PROTOTYPING: BUILD

Learning Objectives

After reading this brief, you will be able to…

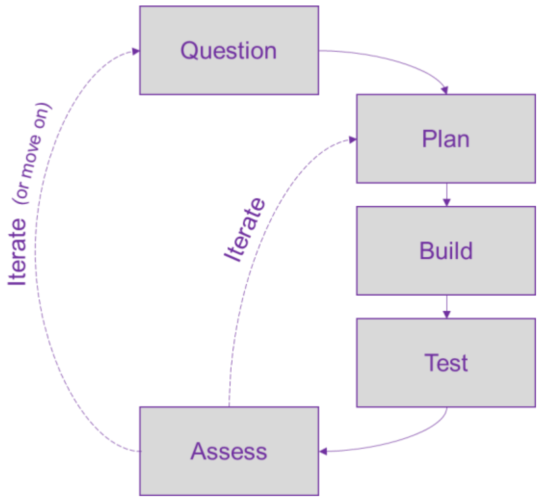
I. Explain how progressive prototyping, rather than an "all at once" approach, is key to making rapid progress.
II. Describe how different types of prototypes (looks-like, feels-like, works-like, is-like) can be used to address different project questions.
III. Consider relevant materials, tools/processes, and expertise when creating different prototypes.
IV. Explain in which instances to build versus buy versus borrow.
V. Describe the importance of design documentation in early prototyping.

Getting Started

The build stage of the six-step prototyping process shown in Figure 1 is when a project starts to become more concrete. This is because building and testing prototypes – whether physical devices, apps, workflows, or other innovations – fundamentally requires the specification of quantitative implementation details.

Teams can talk abstractly about different technology designs/sizes/shapes all day, but when it's time to get real data you must choose which one(s) you'll actually build and test. This makes sense if your prototype is intended to investigate different shapes by building them with representative materials and testing their functional properties. And it's just as true if your prototype is in a computer-aided design (CAD) program that will be shared with users to get their preferences, since you'll have to set specific details to get coherent data. Indeed, even if your prototype is not a physical device at all – maybe it's a mathematical model or a simulated workflow – you still need to choose specific operating parameters or ranges to investigate.

Figure 1 – Six-Step Prototyping Process: Question-Plan-Build-Test-Assess-Iterate



Once you've defined meaningful prototyping questions and created actionable plans, you're ready to move into building

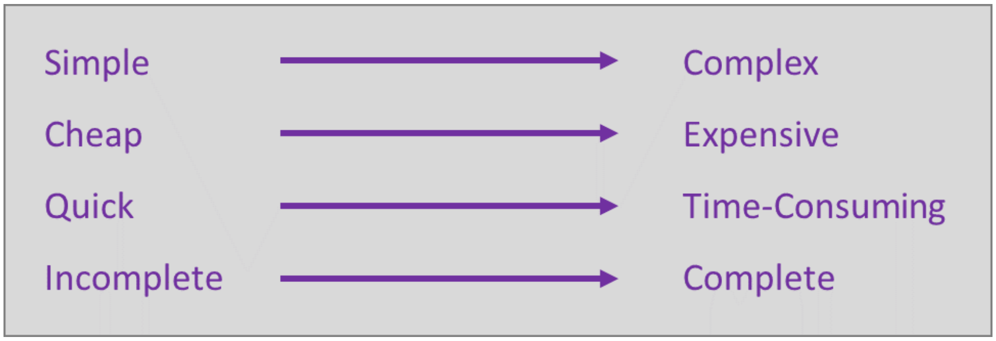
There are three driving topics that can help you establish relevant details for building: 1) what to build; 2) how to (actually) build it, and 3) how the build can enable testing, assessment, and iteration. Each of these is covered in the sections that follow.

What to Build

Progression of Prototypes

Prototyping is fundamentally iterative and progressive, and prototyping builds should be this way too. Early prototypes should be simple, quick, cheap, and incomplete. Future versions can add complexity or expense, with commensurate increases in time and completeness (see Figure 1).

Figure 1 – Prototyping Progressions



The creation of early prototypes should be efficient, with more time and resources invested as the project progresses.

The most common mistake novice prototypers often make is attempting to build the end-to-end concept/system from the beginning, rather than progressively building and testing from single parts up to a gradually more complete system. This is often in an attempt to rapidly see if the whole concept works. While this approach is understandable, teams can make quicker progress by building only what is necessary to address one or two narrow questions at a time, and then using the resulting learnings to advance the team’s understanding and build project momentum.

For example, if you want to make an assembly with some off-the-shelf components and a novel 3D-printed part, the first step is not to buy all of the parts and head straight to the 3D printer. The first version(s) should be hand-drawn sketches, which you can use to efficiently play around with different options while appreciating practical details that may not be obvious (e.g., How many screws to use? Will the screws connect via nuts or do we need threaded holes?). Once those first sketches make sense, a good next step might be to make a model with a combination of cardboard, tape, and a subset of available components. Then, it might be time to proceed to a CAD model (pro tip: find CAD versions of the off-the-shelf components on supplier webpages). Finally, depending on the size and constraints of the 3D printer, you might make some scaled-up or scaled-down versions before getting to the “right-sized” 3D printed model. While this might seem like more steps than necessary, the smaller steps go quickly, and only experienced builders in a particular medium can anticipate which steps can be skipped without having to backtrack.

Expand the section below for another example. And watch the videos called Miraclefeet Brace: Prototyping Approach , Miraclefeet Brace: Progression of Prototypes, and Miraclefeet Brace: Low Resolution Prototypes for more relevant information.

Progressive Prototyping Example: RF Ablation

Imagine a team that is working on a new minimally-invasive surgical tool that uses radiofrequency (RF) energy to ablate a specific nerve. A big-picture question that immediately arises is whether there will be “collateral damage” to nearby anatomical structures that makes the procedure too risky. While it could be tempting to try to jump right into an animal study to investigate the effectiveness and side-effects of ablating that nerve – and it's a good idea to think about this as a future study – there are several intermediate questions that potentially should be addressed first.

For example:

- Is the team sure about the size and shape of the delivery tool in providing access and an appropriate angle of approach to the nerve? Maybe a quick-and-dirty looks-like model, made out of pipe-cleaners and with a couple of anatomical variations, would be helpful to answer these questions.
- Does the team know how to titrate local power deposition near the nerve? If not, they might proceed with the animal study only to realize that they we were not adequately controlling the power, rendering the work inconclusive. Perhaps a quick study using meat from the butcher shop could provide enough relevant data to inform titration methods.
- Do they have a way to know when the ablation is "enough" to turn off the power? If not, how will the team know when to turn off the RF unit? Here again, a simple study with butcher-store meat may be necessary or effective.

In all of these cases, early/intermediate prototyping work would help confirm details that are crucial for the success of later, more complex prototyping work.

As the examples illustrate, progressive prototyping complements the six-step prototyping approach in that it enables teams to most efficiently address their key questions in a logical, stepwise manner. By keeping each prototype focused on an essential questions, you'll simultaneously develop and evolve the most relevant aspects of your concept and retire the project's biggest technical risks.

Types of Prototypes

Establishing what type of prototype your testing will require is the foundation of a successful build. There are four primary types/categories prototypes you can consider, organized from least to most burdensome:

- Looks-like
- Feels-like
- Works-like
- Is-like

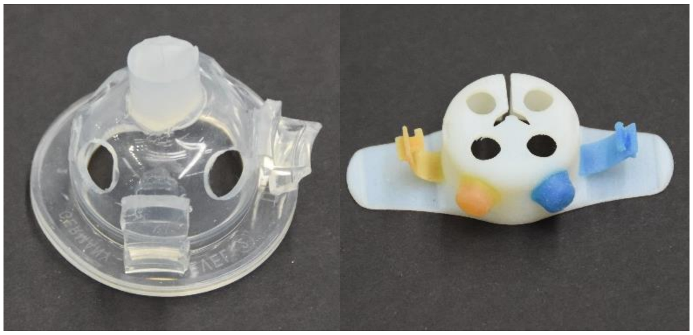
Looks-like Prototypes: Quick and Underappreciated

The first prototypes to make for any project should be looks-like prototypes. These are the simplest and quickest to make, and they help set up future prototypes for success. Some people wonder, at first, what benefit there could possibly be to making, for example, a looks-like version of an electromechanical device with construction paper, twist ties, and duct tape. While it may feel like a flashback to kindergarten, there's significant value in physically building and holding the first 3D version of any design.

These simple models often provide insights on the physical relationships between parts, or allow you to identify forgotten elements. Often times, a looks-like prototype can enable rapid iteration of different design elements, enabling a design choice before committing to more functional (read: expensive and time-consuming) physical prototypes. Some looks-like prototypes are even sufficient to share with potential users and to support data-driven decisions (about user interface design, sizing, etc.).

For example, while developing the LifeBubble umbilical catheter securement device for babies in the neonatal intensive care unit (NICU), the Novonate team (introduced in the Prototyping: Question and Plan toolkit) realized that they needed to understand how nurses and doctors would feel about the different sizes and geometries of their potential designs. They knew their final design would need to be made of a soft, sterilizable silicone in order for it to be safely attached to the belly of a newborn baby. However, making different silicone molds for each design was prohibitively expensive (in terms of both time and money). To address key questions and drive design decisions, as well as facilitate early interviews and comparison testing with care providers, the team developed a variety of hand-made prototypes (out of baby bottle nipples) and CAD models (which they could efficiently 3D print at different scales and with different materials) to bring to the hospital. Figure 2 includes two of the early looks-like prototypes that the team shared to get qualitative and quantitative design feedback from NICU nurses.

Figure 2 – Novonate Looks-Like Prototypes



Left: An early design concept made from modified silicone baby bottle nipple. Right: A 3D-printed design concept with larger holes, side holders, and different colors (courtesy of Novonate).

Challenge yourself to see which of your key questions could be addressed through a looks-like prototype, and if you can get measurable data through a looks-like model. If you're still skeptical, take it as a challenge **not** to learn something with your first looks-like prototype, and then have some fun making and sharing it with your team.

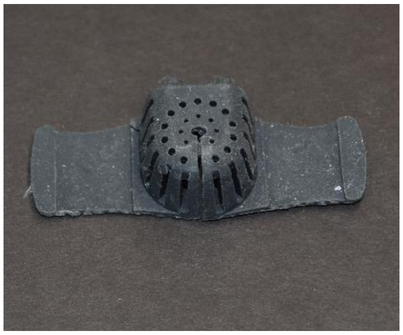
Feels-like Prototypes: Delving into Human Factors

Feels-like prototypes are somewhere between looks-like and works-like prototypes in terms of the effort required to build them, as well as what they do for the project. They must do more than just look the part – they also must enable some measure of user interaction with what the design concept will feel like to use. Yet, they're typically easier to create than prototypes that actually function.

Feels-like prototypes are particularly well-suited to evaluating assumptions or performance related to usability and human factors. This could be an evaluation of the weight, balance, or size of a surgical tool manipulation method. Or, it could be about the experience of using particular data analysis tools as part of a user interface within an app. As always, your goal in developing a feels-like model should be to generate meaningful data that helps you answer a key project question. Teams sometimes struggle to make the data they get from feels-like prototypes quantitative – see the toolkit called Prototyping: Test, Assess, and Iterate (coming soon!) for more information.

Note that while a feels-like prototype does not need to be a looks-like prototype, too, the appearance can definitely matter when users interact with the model. The Novonate team encountered this when using a soft material for 3D printing prototype shapes that would eventually be made of clear silicone. The prototype had a more accurate 3D shape than other models, and it had the right durometer (stiffness), but it was entirely opaque due to limitations with the 3D printing material (see Figure 3). Unfortunately, NICU nurses had a hard time evaluating the feel of the device without getting distracted by the fact that it was not see-through, which limited the usefulness of the feels-like prototype.

Figure 3 – Novonate Feels-Like Prototype



With this 3D-printed design concept the opaque material was chosen for its flexibility, not for design intent. Early user feedback about opacity and color helped to direct design choices as well as future user interviews (courtesy of Novonate).

Works-like Prototypes: Function Over Form

Works-like prototypes are those that have sufficient functionality, in at least one aspect of the design, to enable some manner of functional testing. Importantly, this means that a works-like prototype *does not need to be fully functional*. Indeed, most works-like prototypes barely work, and they might only work in one specific way and under limited conditions. This lean and targeted approach enables teams to make efficient progress. If it works enough to generate useful data that addresses a key question, then it's a works-like prototype.

The breadth of works-like prototypes is vast, encompassing not only the range of possible projects, but also the different individual mechanisms and functions of the subsystems of each project. Such prototypes can be as simple as a single component with one function (e.g., able to test if a handle of a certain shape and material will break under a certain applied force) or as complicated as a multiple-component set of integrated subsystems that accomplish a variety of tasks (e.g., an adjustable, multi-stage optical setup, integrated with a microcontroller-compatible camera and data pipeline that can assess the achievable resolution of a simplified microscopy setup). They can be large or small, physical or virtual.

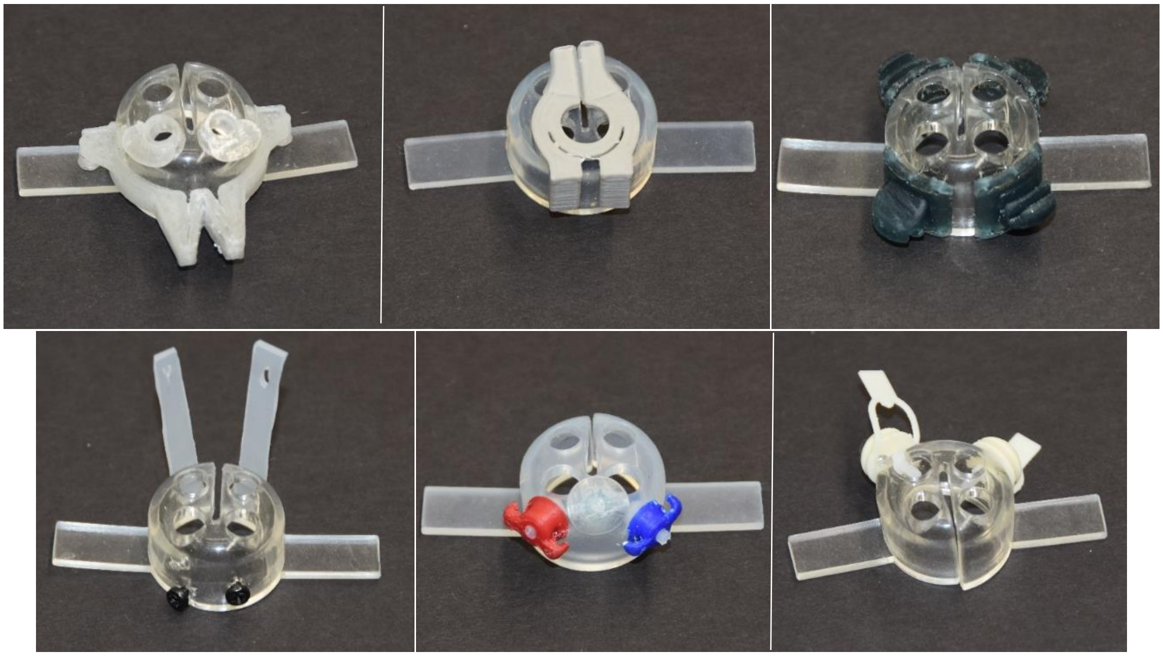
> “Everything should be made as simple as possible,
> but no simpler.”
>
> -Albert Einstein

The goal for works-like prototypes is to build them as simple as possible, but no simpler (as Albert Einstein is considered to have said[1]). Early works-like prototypes typically perform single functions in isolation, before moving on to works-like versions that integrate multiple types of functionality. Early versions also tend not to look or feel much like the final design – they might have different sizes, shapes, weights, etc. in order to enable targeted testing of the underlying function.

Sometimes, a single prototype can embody combinations of looks-like, feels-like, and works-like characteristics, but this is not necessary. Attempting to make a prototype has all three goals can over-constrain the building or testing processes.

To test a variety of different candidate mechanisms for securing an umbilical catheter, the Novonate team made a series of works-like prototypes. They were rough, made from a range of materials, and not particularly attractive. Crucially, they worked well enough to undergo simulated ease-of-use testing for initial catheter securement and removal. Some also worked well enough to undergo mechanism challenge testing via a team-designed pull-force rig (learn more about this in the Prototyping: Test, Assess, and Iterate toolkit – coming soon!). Figure 4 of shows a few of these prototypes, each with different securement mechanisms. All of them were built using a common molded silicone base.

Figure 4 – Novonate Works-Like Prototypes



Examples of works-like prototypes of various mechanisms for umbilical catheter securement. The designs are attached in different ways to an identical injection-molded clear silicone base. The goal was to identify and develop quantitative evidence to support a mechanism (pinching, wrapping, strapping, etc.) that was easy to use and could withstand relevant pull forces on a secured catheter (courtesy of Novonate).

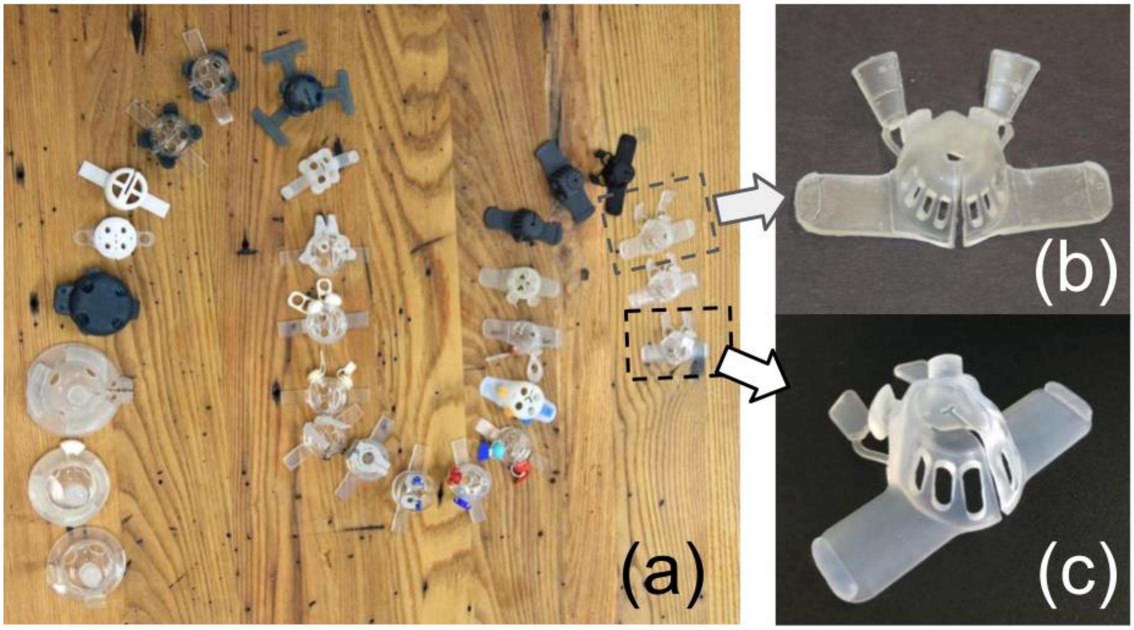
Is-like Prototypes: The Flag on a Hill

Is-like prototypes don't just look, feel, or work like the concept your users will experience. They're enough like the product that your users should be able to actually use them. They're integrated, with fully functional subsystems. They're made with sufficient-quality components. And they function indistinguishably from the final product.

Notably, it’s typically not the right time to make an is-like prototype until well after you think (or hope!) it is time to make one. Such a prototype might be the first one that will be used in humans. Or it might integrate all of the features that you have investigated separately through different works-like prototypes. It’s the proverbial “flag on a hill” – the destination that the team sees way off in the distance. You should aim for it, while appreciating that there are likely several other hills to climb – prototypes to build, test, and assess – along the way.

The Novonate team built its first is-like prototype after locking in all of the major parameters for the LifeBubble design, securing enough funding to work with a prototyping house, and finding a shop with the right combination of materials, tools, and expertise to machine and operate the mold. This prototype (see Figure 5) integrated nearly all of the key features of the LifeBubble design, with the notable exception of the clarity/translucency of the top window, which the team believed could be accomplished through additional mold polishing steps. (Note: Multiple manufacturing shops initially told the team the mold could not be made due to its complexity and fine features. The is-like prototype enabled Novonate to move forward with a contract manufacturer that was initially hesitant.)

Figure 5 – Novonate Is-Like Prototypes



(a) A series of selected prototypes constructed by the Novonate team, in rough chronological order from left to right. The three devices on the far right are is-like prototypes. The preceding prototypes are a combination of looks-like, feels-like, and works-like models. (b) This prototype, from the first milled metal mold, includes nearly all key design features (note: top window is translucent, but not transparent). (c) The production-quality LifeBubble, including final design changes to vent hole number/size and a transparent top window (courtesy of Novonate).

Expand the section below for a summary table that compares and contrasts the capabilities, advantages, and disadvantages of looks-like, feels-like, works-like, and is-like prototypes. And watch the video called and The Novonate Project Using Prototyping to Retire an Early Stage Risk for more information about prototyping on the Novonate project.

Table 1 – A Comparison of Types of Prototypes

	Primary Values/ Capabilities	Relative Advantage(s)	Relative Disadvantage(s)
Looks-like	Exploration: Enables quick modifications of design concept, without burden of required functionality Communication tool (within the team and with mentors): Gets everyone on the same page about the concept; provides common artifact for critique, questions, and modifications/improvements Communication tool (with potential users): Enables user feedback on specific features of a concept, enables user comparisons between different concepts	Cost- and time-efficient to produce	Limited measurable data, due to explicit lack of functionality
Feels-like	Communication tool (with users): When users get a feel for how things will work, the actual working part is often unnecessary for them to provide feedback on the experience that can shape redesign goals or inspire new features User empathy tool (within the team and with mentors): Enables everyone to have a common user-like experience Exploration: Enables quick modifications of design concept, without burden of required functionality	It is often more cost- and time-efficient to produce a feels-like than a works-like prototype, yet you can sometimes get similar data from users	Limited measurable data, due to explicit lack of functionality Data can be qualitative (often requires a second step to quantify feedback)
Works-like	Functionality: It has a working aspect that enables quantitative investigation of a key project question or risk	Functionality, without integration, enables testing of isolated performance aspects, without the	Applicability of the test data is potentially limited or subject to change, depending on how

		need to achieve all functions in a single test (e.g., it does not necessarily need to be fully miniaturized, or use the exact materials, or interface with adjacent mechanisms)	isolated or limited the function is. Working mechanisms are typically more difficult and costly than looks-like or feels-like prototypes.
Is-like	Integration: Enables verification testing of required functionality across integrated subsystems Validation: Enables full-stack, "in vivo" testing with users – gets to the underlying question of "does it really work?"	There are typically few technology-related questions once an is-like prototype is working	Typically difficult and expensive to make and/or test Confounding factors and/or integration issues, rather than fundamental issues, can lead to negative results

After Choosing the Type of Prototype

Beyond establishing whether you'll construct a looks-like, feels-like, works-like, or is-like model (or a hybrid), the following questions can be helpful to consider before you start building:

- What is the minimum fidelity the prototype needs in order to address the question?
    - If it's looks-like, does it need to be nicely painted, or does it need to be painted at all? Will a 2D drawing, a CAD model, or 3D object be needed?
    - If it's feels-like, does it need to have representative weight and shape, or just representative shape?
    - If it is works-like, does the mechanism need to both grab the tissue and drive the needle, or could we hold the tissue another way for now and just make the needle driving part work for this test?
- How many prototypes do you need to make to address the question?
    - Can we make just one prototype? Or do we need replicates of a single design? Or do we need to make several different versions to compare different designs?
    - Will the testing be destructive (intentionally or unintentionally), hence requiring multiple prototypes?
- How do I know that I have built the prototype successfully?
    - Is it built well enough that the testing results will reflect on the design rather than on the quality of the build?
    - What is the minimum build quality that can still yield a useful test result?

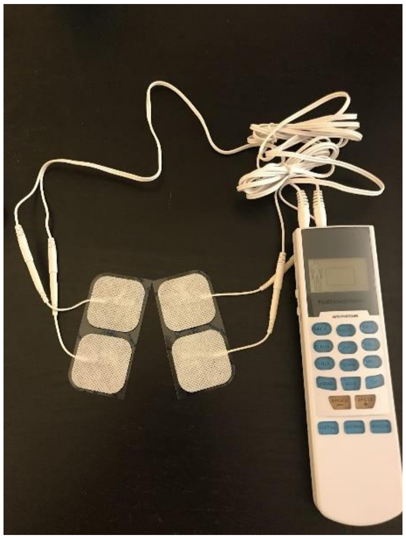
How to (Actually) Build It

Each project's progression of prototype builds is unique, yet the types of resources and how to approach them are common across projects. Whether the prototype you're building is part of the design/concept that will address the team's need statement, or whether it's a prototype of the physiologically relevant test setup that you will use to measure performance of different designs, there are three categories of building resources to consider: materials, tools/processes, and expertise.

Which Aspects (if any) Do We Need To Build From Scratch?

Before getting into the specifics of materials, tools/processes, and expertise, a useful meta-question to consider is how much/little you actually need to build, with your own two hands, to make progress. Sometimes the best prototype is one that someone else has already built. It might not be perfect for your application, but it might be "good enough" to investigate key parameters or to establish proof-of-concept. And, it's frequently faster and cheaper than designing and building your own. Buying or borrowing an existing technology also reduces the amount of time and effort that the team spends "reinventing the wheel," enabling you to focus more effort on your project-specific questions.

For example, teams working on neurostimulation often find that Transcutaneous Electrical Nerve Stimulation (TENS) units, which are over-the-counter, external stimulators, are an incredibly fast and relatively inexpensive way to get a (somewhat) controllable stimulator.



An example of an off-the-shelf TENS unit (as photographed by Stanford Biodesign).

Balancing When to Build versus When to Buy versus When to Borrow

While it's often a good idea to use existing products or devices to accelerate early prototyping, it's also possible to miss important learning or innovation opportunities by relying on the prior engineering of others. How to balance the value of efficiency with the value of learning will depend on the specifics of each project. Your team's key prototyping questions can likely provide useful guidance.

In the neurostimulator example above, the build versus buy balance might shift depending on how much of the project risk is in the technological aspects versus how much risk is in the physiological aspects. If prior studies have already established the mechanism of stimulation and the key challenge is in miniaturizing or reducing power consumption, then earlier efforts might require the team to build circuits rather than buying an off-the-shelf stimulator. If, on the other hand, the miniaturization seems like what other companies have done, but the effectiveness of tuning the therapy is unproven, the team might buy/borrow an existing stimulation system first to be able to get to a clinical effect test more quickly. Typically, measurement tools within your test apparatus (pressure or force gauges, chemical analysis tools, spectrometers, etc.) will be better to buy/borrow, especially if you need both precision and accuracy in your measurements.

Keep in mind that experts in a particular field are often aware of which technological capabilities are better to buy (inexpensive yet effective options exist for your questions) versus borrow (inexpensive or self-made options are unlikely to be effective for your questions) versus build (self-made options are achievable or required, due to the unique circumstance, to address your questions). Experts can also often connect you with resources to borrow, or with other people who have even more specific expertise relevant to your project.

Building Resources: Materials

Materials are the various pieces, parts, odds-and-ends, and found items – physical or virtual – from which you'll construct your prototypes. Anything you build will require materials, from stock supplies (planks of wood, sheets of paper/plastic/metal, rubber tubing) to fasteners (screws, nails, staples, bolts) to electronic components (resistors, breadboards, microcontrollers, motors).

Materials can refer to raw physical objects (acrylic sheets), or manufactured parts (machine screws), or more complex components and assemblies (geared motors, Arduino microcontrollers, syringes). For setups that mimic tissues, materials will include moldable silicones, gelatins, or even tissue samples. In many cases, software code or apps can act as a sort of material for your prototyping (MATLAB for a mathematical model, Excel for a database, Python for a real-time data processing and display environment, Google Forms for a survey).

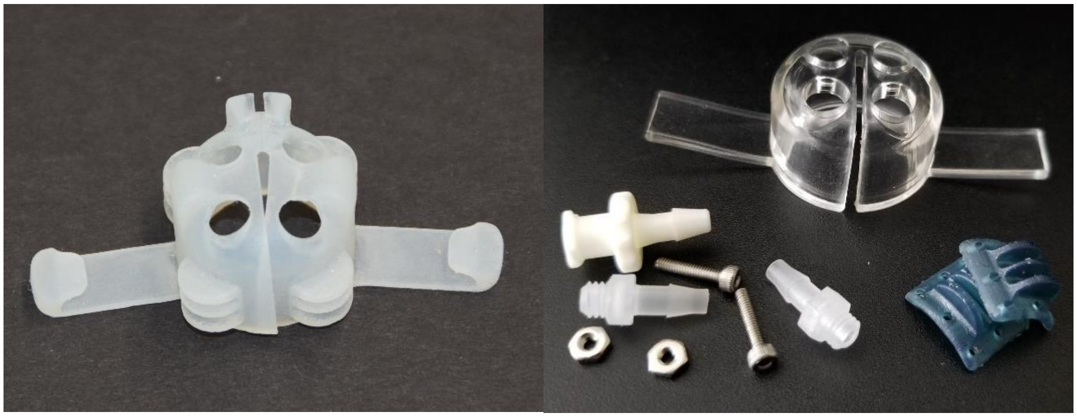
Materials have distinctive properties – either imbued by nature or developed as a technology – that make them more or less suitable for different uses and applications. Finding the right material can dramatically impact the feasibility of a design. That said, early in a project, it's a good idea to be flexible about what materials to use. From a practical perspective, what seems to be the ideal material may not be readily available. Additionally, what the team imagines at first to be the right material may have fundamental, as-yet-unknown challenges (brittleness, cost, manufacturability). In most circumstances, getting started with the materials that are at-hand is usually the right approach, proceeding with the assumption that a future prototype will require more suitable supplies. Sometimes, available and easy-to-shape materials (cardboard, foam core, wood) end up being good enough, especially for an early version of a prototype. And, even if the material ends up not being workable, the early experience with a non-ideal material will inform or confirm what material properties the design requires for a future iteration.

As the design progresses, using well-suited materials is more important (and can be critical for some projects). While practical barriers for specialized materials are evident (cost and lead time), another common issue is simple lack of awareness. This applies to raw materials as well as to existing, purchasable components. If you have an initial experience with an inappropriate material or component and you make a list of the ideal or workable properties that a suitable material would possess, a mentor or expert can often recommend something you wouldn't have known about. Similarly, browsing or searching catalogs from material and component suppliers, such as McMaster-Carr (general parts), can often yield good options. For medical parts, specialty suppliers like Qosina have helpful catalogs. And many teams have found eBay or other reseller sites to be a good source of expired (read: cheaper) medical products. (Pro tip: Canadian eBay sometimes has more options or available products.)

Considering materials as a distinct resource, and working to understand quantitatively what a design or prototype requires (lengths, thicknesses, durometers, frictional coefficients, elasticities, durabilities, temperature coefficients), will help you progress from early, crude prototypes to more functional later prototypes. Putting the material selection and ordering process in parallel with initial prototype building, and perhaps making material choice a specific question for a prototyping cycle, will help you avoid getting stuck waiting for that back-ordered specialty materials to arrive.

Going back to the Novonate example, the team sought a way to make prototypes out of silicone because silicone materials have a unique set of properties. While there were some cheaper and quicker ways to make silicone-molded parts – for instance, using simple 3D-printed molds – the results were suboptimal. They discovered that the combination of material characteristics and design specifications would require a precision mold, which would involve a long lead time and significant expense. Taking this into account, the Novonate team opted to make works-like prototypes through two parallel mechanisms: injection molding of an end-to-end design and hand-constructing features on a silicone-molded base. They contracted with different prototyping houses for these two purposes. The initial end-to-end silicone prototypes ended up being feels-like, since the securement mechanisms did not function. The silicone base ended up enabling a "Mr. Potato Head" approach for investigating and testing many different working mechanisms (se Figure 6). Learn more about this approach in the video called The Novonate Project: Efficiently Prototyping Several Design Ideas.

Figure 6 – Novonate: Different Materials for Different Types of Prototypes



Left: A fully-molded silicone prototype with several non-functional design features – the silicone was too floppy or thin-walled and was only translucent. Right: The silicone base with higher durometer and optical clarity, along with various connectors and 3D-printed parts for experimenting with different functional securement mechanisms.

Building Resources: Tools and Processes

Tools and processes are the means through which teams build materials into testable prototypes. When considering tools and processes for prototyping, there are a few distinctions worth keeping in mind.

- Tools versus Processes
- Knowledge versus Skill
- Creative Usage versus Ill-Advised Misuse of Tools

Tools vs. Processes

In the prototyping context, "tool" refers to physical equipment, while "process" refers to an action that the tool accomplishes. Tools and processes depend on one another, with know-how, experience, and best practices guiding their effective combination to create useful outcomes. It's helpful to distinguish between the tools and the processes in order to enable flexibility, especially in constrained circumstances, such as early prototyping, when time and money are tight.

By not assuming a particular tool when considering a process or prototyping goal, prototypers can find creative ways to accomplish their objectives with fewer constraints. The process of fastening provides a good example. Many students think first of a hammer and nails when seeking to connect two pieces of stock material,[2] yet there are many potential tools and processes for fastening, including brads, machine screws, self-driving screws, all manner of epoxies and glues, tapes, rubber bands, stitching, etc.

Another example of confusing tool with process involves approaching every physical prototyping task with the use of a 3D printer. While additive manufacturing via 3D printing is a useful way to construct a 3D shape, there are other effective processes (and associated tools) for constructing 3D objects from stock materials, including laser cutting mating parts for assembly, connecting pieces of stock via fastening hardware, etc.[3]

Expand the section below for some illustrative examples of tools and processes from different technology areas.

Table 2 – Examples of Tools and Process from Different Technology Areas

	Tools	Processes
Machine or electronics lab	Might range from simple hand tools (scissors, calipers, soldering iron) to more complex machining or test measurement equipment (lathe, CNC mill, laser cutter, 3D printer, injection molding machine, oscilloscope, spectrum analyzer)	In a mechanical lab, might include cutting, additive manufacturing, assembling, shaping, measuring In an electronics lab, might include soldering, producing reference signals, measuring time waveforms, performing frequency-domain analysis
Biological or chemical lab	Might range from simple lab equipment (centrifuge, pipet, incubator, fume hood) to more complex equipment (gel boxes, spectrophotometers, microscopes, chromatography columns) to more abstract tools (restriction endonucleases, promoter sequences, reporter genes)	Might include titration, centrifugation, passaging of cells, growing up cultures, running gels, specific assays
Software context	Might range from coding support systems (debugging/ development environments, online repositories), to 3rd party apps (wireframing apps, plug-and-play frameworks), to more complex stacks (cloud databases with multiple layers, chains of apps with various APIs)	Might include debugging, data processing/storage/retrieval, user interfacing

Knowledge versus Skills

Knowledge of different tools and processes, especially including their unique capabilities and constraints, helps prototypers develop options for how to accomplish their goals.

The more quantitative that knowledge is, the more helpful it is. For example, knowing that the minimum resolution of most laser cutters is ~1mm can help a team design a production process that will work – or it can rule out attempting to create a cut with ~10 micron thickness.

Sometimes simple awareness that a tool or process exists can open important possibilities for a team. For example, many oscilloscopes have internal computation capabilities to display frequency-domain data, so you might be able to get real-time feedback during operation/testing instead of a more cumbersome plan to analyze the frequency-domain data offline.

Skills – the ability to use different tools and processes – are the next-level enabler for building. While skills are related to knowledge, they are a distinct resource. In other words, knowing about a tool or process is often necessary yet insufficient preparation for the skillful use of that tool or process. New users of tools and processes should assume that some hands-on experience will be required before they have sufficient skill to be effective with prototyping. Teams should be prepared to learn by doing, and they should seek help from a more experienced user for their first attempts.

It's worth noting that people with skills are often harder to find than people with knowledge. And people who have skills may not have experience helping others learn that skill. A team developing a new skill can often benefit from the model in medicine of "see one, do one, teach one," in which one team member gets help from a mentor, then tries it on their own, and then helps another team member to learn. See the section on expertise below for a discussion of particular types of prototyping experience to seek.

Creative Usage vs. Ill-Advised Misuse of Tools

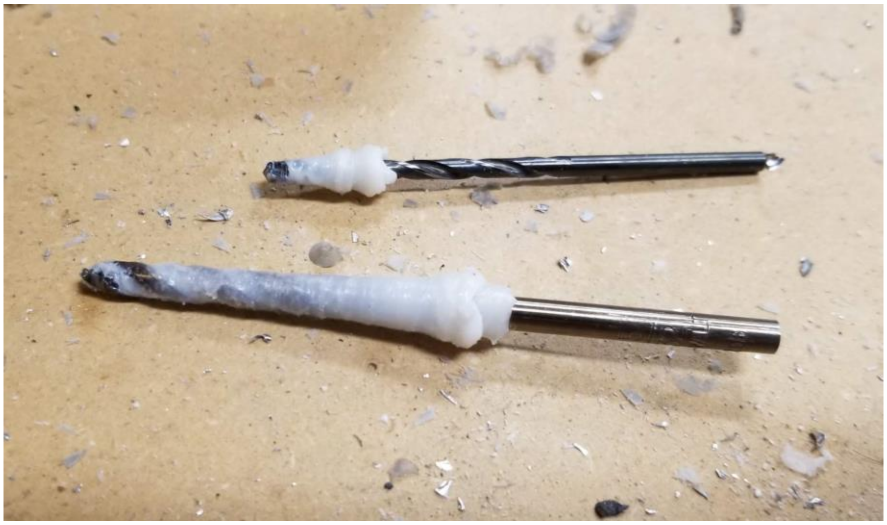
Teams often find themselves without access to the tools they desire, or without the knowledge or experience of the best tools for the job. In this circumstance, they sometimes devise creative ways to use the tools they know/have to accomplish the task at hand. Surely, creativity is a virtue, especially during early prototyping. Yet, when it comes to using tools, it's possible to be too creative. The overly creative use of tools is often counter-productive, and it can be both destructive of the tools and dangerous to the users.

The notes below can help you balance creativity with tool effectiveness and safety. While these examples focus on mechanical tools, the principles apply across all prototyping media – including physical, biological, chemical, and digital/virtual.

- **Fit for purpose**: Tools have been designed for particular use cases and they're fit for those purposes. They're typically unfit for purposes for which they were not designed. For example:
    - Screwdrivers can be used to pry things open, or as a cutting tool, or as a chisel – but they are a poor choice for those tasks. They're intentionally straight, so they don't provide a lever for prying. They're intentionally dull, so they're not good for cutting. And their handles aren't built to be hammered, so they break when chiseling.
    - Using a razor blade to turn a flat-head screw might sort-of work, but it's a bad idea in several ways. Aside from being likely to result in a laceration, it's both likely to yield a poor screw strength and to break the tool.
- **Specific tools for specific materials**: Different materials will shape or machine differently across processes, and toolmakers create different tools or settings to deal with this. It is worthwhile to find the tool that works and/or to seek explicit instruction in the combination of materials and tools to establish a workable method. For example:

- Making holes in different materials requires different tools and/or actions. Different speeds are needed for drilling aluminum versus steel. Different drills/bits work better for wood versus metal versus plastic.
- A common mistake in the lab is to attempt to drill a simple hole in a plastic 3D printed or laser-cut part using a drill press or hand drill in the same way you would cut through wood. The result is often a ruined drill bit – gummed up or totally encased in plastic (see Figure 7). This happens because drill tools/bits are designed to excavate chips of material from the hole to the surface using the drill flutes. Yet, drilling operations also generate heat, which quickly melts small plastic chips and prevents them from excavating, leading to a hot mess of melted plastic.

Figure 7 – Illustrative Results from the Misuse of Tools



Drill bits used to create holes in 3D-printed plastic parts.

Fortunately, there's good alignment between using a tool for its intended purpose, using it effectively, and using it safely. And a little bit of knowledge can enable novice users to be safe and effective in their tool use, applying most of their creativity to keeping alignment between their questions/risks and their prototyping activities.

Building Resources: Expertise

It's almost guaranteed that your team lacks important experience with at least one of the relevant aspects of your project prototyping, so you should plan to reach out for help early and often. Prototyping takes a village. Thankfully, most prototypers tend to enjoy sharing the pitfalls, frustrations, and workarounds they have encountered with others who are working on interesting projects.

Advice About Getting Advice

Beyond looking for someone kind enough to help you, something to consider is that there are different aspects of expertise that are relevant to building. Among these are design-specific expertise, stage-specific expertise, and level of engagement. When trying to get advice, the more specific you can be about what kind of help you need, the more likely you are to find the person who can provide that assistance.

- **Design-specific expertise** is about the materials, tools, and processes that are directly relevant to the design at hand. Ideally, you want to talk with someone who has used the same combination of materials and tools/processes that you plan to use. And it's even better if they've tried to make something nearly equivalent to your design. If the resource you find does not have such well-matched experience, of course, they can still help you accelerate your progress. You should just be prepared to extrapolate or interpolate their advice.
- **Stage-specific expertise** is about whether their experience is in early prototyping versus product development versus manufacturing. Depending on which risks/questions you hope to address with your build, one or the other of these types of expertise may be more useful for you and your team. Manufacturing expertise is precious if you're trying to figure out how to make something inexpensive at volume, dial in specific properties, or to make >10 identical units. But those insights may not be the right fit for helping you build a quick proof of principle, or for cobbling together a device for early animal testing.
- **Engagement level expertise** refers to whether someone has experience as a hands-on builder versus having direct but hands-off involvement versus indirect involvement in relevant projects. It's likely that the hands-on person is typically the most helpful when it comes to the details of how to build things. However, those with hands-off or indirect involvement may still be able to help identify obstacles or otherwise connect with someone who has the right hands-on experience. And, those folks may have a combination of experience and objectivity that enables them to help your team identify different challenges or opportunities that the hands-on expert could overlook.

Building Your Own Expertise

Keep in mind that you're not only building a prototype, you're simultaneously leveling-up the skills and experiences of you and your teammates. Having a solid prototyping approach can help make up for a lack of building experience. And, as an old proverb says, "the best place to find a helping hand is at the end of your own arm." By being deliberate with an iterative and question-driven prototyping process, you can build your way into your own expertise.

> "The best place to find a helping hand is at the end of your own arm."
>
> -Unknown

How Build Enables Test, Assess, and Iterate

By following the prototyping process, you probably have a good idea of how you'll test your prototype once it is ready. Still, it's worth pausing your build periodically to double-check that what you're building will work with your planned testing, as well as to document the in-progress design.

Aligning Build with Test

Thinking about testing during the building phase might lead a team to modify their build in a variety of ways. For example, they might:

- Add a feature to the design in CAD that allows direct connection of a measurement tool to the prototype (e.g., via a simple screw or fastener).
- Make a larger/smaller-scale version of the prototype, because the measurement tools on hand will better accommodate a different size (as long as it still addresses the underlying question at a different size without compromising the results).
- Reformat the questions in a survey to acquire both quantitative and qualitative responses from participants. Or incorporating a calibrated survey instrument (a survey that someone else has developed and validated for a similar purpose).

Often teams realize that they need to go through a whole question-plan-build-test-assess cycle to create an appropriate testing apparatus or method. Thankfully, when the prototype has been selected and developed to address key project questions, test development will align with those questions and will justify the effort.

The Case for Interim Design Documentation

Skipping documentation in the name of efficiency is a common error among novice prototypers (and some experienced prototypers). As important as moving quickly and efficiently is for your build process, documenting your prototype build is even more important. While it may seem counterintuitive to spend time documenting an interim design that's sure to change with the next iteration, the iterative nature of the process is part of what makes documentation critical. Here are a few reasons why it's so important:

- Without a record of each design iteration, it's difficult to keep track of which features enhance versus degrade performance and can lead to downstream failures.
- Without reasonably clear design documentation, teams can struggle to build adequate replicates for testing. Inadequate replicates degrade test performance and confidence in test results.
- Effective team collaboration on iterations, as well as work with mentors/experts, requires communication about specific design features which, in turn, requires sharable design documentation/artifacts.
- Builders often discover (and sometimes can fix) mistakes or bugs through the process of documenting.

Design documentation doesn't have to be exhaustive (or exhausting), but it does need to capture the key design features and, in some cases, the building methods.

For mechanical builds, a few simple annotated photographs, with some relevant arrows, a good caption and a parts/materials list, can be effective. Pairing this with a CAD image and a link to the underlying CAD can complete the documentation.

For electronics builds, the circuit diagram (with labeled parts/values) is absolutely essential, and it is often sufficient with a short caption or explanation of the circuit function. Pairing this with a block diagram and/or parts list can complete the build documentation.

For projects with code or software, including apps, microcontrollers, mathematical models, etc., revision control and in-line comments for the code are essential. Brief descriptions of the software architecture, in addition to links to sharable code (e.g., via github) and brief change logs, can complete rough build documentation.

For projects with biological or chemical aspects, documentation of the protocols and methods is essential. This includes keeping track of any modifications of steps or deviations. Sources of reagents, and other such details, can yield enabling documentation.

Creating a shared document with a simple template for your team to use when documenting your builds will help you to record your interim designs without forgetting important details.

Key Building Challenge: Scope Creep

"Scope" is an abstract term of art that engineers (and others) use to define the boundaries of a project. Activities or features that the team agrees are truly necessary for project success are within the scope of the project ("in scope") and the team needs to work on them. Any activities or features beyond necessity, are, by default, outside the project scope ("out of scope") and the team shouldn't work on those.

Scope creep is a term that engineers and project managers use to describe when a project scope gradually changes to encompass more (or different) work than the team initially planned. This happens naturally over the course of a project, especially during the build stage. It can be healthy and productive, as team members learn more about what it will take to make things work and adjust accordingly. But it can be unproductive (or counterproductive) if it happens due to loss of focus on the project goals.

Especially for early-stage prototypes, a primary challenge is how to focus the building effort on those few aspects that truly matter for the prototyping question(s) at hand. Teams must strike a balance between over-building (using more difficult/expensive/time-consuming materials than are necessary, or including unnecessary features) and under-building (making a prototype that lacks sufficient functionality or robustness such that it will fail to yield useful test results). Striking this balance is what engineering is all about, and it often requires some trial-and-error or iteration, during which the team can gain useful insights. Thinking through the testing and assessment plans ahead of time, and creating an explicit test plan that traces from each test to the design specification it seeks to investigate, will often save teams valuable iteration cycles on this issue.

BOX 1
Things Never Heard in the Prototyping Lab

"I wish I hadn't taken all of those pictures of our build."

"I sure am glad that I didn't record which parts I was using while debugging our circuit. I love retracing my steps to figure out what I already did."

"Commenting those lines of code was a huge waste of time. And I wish my teammate's code also didn't have all of these distracting comments."

"If only I hadn't recorded the batch of reagents and the dilutions that I used last time."

Getting Unstuck

The following are questions or ideas that have been helpful in getting teams unstuck when building:

- Take a field trip to a local hardware store; browse the aisles with specific materials or mechanisms in mind.
- Visit a local maker space and talk to an expert/trainer for a specific tool or process.
- Ask yourselves:
  - Can scaling up or scaling down help?

- Which aspects can we simplify? Which aspects need to be at full complexity?
- Are there aspects we can leave out entirely from this version?
- What can we build (assuming we're not presently able to build exactly what we'd like, due to some practical constraint)?

See Box 1 and watch the videos called Advice for Rapid Prototyping and Miraclefeet Brace: Prototyping Advice for more tips from Biodesign students.

Credits

Ross Venook and Alex Bradfield prepared this brief with assistance from Lyn Denend. We'd also like to thank Eric Chehab, Ian Connolly, Kate Buellesbach, Craig Milroy, Scott Piper, and Bryce Yao for their assistance with the videos.

Notes

[1] Andrew Robinson, "Did Einstein Really Say That?" Nature.com, April 30, 2018, https://www.nature.com/articles/d41586-018-05004-4 (July 23, 2021).

[2] This might relate to the Law of the Instrument, which many hear in the form of the common adage that, "When you're holding a hammer, everything starts to look like a nail."

[3] See Prototyping Library: Physical Objects To Teach Best Practices In Making, http://prototypinglibrary.com/laser-cutter/ for a useful resource for using a laser cutter to construct different types of 3D objects (Credit: Matthew Wettergreen, Rice University, Oshman Engineering Design Kitchen).