# PROTOTYPING: TEST, ASSESS, AND ITERATE

## Learning Objectives

As a result of reading this brief, you will be able to…

I. Explain the distinct yet interrelated roles of testing and assessment in the iterative prototyping process
II. Explain the importance of metrics in evaluating and retiring project risks, as well as advancing iterative prototyping efforts
III. Define a metric and relevant verification test to assess a health technology prototype for a relevant functional or design specification
IV. Define a metric and relevant validation test to assess a health technology prototype for a relevant user requirement or need criterion
V. Apply the "testing checklist" to identify and correct missing or incomplete elements in a basic test set-up and protocol
VI. Identify different types of potential assessment outcomes and determine appropriate implications/next steps

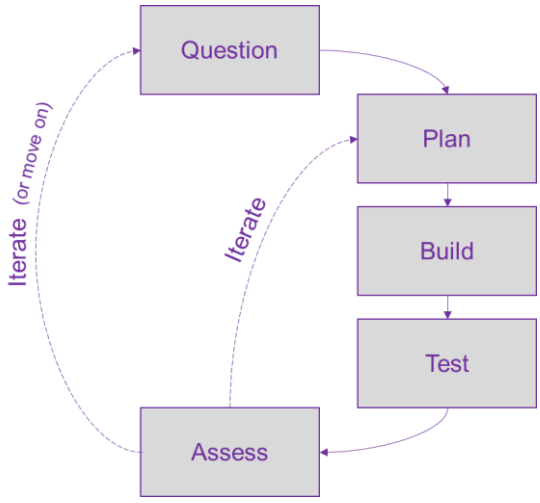
### Getting Started

When you reach the test and assess stages of the six-step prototyping process shown in Figure 1, you're ready to put your prototype to work! This is where you'll try to answer the initial question(s) with convincing data/analysis. If the results are sufficiently trustworthy and relevant, then perhaps the team can shift focus to the project's next guiding question. Alternately, if the results recommend improvements to either the prototype or the test method (or to the initial question), then it is time to iterate. Either way, you'll have productive learning that helps advance your project.

### Testing versus Assessing

Although these terms are used interchangeably in some contexts, it's valuable to clarify the meaning of "testing" and "assessing" in the context of iterative prototyping and technology innovation efforts.

**Figure 1 – The Six Step Prototyping Process**




After establishing a question, making a plan, and building a prototype (or a set of prototypes), it's time to test (make measurements) and assess (make decisions based on your test results). Depending on what you learn, you will iterate by creating a new plan around the same question or by moving on to a new question.

- **Testing**
  The gathering of measurement data to support project development/risk reduction efforts, using a relevant set-up with appropriate measurement tools.

- **Assessing**
  The analysis of the measurement data (from testing) to make decisions about project progress and/or direction.

As you can see, testing and assessing are inextricably linked. Useful assessments rely upon solid data that's generated from meaningful tests; and the purpose of testing is to enable high-quality assessments to drive project decisions.

Although most testing and assessing may be related to the technical aspects of a project, these parts of the prototyping process can cover a wider range of different risk areas. For example, user acceptance risk can be as important as technical risks (or more!). Running early tests with users to understand their perspectives – perhaps via surveys or experience testing with looks-like, feels-like, or works-like prototypes – can help answer early project questions about user adoption and/or compliance (see Digital Prototyping Example-Lower Back Pain and Electromechanical Prototyping Example-Opioid Addiction for related student experiences).

The keys to successful testing and assessing are the quality and relevance of the test methods and data: what you measure (metrics), the way you make measurements (methods), and how well you measure them (quality).

### Testing Step #1: Establishing Metrics – Choosing What Data You Need

Quantitative metrics, which refer to the combination of measured data and analysis of that data for usefulness in the context of the project, are at the heart of testing and assessing. Establishing numerical values for performance is essential, yet teams often try to avoid doing it. They stop short because it's hard work and often requires making uncertain assumptions that can feel uncomfortable. However, there are two basic rationale for why it's worthwhile to push for as much quantification as possible.

First, while early prototyping efforts can have generative or exploratory value, it's easy to "spin your wheels" with endless prototypes if you don't quantitatively evaluate them against objective metrics. The measurement of prototype performance against metrics is where the proverbial "rubber meets the road," and many teams feel like this is how/when the project really starts to move toward its goal.

Second, advancing the project toward a solution that actually works (and delivers value to patients) requires a concrete definition of what it means to "work," along with a method to measure the performance of potential embodiments sufficiently well to inform iterative improvements. Unless you can put numbers to what "working" means, you probably don't yet understand the problem or proposed solution well enough. Pushing to quantify can focus you on key gaps in your understanding.

#### What is a Metric?

Metrics are quantities of interest to the key project questions, ideally with direct relevance to either user requirements or system specifications. Stated another way, metrics are proxies for the functional success of a design or prototype. Sometimes a metric can be directly contained in a single measurement (e.g., how long does it take for the temperature at the target spot to exceed X °C, with a goal: within 3 minutes?). More often, however, they are based on an analysis that

combines multiple data points or types of measurements (e.g., what is the average of the 3-minute integrals of the maximum-heating temperature spot, across multiple trials? And does it exceed the goal temperature by more than the standard deviation of the measurements?). Note: There are often many metrics options for a given function or project question, with some being more useful than others.

### What Constitutes a Useful Metric?

The usefulness of a metric depends on its alignment with what the project needs to do, or what a specific prototype needs to do (or both).

- Project usefulness = metrics aligned with your project goals/need criteria
- Prototype usefulness = metrics aligned with learning about the solution concept

The usefulness of a metric also depends on whether or not you can measure it in a meaningful way, which requires a test environment/context and adequate measurement equipment/tools to capture and record the data.

Metrics are most useful if they can be used to guide decisions about the direction or progress or your project or prototype. Some examples of the types of decisions that metrics can drive during assessment include:

- Can we make a ranking or choose the best candidate among multiple design options? Can we evaluate the effect of tweaks in the design using the metric?
- Can we rule out specific designs based on experimental data?
- Is the prototype working well enough to proceed to more refined/rigorous testing?
- Have we established proof of concept to a level that's sufficient for the potential funders of our next steps?

### How to Create Metrics

Think about defining or creating metrics as a design process that you can execute in iterative cycles (like prototyping).

- **Step 1 - Need Criteria**
  Use your project's list of need criteria to think about what your solution must do to satisfy the needs of your most important stakeholder(s) (see the toolkit called Making Your Research Actionable via Need Criteria). Since the need criteria are solution-agnostic, this typically requires translation or interpretation in the context of your design. Using an example focused on ablating tissue,

**Page Views = Useful Metric?**

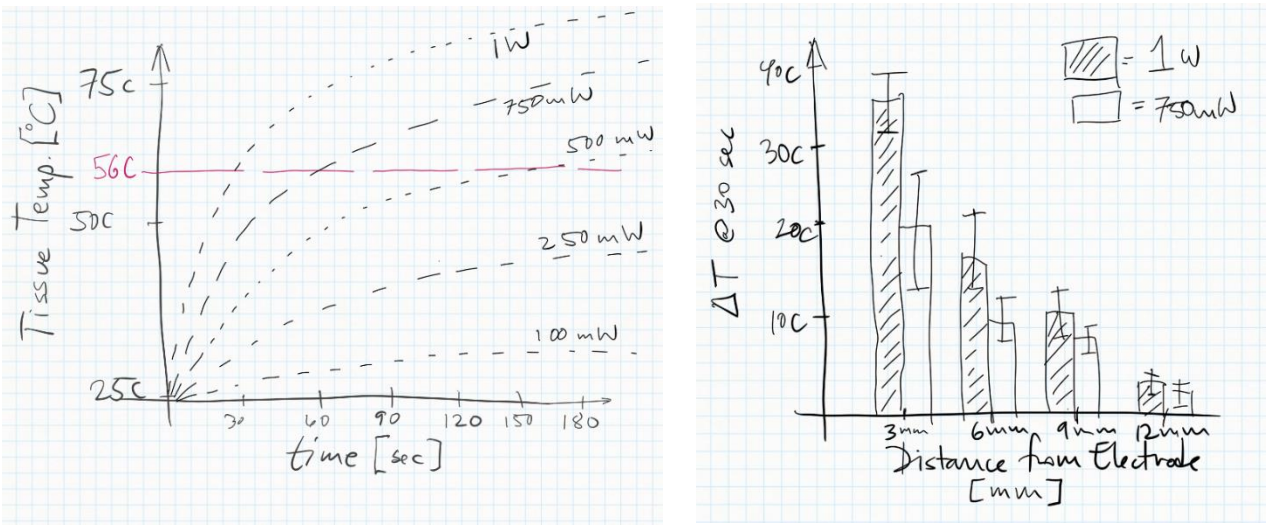
Consider a project focused on building a digital health website. One common measure of website performance is number of "page views," so one might think this is an interesting/-useful metric for the project. Indeed, it might be one component of success. But, is it possible to get a lot of views without having a successful operation? Is the goal of the site to have lots of people see it, or are you hoping that visitors learn something? Do you need them to sign up for a service while they're visiting the site, or is the goal something else entirely? Page views is one possible metric, but it may not be the best proxy or indicator of success. A better metric may be sign-ups per unique page view. Or, a metric around the number of page views before a unique user enrolls may be better aligned with whether or not the site is "'working." It's worth brainstorming about different metrics – the first or easiest measure may not be the one that's best aligned with your project's goals.

you might think about your solution needing to create burn quickly (<N seconds), to fully burn target tissue (including full-thickness), or to burn the target tissue without damaging nearby structures.

- **Step 2 – Common Sense**
  Apply common sense to create a basic test description: If it's doing its thing, how would you know? If it's not doing its thing, how would you know? An aspirational data plot (as described in the toolkit called Prototyping: Question and Plan) can be helpful in defining the relevant metric and considering what engineering details are necessary for creating an appropriate test set-up. An example of a basic test description for the ablation project might be: Take a piece of meat, apply power, see if it burned where you wanted it to (and not where you didn't want it to).

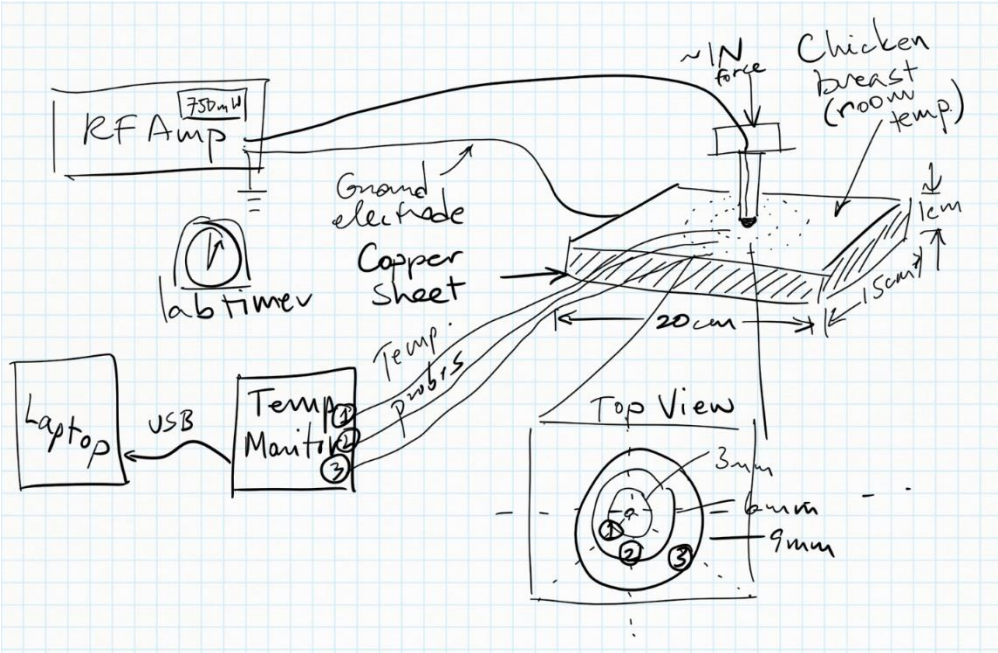
- **Step 3 – Engineering Know-How**
  Apply engineering know-how to envision the details of your test set-up and measurement plan, including units of measurement. Refer to your aspirational data plot (or create one if you haven't already – see Figure 2). Make a sketch of potential test set-ups or workflows that could yield the data in the plot (see Figure 3). An example of a refined test description for the ablation project could be: Use meat with X, Y, Z characteristics, apply power across a range of powers (10mW-1W) for a controlled amount of time (10-120 seconds), measure temperature (°C) in different places (map of locations) over time (150 seconds).

**Figure 2 – Aspirational Data Plots for RF Ablation Example**




Two examples of aspirational data plots that can help to motivate and clarify needed test set-up and prototcol details for different potential metrics. Left: Measuring temperature over time at different input powers to characterize the relationship between input power and temperature rise. Right: Comparing different temperature test locations/distances from the RF input electrode at a set heating duration.

**Figure 3 – Test Set-up Sketch for RF Abalation Example**




Rough sketch of test set-up for measuring temperature profiles surrounding an RF ablation electrode. The set-up depicts a slab of chicken breast meat ~1cm thick, with the RF ablation electrode pressed into the center surface and temperature probes placed on the surface at radial distances of 3mm, 6mm, and 9mm from the electrode. This sketch doesn't include all of the necessary details to run the experiment. However, it has the main elements and many of the key details, so it's capable of helping the team chose viable metrics. It also could help surface missing details that a team would need to create a working test setup.

- **Step 4 – Iterate with Creativity**
  Create and choose among options that have explicit linkage between the needs of the test set-up and the assessment(s). As with system design, iteration with creative thinking identifies ways to simplify or improve. For the ablation example, some potential choices might include:

  - Compare surface temperature at 3 minutes at the target spot and at 1mm radii (to 5mm) to 56 °C limit (from literature)
  - Calculate, over first 3 minutes, CEM43[1,2] (or another standard predictor of thermal damage, e.g., Arrhenius model) at multiple radii to define tissue "burn" region. Compare with tissue damage thresholds in the Yarmolenko reference.
  - Compare top and bottom tissue surface temperatures at 3 minutes at target spot to 56C threshold value
  - Are there easier ways to measure temperature over time (e.g., point measurements versus thermal imaging)? How can you relax the accuracy requirements (e.g., by testing across a wide range or running many trials)? How can you avoid having to set a quantitative benchmark by comparing with a competitive technology? Are there ways to get many pieces of data in one test run to avoid unnecessary duplication of testing?

Check out the video called Using Early Stage Testing to Inform Prototype Development. In this example, a team of students working on a novel tampon technology describes how their early focus on measuring a key metric – time to leakage – enabled them to establish proof-of-concept with various early prototypes and build a foundation for experiment-based development of their solution.

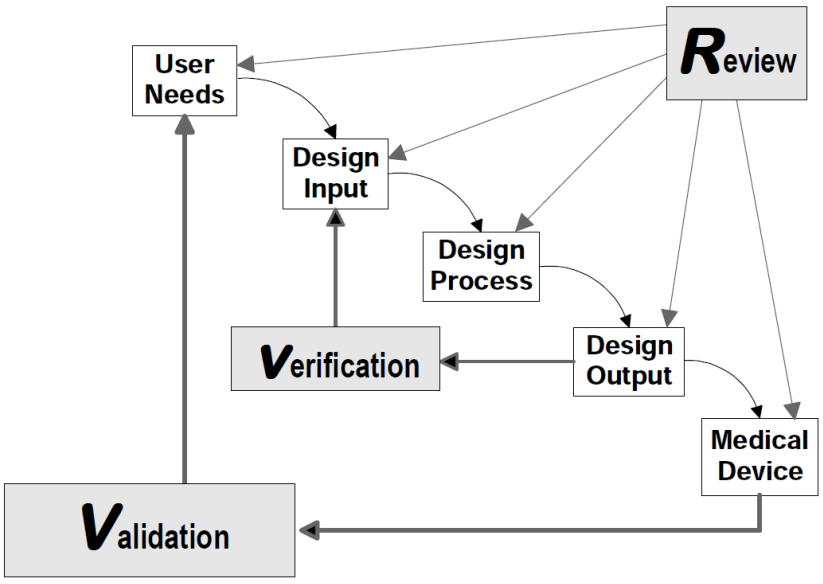
### Defined Classes of Metrics: Verification versus Validation

One important framework for testing and assessing using prototypes is known as verification and validation (often called 'V&V'). While similarly named, these two types of metrics are distinct in important ways. Verification metrics determine to what extent a design's functional sub-units are operating according to their specifications. Validation metrics determine to what extent a solution is (or is not) meeting user needs.

In health technology development, regulatory agencies (including the US FDA) require manufacturers to establish and maintain quality systems that cover the products they deliver. Verification and validation processes are an important part of the international standard for defining and maintaining these quality systems (ISO 13485:2016). Although early prototyping – especially in the context of a project-based class – doesn't necessarily have to be carried out within the rigorous structure of a quality system,[3] the same operating principles will yield sound experimental set-ups and reliable test methods to guide design decisions and future technical development.

The FDA guidance document[4] on design control for medical device manufacturers includes a diagram that helps clarify the relationship between user requirements (targets for validation testing) and design specifications (targets for verification testing). As shown in Figure 4, verification is part of the "inner loop" in the development process, confirming that the design output meets the goals of the design input. Validation, which directly considers user needs, is part of the "outer loop" of the development process.

**Figure 4 – The FDA's Waterfall Diagram for Product Development, Depicting the Roles of Verification and Validation**




Source: US Food and Drug Administration

Take a camera as a simple example. Verification metrics related to the design of the camera might include: 1) Does the shutter open and close at the right times and for the right durations?, 2) How spherical is the lens shape, as compared with appropriate specifications?, 3) What is the dynamic range of each of the pixels in the sensor array, and how do those compare with the specifications? The focus here (no pun intended) is on metrics that determine whether or not the camera is functioning as intended.

In contrast, validation metrics for a camera would center on whether it is meeting its user requirements, For instance: 1) What is the sharpness of the image of a known test object, under specified lighting conditions? How does this compare with user requirements for sharpness?, 2) How long does it take a user to properly focus the camera in a defined test environment that is indicative of indoor lighting and range conditions?, 3) Are user ratings of images from the camera sufficiently better than images from competitive or legacy cameras?

Additional examples of verification and validation activities from student projects can be found in Table 1.

**Table 1 – Examples of Verification and Validation Activities**

| Example Project | Verification Question | Verification Test/Metric | Validation Question | Validation Test/Metric |
|---|---|---|---|---|
| **bLOCKbox**<br><br>Counter-top opiate dispenser with physician-controlled intervals and integrated safe disposal capabilities to reduce unintentional opiate misuse and addiction. | Does the mixing plunger move fast enough to mix the slurry? | Measure the plunger rate between 10-20 plunges per minute when set to mix fluid from the application of water with DisposeRx until the completion of mixing (at 3 minutes). | Is the mixed product appropriately deactivated following mixing? | Compare the viscosities of the slurries, across the planned operating range of amounts of opioids, water, and DisposeRx, following automated processing and manual processing according to the DisposeRx instructions for use. Measure kinematic viscosity per ASTM D445 or ISO 3104. |
| **CathPath**<br><br>Device that leverages women's familiarity with tampon insertion and the relatively fixed orientation between the vagina and the | Will the catheter guide feature in the design(s) accommodate the full range of catheter sizes? | Compare the measured inner diameter of the 3D printed version of the catheter guide feature in each design with the maximum measured/ | How well can the new handle design work to improve cleanliness of self-catheterization? | Measure percentage of 1st-try catheter positioning success (in a simulated-use trial). Test across multiple handle designs, with and without blindfold, with comparison against both competitive commercial product (Astacath) and |

| | | | | |
|---|---|---|---|---|
| urethra to increase the ease and success rate of self-catheterization. | | known catheter outer diameter. | | standard of care (no tool). |
| **OpticLine**<br><br>Device that measures the optical density of white blood cells in peritoneal dialysis waste fluid using spectro-photometry to enable earlier detection and treatment of infection. | Does the photo sensor have appropriate dynamic range for different input fluids? | Measure the output voltage range of the photodiode sensor for the range of expected test fluids with different solutes. | Does the output of the system have a measurable and consistent response across the range of expected "preclinical" concentrations of white blood cells? | Measure the output signal, for N=3-5 samples with white blood cell concentrations from 0-5000 per mm^3 (in 500/mm^3 increments). Compare results with the same samples measured using lab-grade spectrophotometer in the wavelength range. |

To read more about these three projects, see Electronic Prototyping Example-Peritoneal Dialysis, Electromechanical Prototyping Example-Opioid Addiction, and Mechanical Prototyping Example-Neurogenic Bladder Dysfunction.

### Testing Step #2: Developing the Test Methods/Set-Up – How to Get the Data

Because of the central role of experimental investigation and evidence in the iterative prototyping process, it's important to be able to trust the data you generate. However, the various unknown aspects that are always involved in innovation projects make it challenging to find or develop well accepted and robust measurement methods. Even when you identify relevant standards or methods in the literature, these almost always have gaps or ways in which your experiments must deviate from the standard approach in order for the test to fit your specific context (e.g., maybe your test needs to be done at body temperature, or the fluid is non-Newtonian, etc.).

It's useful to treat test method development as its own iterative design process – establishing must-have and nice-to-have criteria for what you want to accomplish, brainstorming different options, prototyping promising options, and iterating based on how well they perform. As in all design processes, creativity plays an important role. However, a word of caution: Because it's so important for test results to be trusted, there's a premium on basing your tests on established methods. As a general rule, if you can use an existing test method for some or all of your measurement protocols, you should do that (rather than trying to invent a new method). Use your creativity to bridge the gap between the established test method or equipment and your application, as needed.

For example, a team that's trying to sensitively determine the mass of a component over time might find it tricky because of how the component is connected to other parts of their device. They could devise their own custom balance, that attempts to measure the part's mass using first principles and unique construction; or they could figure out how to design the part for easy removal and use a

calibrated lab scale. The latter may involve a bit of a redesign of the part and measurement protocol, yet it takes advantage of the significant time and energy the scale manufacturer put into creating a well-calibrated and reliable scale.

### Skepticism in Testing

There's an unattributed adage in science and engineering that captures a challenging duality that's relevant to health technology innovation: "Nobody believes a simulation, except for the person who did it. Everyone believes an experiment, except for the person who did it." This saying has two key messages related to testing.

The first key point is related to the fact that test conditions and/or test set-ups often are a type of simulation or model of the true context of the application. The person who builds that model typically knows why they believe the model can work for its intended purpose. People who are unfamiliar with the development of the model tend to be naturally skeptical about how well it represents the "real world." They understand that while building models is critically important in prototyping efforts, every model – whether mental, physical, computational, or otherwise – has limitations.

Conversely, the person who creates a test set-up and collects the measurements typically best recognizes the messiness inherent in experimental processes (e.g., uncontrolled variables that were not understood when the test was developed; intermittent measurement equipment errors; ways in which the test set-up was artificial, etc.). The tester has a first-hand sense of the limitations of the data. However, people who are unfamiliar with the details of the test are often naturally trusting of the data that has been produced, perhaps believing measurements to be inherently reliable since they happened in the "real world."

The result of this tension is that the experimentalist or innovator needs to be the chief skeptic about the validity and applicability of their own data so as not to fall prey to the biases of external parties or their own confirmation biases (desire for the technology to succeed). Your challenge is to bring a healthy skeptism to your work while maintaining a cautious optimism that iterative development will gradually lead to success.

### Test Development Practice: Recommendations and Examples

The best way to get going with test development is to **start simple**. Reliable measurements are always trickier and more involved to make than they initially seem. Starting with the simplest possible measurement, on a control unit with known/expected test outcome, is a good idea. It can identify early challenges with instrument calibration or readouts (e.g., is the scale zeroed properly? Is the display on the oscilloscope in the right range?), while helping you lock in a data collection process and build momentum for making more challenging measurements.

As one example, the OpticLine team referenced in Table 1 initiated testing by measuring different fluids using a benchtop spectrophotometer in an attempt to replicate the literature values for known fluids. This established a baseline for the range of values they might see with the prototype of their device, while also enabling them to consider how many replicates they would need, how to plot the data, etc. Read more of this example in Electronic Prototyping Example-Peritoneal Dialysis.

Another team, working on a digital solution for managing chronic lower back pain, got started by asking people how often they would respond to surveys about their pain levels sent via text message. With this approach, they gained practical lessons about the challenges of gathering high-quality data via survey. They also got an idea of how often they could ping test-users with their next-step prototypes. Check out the full story in Digital Prototyping Example-Lower Back Pain.

An additional recommendation is to **test independently – with controls – before testing experimental variables in combination**. Whenever practical, make measurement across the relevant range of a single test variable, and then across the relevant range of a different variable, before attempting to test across both variables.

Be ready to make trade-offs. When it comes to designing and building test set-ups, innovators face multiple **test apparatus trade-offs**. There's a tension between the following properties:

- Measurement fidelity (accuracy/precision/repeatability)
- Flexibility/adjustability/ease of use
- Cost/time

While the measurement fidelity is most important – your apparatus must be able to make the measurements that are central to the test – teams often under-appreciate the value of test set-up adjustability, preferring to spend time on prototype design/construction. While making a test set-up too flexible can lead to challenges with operability or repeatability, early prototyping experiments frequently require on-the-fly adjustments due to real-time learning about the system. Time and cost are important considerations, too, that may be dictated to some extent by the resources available to you and the pace of your course/development.

**Scoping a test** appropriately also requires some forethought. The following questions are useful to consider while developing a test and preparing your test protocol.

- How many samples or test objects will we need?
- Will the test be destructive of either the test samples or the test equipment (intentionally or perhaps unintentionally)?
- How many controls should we have?
- How much "touch time" (amount of active person-time, in real time) will each replicate/repetition require, including set-up and fixturing?
- How much calendar time will the full experiment take (including practical limitations on facility hours, length of time for biological samples to grow, etc.)?

It's often helpful to employ multiple **independent measurement techniques**. While most measurements involve a specific sensor for a given quantity of interest, think about whether you can come up with more than one way to measure the same quantity, especially if the primary instrument is being used for the first time or may be unreliable. For example, if you're using an infrared camera as the primary measure of temperature, it may make sense to place a thermocouple at one or more spots to achieve higher resolution (in time or space) or to confirm calibration at a location of interest.

Finally, it's important to **simply start**. Given the many potential variables and conditions to consider, teams sometimes feel overwhelmed and try to control every detail before getting started (acknowledging that the previous recommendations may contribute to such a feeling). While being prepared is important, details that are difficult to foresee often don't emerge until you begin making measurements. Make and use a checklist to cover your bases, and then get going, while being ready to learn along the way!

Refer back to the digital, electrical, electro-mechanical, and mechanical prototyping examples for more tips and pointers. Additionally, watch the video called Advice for Early-Stage Prototyping and Testing for guidance from the team working on the novel tampon technology.

### Testing Step #3: Actually Getting the Data

Once you've decided on your test metrics and developed your test set-up, it's time to begin gathering data. This is an exciting opportunity for learning – yet the value of the data that drives this learning will depend on the integrity of your experiments. As usual, the time and energy you put into building and executing your plan is directly linked to the quality of the data you gather, so double-check to make sure you've taken care of all of the following to ensure the best possible results:

- **Personal protective equipment** (PPE). Safety first. Seriously.
- **Written test protocol**, with all steps and details of the controllable test conditions determined/recorded, so that it's possible for someone else to fully recreate the experiment in the future. Use version control for each revision to the protocol by assigning each one a unique identifier. Be sure your protocol includes PPE guidance, as well as material reuse/disposal instructions. These protect your teammates and community (bonus: they also help lab managers approve your protocols).
- **Data entry sheet**, with demarcated columns/spaces for all of the test conditions and blank spaces to record conditions and results that can only be recorded at run-time, as well as links to the associated data files.
- **Data storage media and/or locations** pre-allocated.
- **Samples/prototypes/materials** that will undergo testing, with unambiguous identification of each unit. These should include "positive control" unit(s), "negative control" unit(s), and calibration standard(s).
- **Other supporting materials** that are not necessarily part of the prototype or test equipment but will nevertheless be needed (e.g., tools for fixturing, drop cloths to protect surfaces, extension cord/power strip, rags/paper towels for clean-up, extra fixturing hardware, duct tape, masking tape, laptop charger, tripod for camera, etc.).
- **Measurement tools** with appropriate ranges for recording test data
- **Training** on equipment (for all teammates). Not everyone needs to be an expert user, or be able to use every function of a piece of complex equipment, but the team needs to know enough to be able to make some 'on the fly' adjustments.
- **Sensors**, calibrated for your test(s), with the range(s) of interest. Note: control measurements should confirm this.
- **Assigned roles for team members before, during, and after the experiment,** including: sample preparation, data collection, experiment logging, photographer, measurement equipment setup, safety officer, and 'gopher' (the person who will run to 'go for' anything the team forgets/needs during the experiment).

Because every project, and each experiment, has its own unique aspects, this set of considerations may need modifications for your situation. With that in mind, teams find it useful to make and use a Testing Checklist when they're ready to initiate data collection on a project.

#### Documentation of Testing = Critical for Assessment

When gathering data, it goes without saying that a primary goal is to capture the measurement data itself. That said, novice experimenters sometimes overlook the importance of capturing and documenting the experimental conditions that give the results meaning, often thinking that team members will remember the specifics without writing them down. A useful adage in health technology innovation is, "If you didn't document it, it didn't happen." Although it may seem like overkill, the goal should be to have enough documentation that any competent engineer could fully replicate your experiment and results (and documentation) without having been a part of the

original testing. Particularly with early prototyping, when part of the experimental goal is to discover potential issues with the testing set-up and the team will want to get input from mentors about how to interpret their data after the experiment, robust documentation of the measurements and conditions is key.

### Testing Tips and Tricks

The following guidelines include some 'lessons learned the hard way' and some other ideas for how to help improve the likelihood that your testing will yield useful results.

- **Rules of thumb for estimating experiment time/budgets**
  Teams typically spend less than 10% of the end-to-end active time to complete an experiment actually making measurements. You should spend the vast majority of your time planning, setting up, analyzing results, and defining next steps. So, if you think an experiment will take two person-hours to make all of the measurements, you should budget for at least 20 person-hours of work.

  Breaking this down a bit: Making one hour of measurements takes ~four hours in the lab, especially if you're executing a new protocol. Pre-game/set-up typically takes two-times the measurement time. It also always takes longer than expected to clean or disassemble the set-up, back up the data, complete the experiment logs, and safely store everything so that it can be used for the next experiment.

- **Make time for "dry runs"**
  Dry runs are an invaluable and efficient way to avoid trouble during your actual testing. The day before running the experiment, consider doing the following: Get an editable copy of the test protocol, a pen, and a big box. Go through the protocol step-by-step, while literally putting everything that you need to run the experiment successfully into the box.

  Once you put everything in the box, virtually run through the test protocol and think about the first thing you'll do when you open the box, then the second, and the third… Additionally, be sure you're clear on what you're going to measure/plot to demonstrate that the experiment is working (or not working). Use the pen to make notes on the protocol, and repeat the process until you get all the way through the protocol without making any additional notes.

- **Trouble with precision?**
  Make lots of measurements and evaluate averages (making sure to include error bars on the plots). Consider if the issue is random or systematic (e.g., changes over time, or perhaps with another variable), since systematic errors will not 'average out.'

- **Trouble with accuracy/calibration?**
  Make relative measurements against a trusted standard. Uncalibrated measurements, with arbitrary units, can still demonstrate quantitative relationships/curves that confirm theoretical understanding (e.g., linear versus non-linear) or identify saturation ranges. And relative measurements can demonstrate quantitative differences without necessarily requiring calibration or absolute accuracy.

- **A vs. B testing**

Measure performance for a set of prototypes under equivalent conditions to directly compare them on a relevant metric. This can be an especially useful technique if there are questions about the accuracy or calibration of a measurement tool and if there is an existing competitive technology for comparison.

- **Plan for failure and plan for success**
  Things rarely proceed exactly as planned, so experienced innovators plan for things to go wrong and have back-up materials/plans for every experiment. Establish a "minimum necessary data" goal, so that the team knows if/when you have "enough" of the experiment completed to draw any conclusions. Additionally, keep in mind that experiments create opportunities for learning that can be significant and unanticipated. Establish "stretch goals" for experimental data that you might be able to gather if/when things run smoothly or the opportunity presents itself.

### Completing Data Collection and Moving on to Assessment:

Teams sometimes start analyzing data in the midst of the data gathering process, and this can be necessary for early range-finding experiments. However, best practice is to complete data collection first to avoid bias during the assessment process and to perform the assessment with the full set of data. Once you have robust and well documented experimental data (and test conditions) in hand, you're ready to start analyzing and interpreting what happened, and it is time to start your Assessment in earnest.

## Assessing and Iterating

When it comes to the six-step prototyping process, some innovators consider the assessment step as part of testing. They are clearly interrelated, but it's useful to consider them distinct when planning a project, as the inputs and outputs are quite different. There also are disciplinary/technology-specific best practices and common methods for assessment that you should consider as you approach experimental assessment (e.g., electronics versus software versus tissue culture processes typically use different assessment methods).

Fundamentally, assessment is about addressing three high-level questions using data from the testing phase:

1. Did the test work?
2. What does the test data mean for the project?
3. What to do next?

Read on for guidance on how to actively address these three questions in your assessment work, and how this often leads to iteration on your project.

### 1. Did the Test Work?

The first step in figuring out what the results mean for a project is determining which interpretation(s) of "working" is/are appropriate for the data at hand. So, what does it mean to "work"?

Whether or not something worked is fundamental to the assessment step of the prototyping process, yet it's challenging to answer this question directly because it's ambiguous. There are at least three different layers to this question, and each layer needs to be considered thoroughly. Some variations of the "did it work?" question may seem obvious or based on common sense, but each is worth addressing explicitly.

**1st interpretation**: Did the *test method* function sufficiently well? That is, to what extent was the combination of test equipment, test conditions, and test protocol able to yield interpretable data? The way to answer this variation of the question is by analyzing the results of the positive (and negative) experimental controls – those tests with known/reliable results – and determining if your data were close enough to the expected results to be trustworthy. This is the first question to address in any assessment because, if the test method was not generating useful data, then the team needs to fix the test method before proceeding and should avoid trying to use the data (except to figure out how to improve upon the test method).

Remember: it's common for protocols or tests to fail against basic controls during a first run. And the likelihood of protocol failure, and hence the need for careful controls, increases with the complexity/number of steps in the protocol.

**2nd interpretation**: Did the prototype or solution being tested perform according to its design? That is, did the prototype behave as a prototype of the design, or did it fall apart/fail/do something else entirely? If the prototype didn't function according to its design during the test, then the results will likely be irrelevant (or worse, misleading). The most straightforward way to have confidence that your prototypes are performing according to their design is through repeated testing of several replicated prototypes of the design (note: replication is a statistical term). While it can be costly, in terms of time and other resources to create multiple units and perform multiple tests on each, it's often more costly in the long run to draw inappropriate conclusions from a single prototype that may or may not have representative behavior. Some technologies are more likely to require true replicates (e.g., biological samples), while replicates may not be as important in other situations (e.g., software, electronics), depending on whether or not manufacturing or other variables could conceivably impact performance.

**3rd interpretation**: How (well) did the prototype perform in the attempted test? That is, does the data indicate that the design accomplished the task at the heart of the test? How well did it do the thing(s) it was designed to do? Once you have confidence in the test and in the prototype/system being tested, then it's possible to perform the assessments that have broader implications for the overall project (e.g., proof-of-concept for a mechanism, characterization of performance across a range of conditions, etc.).

### The "Walking Wounded" Prototype

It's not unusual for early prototypes to misbehave in ways that are unrelated to the purpose of the ongoing test (yet could impact results). Parts break. Seals leak. Programs crash. Samples get contaminated. Sometimes these types of errors are difficult to detect, as the prototype seems to be working. Simple inspection/ monitoring of your prototypes before, during, and after testing can often help during early prototyping. In more advanced testing, and with more complex systems, teams sometimes develop standard or automated test protocols to run before and after the main test to ensure that the prototypes/systems were not broken or "walking wounded" (mostly working, but on the verge of failure) during the experiment. For example, with a neurostimulator that has internal circuits that can break, it's common to establish a set of nodes (or test points) and a procedure for efficiently measuring the voltages and comparing them with their known target range before and after a test to give you confidence that nothing funny was going on with the device that would impair your interpretation of the results.

## 2. What Does the Test Data Mean for the Project?

Beyond determining whether a test worked, you should consider different categories of outcomes when thinking about how to proceed. Each category has different implications for the project that depend on the specific test scenario and results.

- **Category 1: Great experimental outcomes**
  The test and prototype both functioned, and *we have credible evidence that* …
  - …the concept is infeasible under the testing conditions
  - …the concept is potentially feasible under the testing conditions
  - …specific ranges of conditions are feasible for the design to work
  - …either supports or disproves the specific hypothesis under test
  - …etc.

- **Category 2: Pretty good experimental outcomes**
  Either the test or the prototype failed to function, and *we learned that*…
  - …the problem was with the measurement setup
  - …the test setup was too variable because we did not control X
  - …the test setup worked until Y level, and things failed after Z level
  - …the prototype needs a stronger _____
  - …the next prototypes should include a better way to _____
  - …etc.

- **Category 3: Not good outcomes**
  Either the test or the prototype failed to function, and *we did not learn much* more than that because…
  - …we didn't do a great job capturing the experimental conditions or the data
  - …the measurement tool seemed out of calibration
  - …we weren't sure what "working" would mean
  - …we weren't sure which range of inputs to use
  - …we couldn't tell whether the problem was with the test or the prototype because we didn't have good controls
  - …etc.

- **Category 4: Worst case scenario outcomes**
  Either the test, the prototype, or the design failed to function, yet *we incorrectly think that things worked* because…
  - …we didn't have controls in place and the experimental conditions created a trend that made it seem like everything was working
  - …we didn't realize that there were confounding factors inherent in the design or test method
  - …we misunderstood the way the test would relate to the project requirements
  - …we squinted hard enough at the data until it seemed like it was working
  - …etc.

Note that novice teams often misunderstand their goal during assessment to be "how can we interpret the data in such a way that that we can say the test worked?" However, the most

important part of assessment is actually figuring out what *didn't* work, so that you can understand why the test failed and how to fix it. Curiosity and honesty are crucial to successful assessment, as well as to driving project progress.

When Results are Difficult to Interpret (and How to Avoid This)

Importantly, the results of system performance tests often can be difficult to interpret. This is expected – the whole reason to do the experiment in the first place is because the team is unsure about what will happen. It understandably can feel discouraging to spend a whole day (or week) on an experiment and then not be sure what the results might mean. However, it also can indicate that the team is about to learn something fundamentally important about the design concept. One way to increase the likelihood of interpretability in early prototyping is to perform experiments across a range of conditions, with different combinations of input variables, and a goal of creating a **characterization curve** (as opposed to a binary "does it work or not" result).

Although not all prototyping efforts require or fit with a characterization curve approach, it's useful to have a working hypothesis of what the data should look like if the design functions according to the present theory of operation (refer back to your aspirational data plots). This way, the data either can support or recommend adjustments to the team's understanding of the design. For example, when creating a stable environment for a testing rig at human body temperature, a team may want to achieve ~37°C +/- 1°C during operation. If it seems like it is working, the team might gain further confidence by hypothesizing that the temperature would tend to be a little higher during the day and lower at night due to the external conditions, and this might be seen in the data. Alternately, the team could hypothesize that the electric current required to hold the temperature steady would be higher at night or when the room is cooler, and a plot of the required operating current and the external temperature could reveal if they correlate as expected. If the correlation is different, closer analysis might yield the confounding factor that solves a problem.

Check out the video called The Novonate Project: Using a Test Rig to Evaluate and Compare Prototypes for an example of how the Novonate team developed a series of testing rigs for their neonatal catheter securement device. Using calibrated measurement tools, straightforward data analysis techniques, and some creativity, they were able to develop tests to reliably and quantitatively distinguish between their candidate prototypes and the standard of care.

Thinking about Data

Data analysis is an important and nuanced effort. It can be incredibly complex and challenging – but it doesn't have to be. Here's some guidance to help you successfully analyze data in early-stage prototyping.

- **What data is, and what it's not**

  Data comes from discrete measurements (numerical results, with units of measure) of what happened under specific conditions. The data must be analyzed in that context, and it's only as good as the underlying metrics and the measurements actually made, including knowledge of the relevant conditions.

  Interpolation – making data-informed guesses about what might happen *between two different measurement conditions* – is often reasonable and acceptable. This is sometimes seen when teams plot data from experiments with linear or smooth curves connecting their discrete measurement points. However, a best practice for reporting data (especially early prototyping data) is to only plot the data points themselves, perhaps with error bars that reflect either known or expected variability.

Extrapolation – making data-informed guesses about what might happen *beyond the range of the measurement conditions* – is generally not advisable or acceptable. A potential exception in early prototyping is if a team doesn't know the appropriate testing range ahead of time so they perform experiments with conditions that are either entirely above or below the needed range for their prototype. In such circumstances, they might extrapolate from the data to choose their next testing range. However, the best practice for avoiding the need to do this is to start with a "range finding" experiment, in which you make a set of measurements across a wide range of input conditions to determine on which range of inputs to focus your more detailed experiments.

- **What is and is not (useful) data**

  Having measurement numbers is a necessary, but not sufficient, condition for having useful data. Measurement equipment will (almost) always produce a numerical output, yet those numbers are only meaningful if they represent test performance.

  Data is replicable – you and others would get the same (or nearly the same) result if you replicated the test. If you're unable to repeat an experimental result, it's hard to base future project decisions on that result. Sometimes, "the stars align" and something works once, but not upon repetition or replication. That measurement may not be useful until you have specific reasons to believe that the repetition requires different conditions and a plan for achieving those conditions.

  Data contains uncertainty – there's always some amount of error or variability in the test method. If the data seems too clean to be true, then it probably is. If you need the data to be error-free in order to demonstrate or assess prototype performance, then you probably need a different experimental setup/system.

  Simpler data analyses are almost always better. The more complex the analysis is, the more likely it is to lead you astray. There are valid techniques for capturing a useful signal from what seems like terribly noisy data. However, most of these methods require a clear understanding of the underlying physical mechanism(s) to work (fitting to a square-law relationship, spectral analysis, exploiting sparsity, etc.). And almost all can suffer from overfitting of one sort of another.

  For example, teams sometimes try to use higher-order polynomial fitting when creating calibration curves. This is because higher order fits tend to have lower R-squared values that make it seem like the gathered data fits well to an underlying curve. In many cases, though, the physical phenomenon is a linear or square-law relationship and the data is simply noisy because of inconsistencies in the test set-up, challenges with the sensors, etc.

Plotting and Interpreting Data – An Example

The plots shown below include examples of characterization data from a team project that used off-the-shelf sensors to measure pressure in a wearable application. The team wanted to make sure that they could make reliable measurements with the sensors before integrating them into the wearable. The first data plot (see Figure 5), which includes averages across different sensors and different trials across a range of pressures, demonstrates several best practices that enable assessment, along with at least one opportunity for improvement.

Best practices:

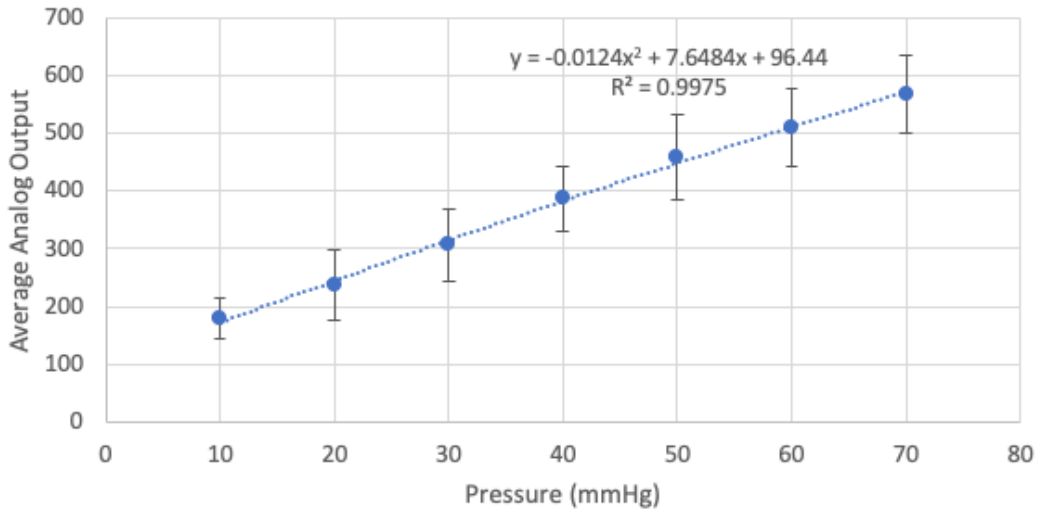
- Includes specific data points, not only the fit line

- Includes error bars that reflect distributions of underlying data
- Includes clear axis labels, including units of measure (mmHg) where appropriate
- Includes equation for fit curve, along with $R^2$ value

Opportunity for potential improvement:

- The underlying relationship seems linear, not square-law – ought to include the linear fit as a potentially simpler relationship

**Figure 5 – Average Analog Output Across Sensors and Trials at a Given Pressure**




Example plot, from a student team project, of pressure sensor system output data (arbitrary units) vs. input pressure (mmHg) for a set of four sensors across a range of input pressures. The plot demonstrates reasonably linear sensor performance, along with relatively high variability across measurements at nominally equivalent input pressures.

Continuing the example, there are a few reasonable assessments from the first plot:

- Pressure sensors seem to be functioning properly
- Sensors have a relatively linear response to pressure across the range of inputs
- While averages across sensors are well-behaved, the variability at each point seems a bit high (e.g., at 50mmHg, outputs range from ~380-550, which is +/- 15-20%)

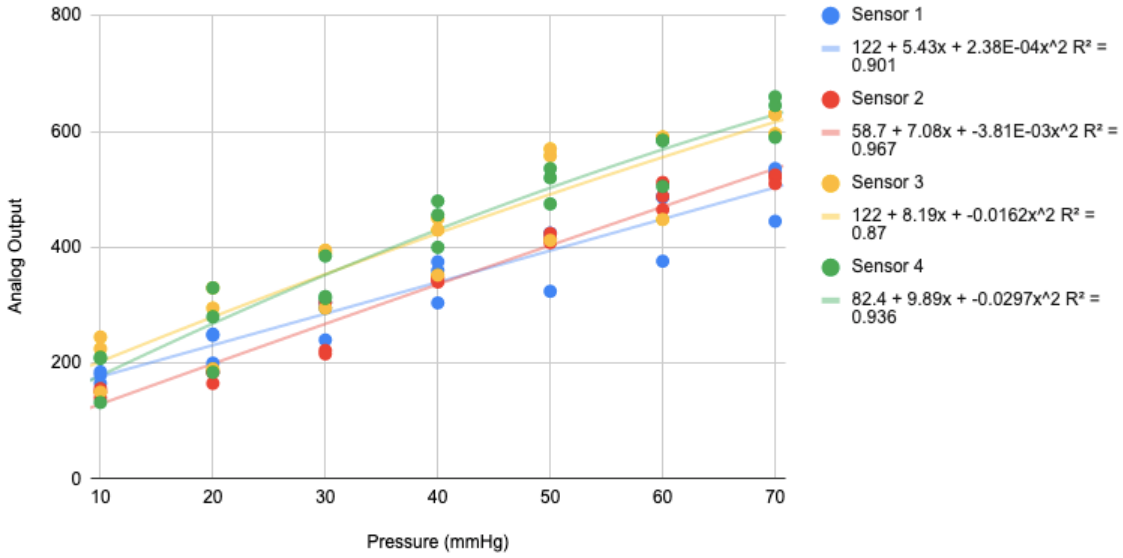
Based on the first two points, the team could feel confident proceeding with sensor integration. The third assessment point suggests a few possibilities for future investigation:

- Individual sensors may behave differently and require different calibrations
- The pressure sensors may be sensitive to positioning or other environmental characteristics
- The inherent precision of the sensors may be ~15-20%

As it turned out, a sensor-by-sensor analysis (see Figure 6) demonstrated that different sensors had different intercepts in their lines of fit, so a unique (linear) calibration curve would be necessary for each sensor, as opposed to the team's hope that a single curve could work for all sensors. One issue was that the individual sensors still seemed to have significant variability across measurements at nominally equivalent pressures, so a further analysis would be needed to see if

individual fits for each sensor could get the variability low enough to meet the team's measurement needs.

**Figure 6 – Flexiforce Analog Output at Various Pressures**




Measurements of sensor output [arbitrary units] vs. pressure [mmHg] for four different sensors, along with calculated fitting curves. Because the sensors have different intercepts on the fits, individual calibration may be needed for these sensors to work for this application, since the team's need criteria required pressure control accuracy in the ~10% range.

## 3. What To Do Next?

The decision about what to do next on a project is often a question about priorities and resources, which may require a judgement call under conditions of uncertainty. Such choices are challenging as teams face many different risks that they could be addressing, especially during the early stage of a project.

At a high level, there are two options for the next step:

1. Document the assessment and iterate to address the original project question/risk
2. Document the assessment and move on to a new question/risk

In both cases, documentation of the assessment is critical, as the project value that the team is building depends entirely on your ability to explain your decisions and the insights you've gathered (ideally backed up by experimental data). Time spent at this juncture creating a summary plot, with a few explanatory sentences and/or bullet points with links to the experiments, is always worthwhile. If your project is part of a class, this work will be part of your final presentation or report. If you're seeking further support for the project, via investment or a grant, the documentation will be part of your pitch deck or funding application (see the toolkits called Advancing Your Project Beyond Class and Presenting Your Project).

At the end of each round of experiments/prototyping, and especially before jumping into a new iteration, you should ask whether there is enough evidence (for now) on the specific target question to shift focus to a new risk. Once you've been able to analyze the test data, and you're confident

about what happened during testing (or you at least know what you're not confident about), you're ready to consider the questions in Table 2 to help you determine how to proceed.

**Table 2 – Questions to Help You Determine What's Next**

| | |
|---|---|
| What specific question(s) led the team to do this experiment? | • Are there answers to those guiding questions? How preliminary or convincing is the data? Make a plot for each question to help you clarify this.<br>• If there's not data relevant to the team's guiding questions, why not? Did new issues came to light, or did the team get distracted? An iteration may be in order to get back to the central issue. |
| What new question(s) do we have? | • Do you have better (more refined) versions of the original question(s)?<br>• Are there some practical questions that emerged regarding the test methods or setup?<br>• Do you have entirely new questions about the design/concept that will guide the next steps in a different direction? |
| What does it mean to be 'done' (for now) with this prototyping/testing effort? | • Did you establish a clear goal in advance of this round of prototyping? If so, can you tell if you either met or failed to meet it?<br>• Can you quantitatively connect the existing experiments and data to the need criteria? A plot with a target value (using experimental data, or even aspirational data) can help with this. |

Typically, there are two main pathways that teams take at this stage:

- **Option 1: Another Iteration on the Same Question(s)**
  If the choice is to continue to focus/iterate on the same question(s) or risk, your immediate task is to identify which things need to change in the next iteration, and which things need to stay the same.

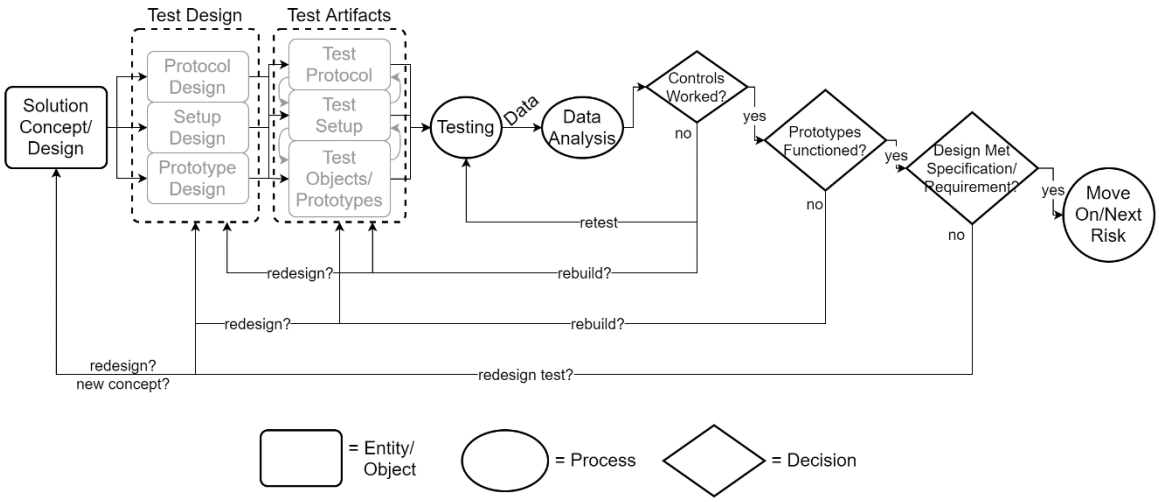
  In many cases, determining which level of "working" the experiment achieved (i.e., which of the three previously mentioned interpretations apply to the data) directly indicates which type of iteration or next step is necessary. For example, maybe you need to work on the test setup/method, on the test objects/prototypes, or on the concept/design. See Figure 6 for a flow chart to clarify this approach.

  Note: Most early experiments run into minor practical issues, and most iterations involve repetition of prior work with a few critical updates. Accordingly, you'll rely heavily on your documentation from the first iteration as you proceed with the planning and execution of your next iteration. Your level of efficiency and success will depend upon the quality of your documentation, as well as how much you learned and your ability to integrate it into the updated test methods/setup/objects.

- **Option 2: Moving on to a New Question**
  If the choice is to move on to a new question, the scope of work will depend entirely upon how different the original question is from the first. However, the overall process is the

same. Get started by going back to the toolkit called Prototyping: Question and Plan and repeating those steps with a new risk in mind.

**Figure 7 – Decision Diagram for What to Do Next**




Process flow diagram for the test/assess/iterate portion of the prototyping process, with decision points and feedback based upon experimental outcomes. If the experimental controls fail to work, sometimes simply repeating the test is the right course of action because the error was in the testing itself. However, it might be that the failed controls indicate a problem. Also, multiple issues can arise from a single experiment.

## Pitfalls when Assessing and Iterating

As with every step in the six-step prototyping approach, assessing and iterating have their own potential pitfalls. Watch out for the following issues, which are common among project teams.

- **Analysis Paralysis**
  Getting stuck in a constant state of analysis (rather than making a decision and proceeding) sometimes happens to teams when there's significant uncertainty about how to analyze or interpret data. In these cases, some members of the team may feel more comfortable analyzing data than making potentially risky decisions, especially when the next experiment may be costly in terms of time or other resources.

  One way to avoid analysis paralysis is to have clear analysis methods and goals defined ahead of the experiment, so that the data and results can make the decision for you. For example, a team might state/agree "if the results are above X level of performance, the team will declare victory and move on. But if the results are below Y level of performance, the team will iterate on the design."

- **Failure of Imagination**
  If things are not working and the team has no good ideas about what the issues might be, one way to get "unstuck" is to get help from an objective observer and/or an experienced mentor. A fresh perspective can stimulate creativity in the team and open up new ideas regarding the best way to proceed. Summarizing your results and conclusions into a few plots and bullet points – being sure to include the driving question for the overall prototyping effort – is a good way to help someone help you.

- **Attempting to Use Irrelevant or Inapplicable Test Results**
  After running a series of tests, especially ones that require a lot of time or effort, there's a lot of motivation to use and interpret the data, even if the test results are highly suspect (e.g., the control measurements failed, the prototypes broke in the middle, there were missing data points, etc.). Having different team members lead different aspects of the experiment (data collection versus data analysis) can help avoid this issue. Another good idea is to appoint one member of the team (perhaps on a rotational basis) to play the role of the "team skeptic," creatively and constructively criticizing the data/analysis.

- **Different Prototypes Need Different Tests**
  When the purpose of a test is to help the team decide between different design ideas, but the design ideas are sufficiently different, the test results sometimes only apply to one or two of the designs. If this is the case, it can feel like trying to make an "apples to oranges" comparison. Recognizing this is the first step to finding a solution. Typically, this involves developing a set of test methods that are each applicable to a different design/prototype.

## Practice Makes Perfect

In the same way that a solution concept evolves and improves with iterative prototyping, your testing and assessment skills will develop with practice. Whether you're working in a team or on your own, be mindful of your approach and actively apply your learnings to each successive round of testing, assessing, and iterating to increase the pace and efficiency of your progress on your project.

## Credits

Ross Venook prepared this brief with assistance from Lyn Denend. We'd like to thank Amanda Calabrese, Eric Chehab, Greta Meyer, and Bryce Yao for their assistance with the videos, as well as Alisha Birk, Mark Buckup, Matthew Carter, Maurice Chiang, Karen Dai, Gabe Ho, Maria Iglesia, Isaac Justice, Janelle Kaneda, Sai Maddineni, Yash Pershad, Adeline Petersen, Hannah Schofield, Tara Shannon, Daniel Tang, and Amanda Urke for sharing the example projects linked within the brief.

## Notes

[1] SA Sapareto SA, WC Dewey, "Thermal Dose Determination in Cancer Therapy," International Journal of Radiation Oncology, Biology, Physics, April 1984, https://www.sciencedirect.com/science/article/pii/0360301684903791 (August 3, 2022).

[2] PS Yarmolenko et. al., "Thresholds for Thermal Damage to Normal Tissues: An Update." International Journal of Hyperthermia, 2011, https://pubmed.ncbi.nlm.nih.gov/21591897/ (August 3, 2022).

[3] "Quality Management Systems," Wikipedia.org, https://en.wikipedia.org/wiki/Quality_management_system (August 3, 2022).

[4] "Design Control Guidance for Medical Device Manufacturers," US Food and Drug Administration, March 1997, https://www.fda.gov/regulatory-information/search-fda-guidance-documents/design-control-guidance-medical-device-manufacturers (July 22, 2022).