edge computers, and also provides the vital conduit for interaction and other forms of input and output. Indeed, many emerging applications and markets for technology – from smart homes to smart cities, and from interactive wearables to the industrial internet – are drivers for both the number and the variety of available hardware devices. Developing and deploying innovative hardware with new forms and functions is therefore as important as ever.

The process of realizing new hardware can broadly be split into two phases:

- Phase 1: A period of ideation, experimentation and design iteration leads to new device concepts and ultimately working prototypes.
- Phase 2: Fruitful concepts transition beyond the basic prototype, typically resulting in hundreds to hundreds-of-thousands of units.

The hardware device research community and the industry it serves have developed many tools and techniques to aid in Phase 1 above [13,16,34]. Powerful and well-documented hardware design systems are easy to access and use. The physical prototyping process is easier than ever given the sophisticated tools and components which are readily available. Collectively, these on-going innovations mean that experts can build all manner of interactive and embedded hardware prototypes quickly and cheaply. At the same time, non-experts are empowered to work in this space [13]; even high-school students can design and build innovative and useful devices – typically combining an off-the-shelf microcontroller board, some external components and a 3D-printed enclosure. This has led to a surge in hobbyists and tinkerers who are able to build one to maybe tens of units of a product, either for personal or small-scale commercial use.

However, we have not seen analogous advances in Phase 2. Certainly, large device companies can leverage carefully optimized manufacturing facilities and processes, resulting in mass production of high quality, keenly priced devices in volumes of upwards of a hundred thousand units a year. But without an up-front commitment to these high volumes a new device needs to go through a period of much lower volume production. Initially, tens or hundreds of samples of the device are necessary for effective evaluation; and if successful the device must then prove itself in the market as



**Figure 1: The two main phases of hardware development encompass six main activities. The transition from Phase 1 to Phase 2 typically happens when a handful of functional working prototypes have been made.**

a fully-fledged but initially low-volume product. Based on personal experience combined with anecdotal evidence collected over many years from others working in the hardware device industry, we have observed that it can be remarkably hard for small organizations to be successful during Phase 2. Without scale, it is hard to justify up-front costs (also called non-recurring engineering or NRE costs), to have sufficient leverage over suppliers, and to cover fixed overheads. In addition, newcomers to the productization process experience many unexpected and unintuitive challenges. Collectively, these issues make the transition from a few working prototypes in Phase 1 to viable low volume production in Phase 2 difficult.

In order to validate our anecdotal evidence, and to understand the challenges associated with both phases of hardware development more rigorously, we conducted a series of semi-structured interviews with practitioners who have experience building devices in low volumes. Our aim was to develop a balanced and comprehensive picture, and to uncover potential opportunities to reduce barriers and ease the prototype-to-product transition. Through this process we learned, amongst other things, that:

1. Creators (device builders) have knowledge gaps but do not have access to the right resources to fill them.
2. These gaps cover both technical and non-technical knowledge.
3. There is a need for more rigor in manufacturing preparation to avoid issues at later stages.
4. Creators gain value from building strong relationships with partners and peers, and subsequently leverage this professional network for on-going assistance.

We make two contributions in this paper. Firstly, we describe our view of the common challenges faced by those aiming to manufacture a device in low volumes, typically hundreds or thousands of units a year. Secondly, our findings led us to several potential opportunities for new processes and tools to streamline the transition from Phase 1 to Phase 2, which we also present.

## BACKGROUND AND RELATED WORK

We start this section by describing in more detail what was introduced above as a two-phase process to realize a new hardware product. We summarize common challenges associated with the transition from prototype to product and we discuss the limited prior work addressing issues in scaling hardware production, especially in lower volumes.

### Creating a new product: Phases 1 and 2 in more detail

For the purpose of this paper, we are most interested in ‘microcontroller (MCU) class’ devices, that typically combine an MCU with tens or hundreds of other electronic components via one or more printed circuit boards (PCBs). The resulting assembly is frequently housed in a plastic, metal or sometimes wooden enclosure, and distributed in some form of packaging. Examples include bicycle computers, child-friendly music players, USB-based computer accessories and hobby electronics products.

Figure 1 shows how phases 1 and 2 can be broken down into six stages necessary to turn an idea for a new type of MCU class device into a full-blown product. After ideation, the next step involves building one or more functional prototypes to validate feasibility. A single such proof-of-concept might encapsulate the entire functionality of product, but often several prototypes demonstrate different aspects of device operation and collectively provide evidence that the device concept is technically sound. At this point an engineering verification test (or engineering validation trial, EVT) is used to bring everything together in a single “looks-like, works-like” prototype. A handful of EVT units mark the end of Phase 1.

The focus of our research is Phase 2, which takes the first functional prototypes through a productization process. If successful, this ultimately results in the on-going availability of manufactured product. Although a working design exists on entry to Phase 2, a great deal more design engineering is required. The following description of the necessary steps is based on the authors’ personal experience combined with information from the few resources on the topic we are aware of [5,7,18,25].

The candidate design must be verified in anticipation of expected component tolerances and varying operating conditions. Processes of design for manufacturing, assembly, and test (DFM, DFA and DFT) are necessary to ensure that production is viable. A manufacturing test jig must be designed and built in conjunction with the DFM process so that functional tests can be performed on every board as they come off the production line. Packaging must be designed. Radio frequency (RF) and environmental testing are necessary as is regulatory pre-compliance testing. And any or all of these processes may result in iterations in the device's underlying mechanical or electronic design.

Before leaving the design and development phase, it is important to have a comprehensive product specification and well-documented quality and test requirements. Component availability, obsolescence risks and supply chain fragility must all be assessed, and capacity planning with manufacturing partners should be underway.

With all the above completed, the design validation test (or design verification trial, DVT) can begin. In the DVT stage, test procedures and jigs are finalized and deployed by the manufacturing partner, and production is run at a slower-than-normal speed so that yield and quality can be analyzed. Specific tests depend on the nature of the product, but all functional and aesthetic requirements of the product must be verified. Further iteration of the product design or manufacturing process may be necessary. Then the documentation, designs and firmware are frozen, and regulatory compliance certification is completed.

Having completed DVT, the product is ready for launch. This involves a product validation test (PVT) phase, which essentially replicates the requirements of DVT while ensuring that production is carried out at full line speed. This may require 'first time yield' optimization – a process of adjusting production to maximize output quality and throughput. Typically, after around 1000 PVT units have been successfully manufactured, the test and development phases are complete and production moves to the final phase.

This whistle-stop tour of the device productization process is by no means comprehensive, but none-the-less begins to reveal the complexity involved in transitioning from a working prototype to a viable product.

#### Prior work on Phase 1: building working prototypes

Popular development boards such as Arduino, Raspberry Pi and BeagleBone have become staple prototyping tools for interactive and embedded systems [13]. Microsoft's .NET Gadgeteer [34] extends the development board concept to include peripheral modules for communications, input, output and power supply.

Extensive software toolkits have also been built to facilitate quick iteration and testing of interactive devices. The Calder Toolkit [22] allowed builders to quickly iterate their designs by providing modular components that could be used in different configurations to interact with each other. D.tools

[16] is another toolkit allowing users to quickly prototype software functionality and integrate with modular hardware components for quick testing and design iteration. More advanced techniques such as using inkjet printers to quickly prototype functional electronic prototypes have also been shown to be effective [20]. Rapid design and prototyping of interactive products can be facilitated using augmented reality [27]. Similarly, DisplayObjects [1] uses 3D projections to add interactivity to common prototyping tools such as paper and cardboard.

Some prototyping tools target specific aspects of hardware development such as: analog circuit design [30], capacitive touch sensing [29] and FPGA-based systems [6]. Pinpoint is a more generic tool for debugging and analysis at the circuit design stage [31]. Other systems target specific applications such as factory automation applications [12] and shape-changing displays [15].

In summary, whilst challenges and opportunities in the design and prototyping phase inevitably still exist, in our opinion this is becoming a relatively mature field and no longer presents a bottleneck in device creation. Rather, we believe that the Phase 2 challenges of hardware production are limiting the adoption of new ideas.

#### Beyond the prototype: the challenges of Phase 2

As stated earlier, scaling a hardware prototype to hundreds or thousands of instances is surprisingly problematic. Part of the 'surprise' perhaps stems from the community's familiarity with software products – particular websites and apps – that can require relatively little resource to create and deploy beyond the creator's own time. However, when it comes to hardware, there are several important differences.

Firstly, it's important to recognize that software readily scales to thousands or even millions of users using low-cost digital distribution mechanisms. This is not true for hardware. Additionally, no hardware production process will create truly identical copies. This *replication challenge* is key to the difficulty of moving beyond a working prototype and scaling to a viable product [17]. Accommodating component tolerances, dealing with variability in manufacturing, sourcing consistent materials from suppliers, and several other factors conspire to cause subtle and not-so-subtle differences between each copy of a physical product. Controlling this variability requires the design, construction and use of custom tooling.

Another challenge with hardware relates to *iterative development* – leveraging feedback from early users to improve the product. For hardware products it is difficult for users to provide meaningful feedback early on when prototypes are relatively crude. But building a close-to-finished prototype requires many of the manufacturing specifications and processes to be finalized. At this point making improvements based on user feedback may require significant changes to the manufacturing tooling which incurs significant time and money.

In a similar vein, *updateability* is also a challenge in scaling hardware products. With software, remote updates allow developers to fix bugs, add new features, and address other shortcomings cheaply and quickly. Indeed, software sometimes ships with known deficiencies, in anticipation of software updates which will address them. However, there isn't a viable equivalent for hardware: updating hardware incurs dramatically higher costs because it requires physical handling of every affected unit – shipping, processing, re-work, re-testing etc.

There is a common theme among these challenges: the *monetary cost* associated with each of them is much higher for hardware than for software. The upfront NRE cost of a new hardware product is a major challenge: in addition to design of the product itself, it includes design for manufacturing/assembly/test, compliance testing, certification and tooling production. When all this is complete, the device production itself has a significant up-front cost in terms of raw materials procurement, factory set-up, warehousing and shipping.

In the next section we examine prior work to enable the creation of hardware devices at lower production volumes – something previously referred to as the “long tail of hardware” [2,4,17].

#### Prior work on Phase 2: device manufacturing

Unfortunately, there seems to be little research into the topic of low volume electronic device manufacturing reported in the literature. Indeed, the Journal of Electronics Manufacturing ceased publication in 2002! Hodges and Chen [17] describe how economies of scope – efficiencies gained across multiple products by virtue of commonality between them – may be leveraged for low volume device manufacture. There is also some evidence that the long tail of hardware exists, thanks to easy to use platforms such as Arduino: in [4] Buechley and Hill look at hobbyists and tinkerers who are building devices in the order of tens of units. Workshops at academic venues have further highlighted the need to consider the interplay between making and manufacturing [9–11,14].

In terms of research of technologies that support higher volume production, there is a plethora of research around *silicon fabrication*, such as [28] and [32] which consider the benefits of DFM in a silicon foundry. In particular, Tam and Blanton use failing IC diagnosis results to systematically evaluate the effectiveness of DFM rules [32]. However, we have not found equivalent research which expands on the trade-offs when designing for manufacturability of devices.

Tian et al. used lathes to demonstrate an improved fabrication experience via a richer set of haptic feedback that mediates a user's interaction with a fabrication tool [33]. Their work focuses on building a collaborative experience between the machine and users (novice and expert alike), but it also has the potential to bridge the gap between the design of fabrication tools for varying scales.

In terms of assembly, industry best practice includes automatic solder-paste inspection of a PCB prior to component attachment to ensure that the solder has been applied consistently, along with a subsequent automated optical inspection step to check correct component attachment [19]. However, to achieve acceptable yield, these inspection steps are typically complemented with functional testing: every assembled PCB is tested electrically using a custom-built “bed of nails” test jig. We are not aware of published research which evaluates or refines the standard industry approaches to designing test jig hardware, or the software which drives the jig to complete functional test. We are also unaware of any research which considers the co-design of an electronic device in parallel with design of the functional test procedure.

In the domain of woodworking, JigFab is an easy-to-use tool designed to empower a wide audience to build complex joints and objects with wood [23]. It does this by generating low-cost tooling, which in turn makes accurate fabrication much easier. We believe that a similar technique may be fruitful for building hardware test jigs.

In addition to an assembled PCB, many electronic devices require an enclosure. Injection molding is perhaps the most popular process for manufacturing high fidelity enclosures with good material properties, but the NRE process is expensive (up to US\$10k per piece for small electronic devices) and time consuming (around eight weeks). King and Tansey used selective laser sintering to generate mold geometry with bronze [21] instead of the typical machined steel approach, thereby reducing the lead time two weeks and also reducing cost. Rapid prototyping technologies can also be used to manufacture enclosures directly. Laser cutting is sometimes used for producing simple enclosures in low volumes, but low fidelity and poor aesthetics limit applications. In theory 3D printing is a good match for low-volume production, but in practice it is still expensive and material properties limit its utility.

In summary, we believe that Phase 2 manufacturing is a bottleneck for the success of new devices. To verify this, and to understand the relevant issues in more detail, we have taken an empirical approach which we describe in the rest of this paper.

## METHODS

### Pilot

We first developed a list of around ten questions designed to elicit the challenges of low-volume hardware production and approaches taken to overcome these. We then ran two pilot interviews with those involved in low-volume production. The interviewees discussed their pain points, corroborated aspects of our hypothesis that Phase 2 of manufacturing was still problematic, and provided valuable insights into a typical process (or lack of process!) that is followed by a hardware creator to manufacture devices. We quickly realized that each interviewee's journey and process would

be different and that specific questions would be unlikely to capture sufficient details for us to draw meaningful insights. Instead we selected the semi-structured interview process described in the following section.

### Study Procedure

Following our pilot, we conducted 25 semi-structured interviews to understand the common challenges associated with low-volume electronics manufacturing. The interviews were conducted in-person where possible but were primarily remote, leveraging video or audio teleconferencing. Each interview lasted an hour on average. Based on our experience with the pilot, we modified our interview protocol: we asked participants to walk us through their journey from conception to productization and only used our questions as probes to dive deeper into certain topics when required. The major themes of our interview questions were challenges faced in: (1) sourcing components; (2) finding a manufacturing partner; (3) designing for manufacturing; (4) testing the functionality of the product (5) certifications and regulations. We also questioned them on their personal motivation, tools used, and statistics about cost, pricing and profit. Other questions were tailored to each interviewee based on their journey as it unfolded.

### Participants

We divide our participants into two categories: *creators* and *enablers*.

We define creators as individuals who have undertaken low volume hardware device manufacturing either alone or as a part of a small team. Creators may include anyone from a single person running a crowdfunding campaign to the CEO of a startup. As such, creators have a variety of backgrounds: for example, they may or may not have any electronic engineering experience. For our purposes, the key characteristics of a creator are that they are personally invested in taking a hardware prototype into production and that they are part of a small team with relatively little productization experience.

On the other hand, enablers are individuals who assist creators in achieving their goal. They include (but are not limited to) individuals at hardware start-up incubators and accelerators, crowdfunding organizations and contract manufacturing companies. Each enabler typically interacts with many creators over time, and as such has built up knowledge of different stages of the manufacturing process. Some of the enablers that we interviewed were previously creators who subsequently found that their experience allowed them to move into an enabler role.

We used snowball sampling to recruit interviewees. We tried to balance interviewees in terms of creator vs enabler and by geographic location, see Table 1. Many of our creators were self-funded, but others had run successful crowd-funding campaigns and/or received venture capital of up to several million US\$. We found it hard to quantize and compare years of experience, but we had a mix of participants including

first-timers, repeat creators, newly turned enablers and veteran creators with over 20 years' experience.

**Table 1: Study participants by region, role and gender.**

	America	Europe	Asia
Creator	10 (9 M, 1F)	6 (4M, 2F)	2 (2M)
Enabler	5 (3M, 2F)	1 (1F)	1 (1F)

### Data collection and analysis

All interviews were audio recorded and later transcribed for data analysis. We then performed open coding on the transcribed audio interviews to identify recurring themes. Next we analyzed our codes using thematic analysis [3] to identify (1) common challenges and issues associated with low-volume manufacturing; and (2) any mitigation strategies adopted to overcome these issues. Finally, we used member-checking [8] with one enabler and one creator to validate our results.

### FINDINGS

Four major themes emerged from our analysis, discussed below. These themes span common challenges and key factors for successful manufacturing at scale. Note that across these themes we will only present findings that were corroborated by a significant number of participants.

Before we present the themes, we would like to set the context regarding scale. The enablers and creators we talked to consistently considered five thousand units as a transition point where economies of scale kick in and cost-effective electronic device production is much more comfortably achieved. It's not that the smaller volumes targeted by many creators were outright blockers – it just becomes much more important to find the right manufacturing partner:

*"The better the factory / the bigger the factory, the higher their MOQ [minimum order quantity]. 5000 is typically enough to get you into any factory. You can work with a smaller factory with less units" [C9]*

*"if you end up with a supplier who is too big, you won't have any chance to get the right deal in terms of amounts and price because the big supplier also has to finance the big organization behind it. And smaller suppliers are okay in the beginning..." [C5]*

The second quote above raises an important point: it's important to ensure that partners are operating at a scale which matches the creator. If a manufacturer is too small, whilst they may be responsive, they cannot necessarily deal with the required volume. Conversely, a large manufacturer may prioritize bigger contracts from other customers. Therefore, if a creator's business grows, it is necessary to change partners as the production volume increases.

### Theme 1: Gaps in technical knowledge

First-time creators typically commit to the productization process without the technical knowledge necessary to be successful. Many of them have a university degree in a

technical subject (e.g. computer science, mechanical engineering or electrical engineering), which likely helped them design a prototype. But none of our creators described having any formal education regarding the transition from design to manufacturing. They typically imagine that the process will be straightforward and that they can readily pick up any necessary skills along the way.

The upshot is that a novice creator may not even realize that there are important differences between making a few prototypes vs. replicating those prototypes in a manufacturing process, even at relatively low volumes. As one enabler we talked to explained:

*"There aren't many teachers. Maker Spaces don't teach this. You couldn't even go to a tech shop and learn any of this. It isn't there. Tech shop is for unit volume one. [...] They never teach you how to reproduce anything, even in low volume] and the constraints that come out of that. So you don't learn it in university. You don't learn it in pre-graduate education. A big company will teach you, but they may teach you over engineering it. So it falls on folks like me to teach that."* [E1]

Therefore a gap exists not only in **what** the creators know, but also in **how** they access resources to fill in the knowledge gaps. This appears to be in contrast with the software industry where a plethora of resources have evolved over time – blog articles, tools, design patterns, checklists, and even online video courses. Collectively these help a software creator obtain a comprehensive set of skills necessary for releasing a viable product. This lack of access to know-how leads to failures:

*"We, I think, made a mistake by going down the road of 'Let's put everything we want into this first shot at building' and that ended up being like a ten-layer board that was incredibly complex that really tried to do too much [...]. I think we would have been better served to isolate individual systems, put them together in a prototyping fashion and build 100 of them or 50 of them or something, rather than trying to go down our route of trying to turn out 5,000 or 10,000[...]. I think we would have learnt more and moved a lot faster if we had split those things out and just focused on individual ideas first."* [C11]

There have been a few efforts to make knowledge sharing with a wider audience more systematic (e.g., Hardware Studio<sup>1</sup> by Kickstarter), but our creators largely relied on picking up knowledge via word-of-mouth – either from more experienced creators or enablers who were willing to share their prior experiences, tips and tricks. This naturally results in quite subjective and somewhat patchy advice.

One frequent area of unknowns that we identified during our interviews was the topic of compliance and certification:

*"Let's say I want to do an FCC [Federal Communications Commission] certification, it's still like a big black box on*

*how to get that done as a small business, like finding out what's the rough estimate, rough investment, what are the options locally or in China."* [C12]

### Theme 2: Gaps in non-technical knowledge

In addition to technical skills, many other areas are critical to success. These span a wide range, including finance, marketing, distribution and customer service. A few particular themes we think are worth highlighting emerged in our interviews.

The issue most consistently raised by both enablers and creators was underestimating the cost and complexity of taking a hardware product to market. For example, here are reflections from two of our creators:

*"whatever [funding] you think you need to mass produce—you need more. You need way more. That is always the rule of the thumb. Nobody ever comes in under budget."* [C9]

*"[with our first product] we didn't lose any money because we had won a bunch of pitch competitions, so that helped us not lose anything other than time. In the case of [our second product], yeah, we were definitely in a negative."* [C10]

Underestimating costs across all activities relating to hardware manufacturing and supply frequently has the knock-on effect that a product is priced too low:

*"They [the creators] run a crowdfunding campaign and they price their product too low. ... They don't do a thorough calculation of all the cost required and basically price themselves out, so that they are never able to make it."* [C9]

Armed with this insight, perhaps it's not surprising to learn that around half of the creators we talked to hadn't made a profit from their products, in some cases after many years of operating. It was not uncommon for creators to provide their time for free, and in one case a creator explained how open source contributors had also spent their free time helping to develop software features for his product.

Finance wasn't the only non-technical topic raised by creators and enablers. To take another example, even creators aware of the technical requirements of compliance (see previous section) were typically surprised by the complexity of implementing them:

*"Okay, you have packaging and batteries and electronic waste, and you need registrations for that in each and every country. Even in Europe, where you think it should be harmonized, no, it isn't. You have to be registered in each and every country and each month you have to tell each country how much of the products you have imported."* [C5]

In addition to specific difficulties, the complexity of simultaneously managing the wide range of activities related to a hardware business was also apparent:

<sup>1</sup> <https://www.kickstarter.com/hardwarestudio>

*“The lack of knowledge is challenging, but the biggest challenge: imagine everything we’re talking about happening simultaneously. That’s the challenge of hardware”* [E1]

The gaps in non-technical knowledge are likely applicable to other domains beyond electronics manufacturing. And as such, there are many resources that can arm a creator with the required skills. Given this, it could seem surprising how often our participants were caught out by what is arguably ‘common knowledge’. However, many of the hardware creators we interviewed came from a software background, where these aspects of bringing a physical product to market are less relevant. Certainly our interviews indicated that a lack of non-technical knowledge was a common issue, sometimes leading to failure in the transition to production.

### Theme 3: Minimum viable rigor

Having discovered the common knowledge gaps of many creators, we also learned that even when creators do broadly know what they should do to address certain aspects of the replication challenge, they don’t always do them effectively! We repeatedly heard examples of creators making decisions which, with hindsight, they acknowledged did not follow best practice. In essence they did not apply the minimum level of rigor necessary to avoid manufacturing issues, resulting in avoidable delays and costs.

One example stems from the apparent ease of engaging a PCB manufacturing and assembly partner. Many online services provide a web-based interface for uploading design files and the bill of materials (BoM), and within a few days or weeks assembled circuit boards are shipped back. This remarkably low-friction process is great for prototyping. However, for production many of our creators followed the best practice described previously, requiring per-unit functional testing by the PCB assembly partner. This requires investment in a custom manufacturing test jig and associated firmware, designed in concert to verify key elements of the product’s functionality. If this is not rigorous enough, errors can be missed. One creator experienced this: an unauthorized and incorrect component substitution by the manufacturer was not detected until the whole batch of units had been shipped to the creator. At that point it was much more costly to resolve than it would have been in the factory.

A second pitfall relating to testing stems from lack of rigor in understanding and documenting the environment in which the product will be used. One enabler commented:

*“...they didn’t specify their product properly at all and they didn’t put the rigor in figuring out how they were going to test it and validate.”* [E1]

Another example of an ‘avoidable mistake’ comes from a creator who based their product design on a common off-the-shelf processor board, customizing it for their use. This ended up costing substantially more than expected.

A final example relates to a creator who didn’t check the availability of supply of a key component, the microcontroller, until late in the process:

*“we placed that order I think somewhere in January and it took like three months to get all of the ATmega chips.”* [C10]

Several enablers informed us that they typically maintain a checklist, or work with creators to maintain comprehensive specification documents to help avoid these (and other) common pitfalls. These documents usually involve technical specifications for all components, design files, supplier information, a BoM, testing procedures and more.

### Theme 4: Building relationships & professional networks

Perhaps not surprisingly, creators and enablers consistently described the importance of building relationships with a network of partners. The stronger these relationships become, the more value they typically provide back to the creator. For example, we heard about a PCB manufacturing partner who felt comfortable providing feedback to the board designer and how this had reduced cost for them:

*“They did that [to their PCB design] – shifted it all in by half an inch – and suddenly it was like \$12 rather than \$89 per board [to manufacture]. But they didn’t know that ahead of time.”* [E7]

The choice of a manufacturing partner and the strength of the relationship a creator builds with them can therefore make a big difference.

We noticed that creators used one of two quite different strategies when picking a manufacturing partner. The first set of creators valued physical proximity and were willing to accept a higher quoted cost in return for a partner that they would be able to visit often and communicate with easily. These creators anticipated that the benefits of proximity would outweigh the higher quote:

*“I had one rework problem where I sent the boards back to [local manufacturer anonymized] and it cost me \$40 to overnight the batch of boards to them and then they rework them and send them back to me. Got them back in two or three days from when they left here. If I had to do that with a Chinese manufacturer, just the customs back and forth would have added at least two weeks to that process, never mind the shipping.”* [C8]

Additionally, several of these creators raised the concern that working with a manufacturer located in another part of the world was risky. For example, the process of returning sub-standard or defective products may be costly or difficult.

The second set of creators optimized on the cost of materials and manufacturing instead of physical proximity, typically picking factories based in China. However, in most cases creators ultimately concluded that visiting the manufacturing partner in person and spending time on the factory floor was essential. Just as with a local manufacturer, seeing the production facility first hand and explaining the nature of the

product directly to the manufacturer helped them build a closer working relationship and ultimately address any issues that arose. Lindtner describes the Chinese culture of open manufacturing and how it differs from its western counterpart [24], further illustrating the value of in-person engagement. In fact, only one of our creators never visited their manufacturer in person, instead managing a remote collaboration without issues. One creator elaborated:

*“Okay, so definitely go to China and work there. That was one of the things I learnt ... because we did everything remotely and it took much longer.”* [C10]

No matter whether the manufacturing partner is local or remote, once a relationship is built, a creator can feel more confident in the manufacturing setup and do future manufacturing runs for the same product without much intervention. In fact, some creators enjoyed lower costs for future production runs when they leveraged an existing manufacturer relationship, presumably because manufacturing tooling such as test jigs was re-used.

Beyond manufacturing, creators often rely on a network of partners to help them with recommendations and introductions to various experts, component and equipment suppliers, or ancillary services. New creators rely on introductions from friends and family, LinkedIn connections, or the network of an enabler such as a consultant at an incubator.

In addition to a network of partners, we found that creators often also built professional network of peers – like-minded individuals that the creators would often meet at events of common interest such as Maker Faire and Teardown, or while working in maker spaces and tech shops. This network primarily appears to provide a community and sense of belonging. It also allows creators to seek advice, learn new skills, get additional recommendations and introductions, and sometimes even gain customers.

Lastly, creators valued building an early relationship with their user base. Creators and enablers with crowdfunding experience saw building a community of users as one of the fundamental tenets of a successful campaign. Other creators reported using forums or group messaging platforms like Slack and bulletin boards to actively engage their users for market research, beta testing and in fact, one of creators was even able to recruit individuals to contribute to the firmware development for free!

## OPPORTUNITIES

Having reflected on the challenges that surfaced during our interviews, we unearthed five opportunities for improving the status quo. Three of these emerge directly from the themes highlighted above, while two relate to the recurring issue of developing low volume production processes in a cost-effective and reliable way. These are summarized in Figure 2 and described in more detail below.

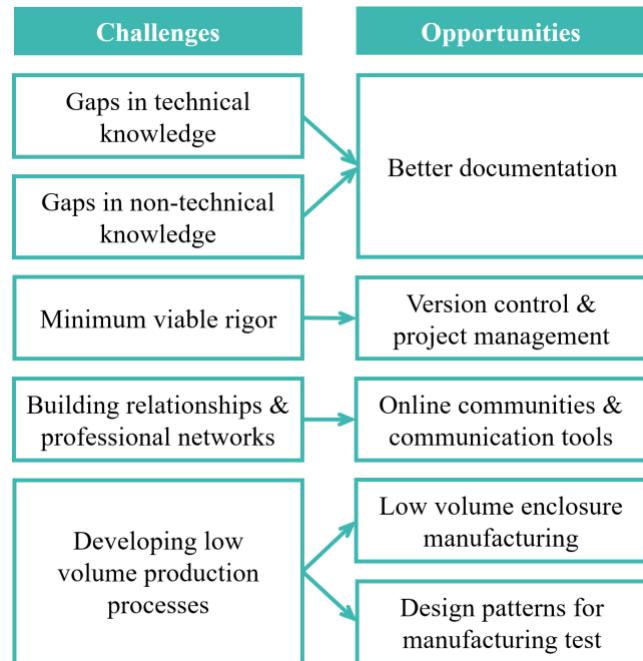


Figure 2: Overview of challenges identified in scaling hardware production and the opportunities to improve the process.

### Better documentation

The lack of both technical and non-technical knowledge clearly points towards a simple need for better documentation. One of the enablers we interviewed is currently writing a book that covers some of the relevant information, but right now there is a dearth of material. As another of the enablers we interviewed put it:

*“It is this knowledge you can’t buy. This is the thing. All of this stuff [how to design for manufacturing] needs to get written down in a set of bibles and it’s just not.”* [E1]

### Version control and project management

We observed that enablers typically have a defined process to organize and keep track of a creator’s progress and overall development status. One enabler leverages Github’s version control features for this – they ask creators working with them to document progress and place product specification documents in a repository. Another enabler did not mention a specific tool but used a similar process. However, traditional version control tools such as Github are developed with software products in mind and have limited support for hardware files other than simply storing them. For example, they don’t provide the ability to preview a native electronics design file or a Gerber file (the de facto standard for PCB images) or to examine differences between iterations. There is a need for targeted tools that support the most common processes used in hardware design.

Additionally, we observed that the lack of such software has an impact on the communication between the creator and the manufacturer. Creators often rely on e-mail exchanges to share files with a manufacturer, or in a few cases, a web link to upload files. Both of these approaches are susceptible to

human error. A version control system designed for hardware products, which allows creators to communicate, share and edit designs with their partners and stakeholders would improve the overall process.

#### **Online communities and communication tools**

Over the past decade or so the evolution of tools to support new software creators has dramatically lowered the bar for successfully commercializing a software product without prior knowledge or experience in the industry. In contrast, the professional network of the creator of a hardware product is currently dictated by geographic proximity to a hub of hardware startups and enthusiasts, or their willingness to regularly attend conferences and events in different parts of the world. We see this as an opportunity for a new social networking platform of sorts, designed to allow new creators to connect both with more experienced creators and with enablers. This would reduce the burden of establishing a professional network that creators typically have to bear before they can seek out help. We envision a future where hardware creators have ready access to a virtual community in the same way that software developers do today.

#### **Low volume enclosure manufacturing**

The difficulty of manufacturing enclosures at low volumes was mentioned by many. The most common production method is injection molding, but the initial cost of the necessary tooling (*i.e.* the molds) is typically US\$5-10k per piece and they are time-consuming to produce. This is too much for many low volume products.

*“Oh yeah, there’s a lot of people that can build circuit nowadays, but there’s many fewer people that can actually build a case to hold those electronics. So how do you make those processes easier?” [C1]*

Other plastics manufacturing processes exist, including vacuum forming, vacuum casting, roto-molding and laser-cutting, but none provide the fidelity, finish and material strengths of injection molding. For some applications 3D printing is the next-best choice, but despite steady improvement in quality, material properties and speed, our creators rarely found it cost effective. Enablers supported the view that 3D printing is not yet a viable substitute for injection molding for most applications. If a new technique could better bridge the gap it would likely have a significant impact. For example, LaserOrigami [26] uses a laser cutter in a defocused setting to heat and form plastic into different shapes, albeit with a limited amount of customization. An innovation such as this could potentially be valuable.

#### **Design patterns for manufacturing test**

A recurring theme in our interviews was testing. Creators sometimes define the test procedures for their product, but the actual “bed-of-nails” jig is often designed and created by the manufacturing partner. Indeed, a surprising number of creators had never even seen the jig itself! This lack of visibility has several side-effects. Firstly, unless the creator was rigorous in specifying their test procedure, the jig is

unlikely to provide complete test coverage. Secondly, the cost of developing the test jig is often hidden – amortized into per-unit production costs in an ill-defined way. And thirdly, the creator can become locked-in to the manufacturing partner because they feel reliant on their established test procedure.

Whilst there has been prior work to in the area of testing for electronics design [31], to our knowledge there is a gap in work on testing for electronics production. As our study progressed, we began to consider the possibility of creating a set of open source test jig “reference designs” – hardware and firmware templates or design patterns that creators could readily modify to suit their particular testing requirements.

#### **CONCLUSION & FUTURE WORK**

We started the research reported in this paper knowing that in the past decade it has become much easier to design and build a prototype of a hardware device. This is due to the proliferation of research, commercial products and online communities which support Phase 1. But scaling from a prototype to the production of hundreds or thousands of units is still challenging – an observation borne out by our study. We learned that many factors that were not relevant at the prototype stage conspire to create this challenge. As a result, creators who have not had first-hand experience transitioning to manufacturing have not encountered them. But even seasoned creators who know the pitfalls struggle to manufacture cost effectively at low volumes due to up front production costs and reliance on other partners – difficulties not typically faced for software-only products. The experience of low-volume creators contrasts with large companies manufacturing devices in high volumes, typically ten thousand units a year and above, who can leverage well-established manufacturing processes and partnerships to build high quality products at a reasonable cost. Our study highlighted that the threshold between ‘low’ and ‘high’ volume is typically around five thousand units a year; below this the economies of scale necessary to make electronics goods viable are hard to achieve.

Collectively, the creators and enablers that we interviewed have amassed a great deal of knowledge concerning the constraints of low volume manufacturing. Whilst some of the issues they reported may seem straightforward to the uninitiated, in reality they are significant pitfalls which persist today. The four themes which emerged from our interviews naturally point to a number of opportunities to improve the status quo for low volume hardware production as shown in Figure 2. There is a clear need for better documentation and other education mechanisms covering both technical and non-technical information. The little information that currently exists is so patchy, not only is it hard to find but it’s also hard to put into context. In addition to documentation, the hardware development community would also benefit from a stronger online presence to share experiences and professional contacts. Finally, we see opportunities for new tools and resources which reduce the

cost and/or friction associated with key activities in the transition from prototype to product. Examples include better hardware version control, project management, enclosure manufacturing and manufacturing test. We appeal to the community to join us as we continue our research in this domain.

Our vision is a world where low volume manufacture is commonplace. Rather than relying on the economics of scale enjoyed in high volume manufacturing, we believe that innovations like those outlined above can provide sufficient economies of scope to make small batch manufacture more efficient and thereby cost effective. This would empower both experienced creators and relative novices to turn their prototype interactive and embedded devices into viable products. Ultimately, we hope that reducing the friction and cost associated with low volume manufacture will allow more hardware ideas to become reality, thereby driving innovation and increasing consumer choice.

#### ACKNOWLEDGEMENTS

We would like to thank Nicholas Chen for his guidance and support throughout this project, Anja Thieme for her help navigating the various necessary project review processes, and Anna Thornton and our reviewers for their feedback and suggestions on earlier revisions of the manuscript.

#### REFERENCES

- [1] Eric Akaoka, Tim Ginn, and Roel Vertegaal. 2010. DisplayObjects: prototyping functional physical interfaces on 3D styrofoam, paper or cardboard models. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, ACM, 49–56.
- [2] Chris Anderson. 2006. *The long tail: Why the future of business is selling less of more*. Hachette Books.
- [3] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qual. Res. Psychol.* 3, 2 (2006), 77–101.
- [4] Leah Buechley and Benjamin Mako Hill. 2010. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM conference on designing interactive systems*, ACM, 199–207.
- [5] Elaine Chen. 2016. *Bringing a Hardware Product to Market: Navigating the Wild Ride from Concept to Mass Production*. CreateSpace Independent Publishing Platform.
- [6] John Clow, Georgios Tzimpragos, Deeksha Dangwal, Sammy Guo, Joseph McMahan, and Timothy Sherwood. 2017. A pythonic approach for rapid hardware prototyping and instrumentation. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, 1–7.
- [7] Alan Cohen. 2015. *Prototype to product: a practical guide for getting to market*. O'Reilly Media, Inc.
- [8] Susanna Doyle. 2007. Member checking with older women: A framework for negotiating meaning. *Health Care Women Int.* 28, 10 (2007), 888–908.
- [9] Verena Fuchsberger, Silvia Lindtner, Martin Murer, and Manfred Tscheligi. 2015. *Rethinking Technology Innovation: Factories, Fabrication & Design Research*. (2015).
- [10] Verena Fuchsberger, Martin Murer, Manfred Tscheligi, Silvia Lindtner, Shaowen Bardzell, Jeffrey Bardzell, Andreas Reiter, and Pernille Bjorn. 2016. Fabrication & HCI: Hobbyist Making, Industrial Production, and Beyond. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 3550–3557.
- [11] Verena Fuchsberger, Martin Murer, Manfred Tscheligi, Silvia Lindtner, Andreas Reiter, Shaowen Bardzell, Jeffrey Bardzell, and Pernille Björn. 2015. The Future of Making: Where Industrial and Personal Fabrication Meet. *Aarhus Ser. Hum. Centered Comput.* 1, 1 (2015), 4.
- [12] Franco Fummi, Mirko Loghi, Stefano Martini, Marco Monguzzi, Giovanni Perbellini, and Massimo Poncino. 2005. Virtual hardware prototyping through timed hardware-software co-simulation. In *Proceedings of the conference on Design, Automation and Test in Europe-Volume 2*, IEEE Computer Society, 798–803.
- [13] Ahmad Adamu Galadima. 2014. Arduino as a learning tool. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, 1–4.
- [14] David Philip Green, Verena Fuchsberger, David Kirk, Nick Taylor, David Chatting, Janis Lena Meissner, Martin Murer, Manfred Tscheligi, Silvia Lindtner, Pernille Bjorn, and others. 2017. Open Design at the Intersection of Making and Manufacturing. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 542–549.
- [15] John Hardy, Christian Weichel, Faisal Taher, John Vidler, and Jason Alexander. 2015. Shapeclip: towards rapid prototyping with shape-changing displays for designers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 19–28.
- [16] Björn Hartmann, Scott R Klemmer, Michael Bernstein, Leith Abdulla, Brandon Burr, Avi Robinson-Mosher, and Jennifer Gee. 2006. Reflective physical

- prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM, 299–308.
- [17] Steve Hodges and Nicholas Chen. 2019. Long tail hardware: Turning device concepts into viable low volume products. *IEEE Pervasive Computing* 18.
- [18] Andrew Bunnie Huang. 2019. *The Hardware Hacker: Adventures in Making and Breaking Hardware*. No Starch Press.
- [19] Mihály Janóczki, Ákos Becker, László Jakab, Richárd Gráf, and Tibor Takács. 2013. Automatic Optical Inspection of Soldering. In *Materials Science-Advanced Topics*. IntechOpen.
- [20] Yoshihiro Kawahara, Steve Hodges, Benjamin S Cook, Cheng Zhang, and Gregory D Abowd. 2013. Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, ACM, 363–372.
- [21] D King and T Tansey. 2003. Rapid tooling: selective laser sintering injection tooling. *J. Mater. Process. Technol.* 132, 1–3 (2003), 42–48.
- [22] Johnny C Lee, Daniel Avrahami, Scott E Hudson, Jodi Forlizzi, Paul H Dietz, and Darren Leigh. 2004. The calder toolkit: wired and wireless components for rapidly prototyping interactive devices. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, ACM, 167–175.
- [23] Danny Leen, Tom Veuskens, Kris Luyten, and Raf Ramakers. 2019. JigFab: Computational Fabrication of Constraints to Facilitate Woodworking with Power Tools. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 156.
- [24] Silvia Lindtner. 2015. Hacking with Chinese characteristics: The promises of the maker movement against China’s manufacturing culture. *Sci. Technol. Hum. Values* 40, 5 (2015), 854–879.
- [25] Richard Marshall, Lawrence Archard, and Steve Hodges. 2019. IoT Hardware from Prototype to Production. Retrieved January 1, 2020 from aka.ms/proto-to-product
- [26] Stefanie Mueller, Bastian Kruck, and Patrick Baudisch. 2013. LaserOrigami: laser-cutting 3D objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2585–2592.
- [27] Tek-Jin Nam. 2005. Sketch-based rapid prototyping platform for hardware-software integrated interactive products. In *CHI’05 extended abstracts on Human factors in computing systems*, ACM, 1689–1692.
- [28] Mark Redford, Joseph Sawicki, Prasad Subramaniam, Cliff Hou, Yervant Zorian, and Kimon Michaels. 2009. DFM: don’t care or competitive weapon? In *Proceedings of the 46th Annual Design Automation Conference*, ACM, 296–297.
- [29] Valkyrie Savage, Xiaohan Zhang, and Björn Hartmann. 2012. Midas: fabricating custom capacitive touch sensors to prototype interactive objects. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, ACM, 579–588.
- [30] Evan Strasnick, Maneesh Agrawala, and Sean Follmer. 2017. Scanalog: Interactive design and debugging of analog circuits with programmable hardware. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, ACM, 321–330.
- [31] Evan Strasnick, Sean Follmer, and Maneesh Agrawala. 2019. Pinpoint: A PCB Debugging Pipeline Using Interruptible Routing and Instrumentation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ACM, 48.
- [32] Wing Chiu Tam and Shawn Blanton. 2011. To DFM or not to DFM? In *Proceedings of the 48th Design Automation Conference*, ACM, 65–70.
- [33] Rundong Tian, Vedant Saran, Mareike Kritzler, Florian Michahelles, and Eric Paulos. 2019. Turn-by-Wire: Computationally Mediated Physical Fabrication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, ACM, 713–725.
- [34] Nicolas Villar, James Scott, Steve Hodges, Kerry Hammil, and Colin Miller. 2012. .NET Gadgeteer: A platform for custom devices. In *International Conference on Pervasive Computing*, Springer, 216–233.