

# Chapter 3

## Developing Local Innovation Capacity to Drive Global Health Improvements



Christopher Moses

**Abstract** In global health, innovation often comes from “outside-in”: industrialized countries develop new drugs, devices, or services, and export them to low- and middle-income countries (LMICs) (Syed et al. in *Global Health* 9:36, 2013). Yet there is a growing recognition that there is real potential for “bi-directional flow of knowledge, ideas, skills and innovation” (Syed et al. 2013). To generate sustainable impact at scale, high-income countries should further encourage this local innovation capacity. One way to do so is to export more than just finished products to LMICs, but also the knowledge, processes, and cultural mindset that support repeated success in new product and service development.

**Keywords** Design thinking · Empathy · Outcome driven innovation · Agile Software Development · Product lifecycle

### Learning Objectives

This chapter begins with an overview of core concepts in product innovation, then presents an innovation workshop. The goal of this chapter and its accompanying workshop is to provide teams with a set of fundamental concepts for thinking about new product development, as well as a detailed set of exercises whose output offers concrete steps for either increasing the chances of success for an existing product or service, or taking a brand new idea into the prototype phase. This workshop has been taught in multiple settings (corporate, academic, and startup) in both the United States and Thailand. By training teams in this way, we might best support nascent innovation capacity and help teams in LMICs bring more desirable, feasible, and viable innovations into their local communities, and potentially to the world at large.

---

C. Moses (✉)

Critical Data, Massachusetts Institute of Technology, 77 Massachusetts Avenue E25-505,  
Cambridge, MA 01239, USA

e-mail: [cmoses@alum.mit.edu](mailto:cmoses@alum.mit.edu)

© The Author(s) 2020

L. A. Celi et al. (eds.), *Leveraging Data Science for Global Health*,  
[https://doi.org/10.1007/978-3-030-47994-7\\_3](https://doi.org/10.1007/978-3-030-47994-7_3)

### 3.1 Core Concepts in Product Innovation

There are many strategies high-income countries can leverage to best catalyze global health innovation in LMICs (National Academies of Sciences, Engineering, and Medicine 2017), but many of these approaches focus on technology investments. However, not all great innovations are technological in nature, and innovations in processes and implementation methods can have significant impact (Donaldson 2014). Interestingly, while the United States and other high-income countries are recognized for taking the lead in defining, refining, and systematizing innovation as a practice, their chief exports are often the products themselves rather than the less tangible, but critical knowledge and processes that enable repeated success in innovation. To build a common foundation for readers, we first review five core concepts in product innovation, and highlight frameworks that implement them. These frameworks are not mutually exclusive, and are in fact closely related in their approach to product innovation. Together, they can offer valuable points of view. These core concepts are (Table 3.1).

**Table 3.1** The five core concepts of product innovation and their corresponding frameworks reviewed in this chapter

Core Concept	Framework
1. Conduct qualitative interviews to build empathy with users to better understand their unmet needs	Design Thinking (Brown 2009; Dam and Siang 2019)
2. Translate users' needs from high-level qualitative descriptions into standardized, testable statements to derive quantitative results that guide product development	Outcome Driven Innovation (Ulwick 2016)
3. Identify underlying assumptions about the user, their needs, or how the product or service will be implemented, then define a test plan to reduce risk and increase chances of success	Sense and Respond (Gothelf and Seiden 2017)
4. Test a new product or service with users as it's being built to better respond to changes in the user, their needs, and their environment	Agile Software Development (Beck et al. 2001; Koch 2011; Lucidchart Content Team 2017a)
5. Products have a lifecycle—a beginning, middle, and end—and there are important trade-offs to consider between product risk and operating cost over the life of the product	Think It, Build It, Ship It, Tweak It (Kniberg 2013)

## 3.2 Core Concept 1: Build Empathy with Users

In global health, those developing new products and services are often not the end beneficiaries of the innovation. As such, it is critical to deeply understand the target population as individuals to enable us to design more relevant solutions that better fit into their lives. Without building this empathy, we're more likely to miss important factors and apply misconceptions that could result in less useful products and failed implementations. As noted by Juma and Yee-Cheong (2005), "Creating appropriate products for low-resource settings requires not only a rethinking of what is considered a health technology, but also cross-disciplinary innovation and in-depth understanding of the particular needs of each country." Design Thinking is one popular approach to innovation with a foundation in building the empathy necessary to understand our target audience:

What's special about Design Thinking is that designers' work processes can help us systematically extract, teach, learn and apply these human-centered techniques to solve problems in a creative and innovative way – in our designs, in our businesses, in our countries, in our lives. (Dam and Siang 2019)

For more detail on how to build empathy with users through qualitative interviews, readers are referred to the chapter by Tony Gallanis in this book and to Tim Brown's capstone book (Brown 2009) on Design Thinking.

## 3.3 Core Concept 2: Define Standardized User Need Statements

Spending time empathizing with users enables the designer to formulate the user's problems and goals in a user-centric way. These user need statements, also called problem statements or point-of-view statements, capture what a user is trying to get done and why. They can be defined in various formats (Gibbons 2019; Dam and Siang 2019; Barry 2010), and can help direct the ideation process when brainstorming solutions. For example:

Carolyn, a frail retiree recovering from knee surgery, needs to learn and practice mobility exercises so that she can quickly return to her normal daily activities that really matter to her.

In his book, *Jobs to be Done* (Ulwick 2016), business consultant Anthony Ulwick proposed an alternative format for need statements as well as a process to enable teams to more exhaustively define, prioritize, and test which user needs are most significant and ripe with opportunity for innovation. His process, called Outcome Driven Innovation, focuses on better understanding what "job" a user is trying to get done, and how users measure the value of a solution they might use when getting that job done.

Given the user defined above, Carolyn’s “job” is to recover from surgery. There are many solutions that might help her get that job done. Some solutions are better than others. But how do we know? We must understand how our users measure value. According to Ulwick, the way we do that is by understanding our users’ *desired outcomes*. These tell us how our users gauge to what extent a solution helps them get their job done. For Carolyn, she might value the speed at which she recovers, her level of pain, and the extent of knee mobility she regains after surgery and physical therapy. Desired outcomes like these are uncovered during the course of empathizing with users, such as in user interviews.

### 3.4 Core Concept 3: Identify and Test Underlying Assumptions

New product innovation benefits from a diversity of input and feedback. We might think new discoveries and novel products are the result of the lone genius tinkering away in his or her laboratory or office, but more commonly innovation is the result of cross-disciplinary teams working together to achieve a common goal. This is perhaps even more prevalent in the global health context, where a multi-faceted and synchronized approach is crucial for successful implementation. This diversity is crucial for uncovering assumptions made by the innovation team.

Akin to testing a scientific theory, assumptions should be formulated into questions or hypotheses statements, then tested to validate or invalidate what the team is building or believes to be true. These hypotheses might be about the innovation itself (e.g. technical assumptions), about the users (e.g. who the users are or what they need), or about how the innovation might be built or implemented. For instance:

- Are we solving the right problem for our users?
- Do we have the right solution to those problems?
- How can we implement our solution to best reach our users?
- Can our solution be built? Does it depend on new science or new technology?
- Can our solution be sustainably financed, and sustainably fixed when it breaks?

One way to test whether or not the right problem has been identified is to rigorously uncover and test users’ desired outcomes, as described in the previous section on the Outcome Driven Innovation framework. This and the other questions above represent risk that should be iteratively reduced over time to improve the chances of success. The next core concept offers one approach to addressing risk.

### 3.5 Core Concept 4: Adapt to Changes with Continuous Iteration

Most global health initiatives are static and follow the traditional, linear product development cycle. In software development, the traditional approach follows the Waterfall methodology, where requirements are fully specified upfront, the design is executed, then the software is finally built, tested, and shipped to users. While the Waterfall approach can be used successfully to deliver “clean” bug-free code and new features, it does a bad job in helping teams quickly respond to changes in the market, such as evolving user needs, or in managing the inherent uncertainty of the software development process itself (Lucidchart Content Team 2017b).

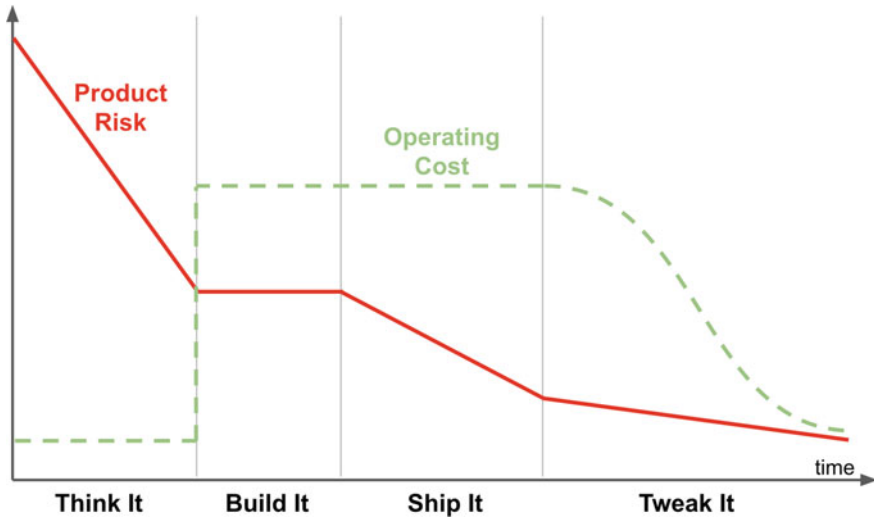
The Agile approach to software development, popularized by the publication of the *Manifesto for Agile Software Development* in 2001 (Beck et al. 2001), was developed in response to the weaknesses of the heavier, more planned Waterfall approach. Several methods implementing Agile principles have been developed over the years, and include Scrum, Kanban, and Extreme Programming (XP) (Lucidchart Content Team 2017a). In general, Agile principles value experimentation and responding to change over following a strict product plan. This helps minimize sunk costs from over-planning or from building too much before testing the product with target users (Koch 2011).

If possible, teams should plan their product development in such a way that the product is developed in steps, where each step offers an incremental increase in value to the user and can be tested. In this way, the build process for each step will be drastically shorter than the build process for the entire finished product. Shorter build times enable teams to test those scoped-down products directly with stakeholders or end users, which provides a vital source of feedback to help guide the team. Regular testing can also help uncover assumptions that had not yet been identified. For instance, perhaps there are local government regulations that influence how a product or service can be sold or delivered, or there are unexpected social factors that affect last-mile implementation. Identifying these assumptions as early as possible helps reduce risk, and iterative product development is a key tool that enables these key insights to occur.

### 3.6 Core Concept 5: Products Have a Lifecycle

The final core concept and its framework are discussed in significant detail, as they provide a useful mental model to help teams better understand the entire innovation process, and allow us to tie together the previous four core concepts.

At the most basic level, products have four stages in their lifecycle: ideation, development, delivery to end users, and end-of-life or sunsetting, where products are taken off the market. Whereas Agile provides teams a set of values and practices to plan, develop, and deliver software during the development phase, it is helpful to



**Fig. 3.1** In the Think It, Build It, Ship It, Tweak It framework, product risk is highest early in the product lifecycle, before the product has been built, tested, and shipped to users. However, the operating cost can be kept at its lowest during this early stage by committing only limited resources (e.g. a small team instead of a large team). Once product risk has been reduced through user testing and prototyping, more resources can be more safely committed (e.g. a larger team) as the product moves into the Build It stage

have a higher-level view of the entire product development lifecycle. This section provides an overview of the product lifecycle using the “Think It, Build It, Ship It, Tweak It” framework, developed by the company behind Spotify, the music app. Readers are referred to the original article (Kniberg 2013) for a more in-depth review of this framework. These four lifecycle stages help us understand important tradeoffs between product risk and operating cost over time.

### 3.6.1 Think It

The goal of the **Think It** stage is to provide the team evidence that their target users do in fact have a significant unmet need, and that the proposed solution might fulfill that need. The evidence must be compelling enough to further develop the product. If not, teams can save time and resources by shifting their efforts elsewhere. Teams should be able to answer by the end of this stage, “Are we building the right product?” To get at this answer, teams should start by defining all their assumptions, as well as criteria to help the team gauge whether or not the product is successful once launched. Basic assumptions that should be addressed include:

- Who is our user?

- What do they need?
- What value will this product provide them?
- What value will this product provide our team (or company) if it's successful?
- Can we build it?
- How will we acquire our users?

Examples of product success criteria include:

- User behavior measures, such as adoption and retention
- Technical measures, such as service uptime, total bugs, and bug severity
- Business measures, such as revenue, cost of customer acquisition or implementation, and length of sales or implementation cycle

Assumptions represent risk, which can be minimized by running tests, usually in the form of building and testing prototypes with potential users. Business and technical risks might also be reduced in this stage using other experiments, like measuring landing page conversion from a Google AdWords campaign that describes the software idea, or by building technical prototypes of the most challenging parts of the software architecture. In general, the more iterations tested, the better the final product (First Round Review 2018). There are a plethora of types of experiments teams can run. For a non-exhaustive list of examples, readers are referred to Jeff Sauro's writing on user research methods (Sauro 2014, 2015). This prototyping approach is extremely important in projects for global health initiatives. In a global health context, the product innovators are often not the same as the end users, even if the team is a local one. These differences in context can lead to a multitude of assumptions that could result in failure of the product or in its sustainable implementation.

Not all assumptions will be satisfactorily addressed during the **Think It** stage. It is a judgment call by the team for when to commit the resources to move forward with building the product. In addition, even if some evidence was acquired for certain assumptions, these tests often extend into the next lifecycle stages as the product is built and delivered. This is a different approach from many current global health initiatives, which rely on more traditional, linear development. The challenge is to strike a balance between enough good evidence to support the potential of a product and the speed of development. This ensures that the innovation cycle is faster and provides relevant, tangible, and timely interventions and products for the global health community.

The Innovation Workshop presented later in this chapter is designed to take place during the **Think It** stage, although it is applicable to the other stages as well. We come back to this point later in the chapter.

### 3.6.2 *Build It*

If the team agrees what's been prototyped is valuable enough to build, more resources are committed and the team enters the **Build It** stage. Operating costs increase as headcount goes up, but because the team hasn't yet shipped product to real users, they can't be sure they're further reducing product risk.

During this stage, it helps to narrow focus and prioritize efforts, as one team can't build everything, all at once. Instead, teams should prioritize work on the hardest, highest-risk assumptions first. A simple analogy is drug development—while it's easier to design the product packaging for a new drug (bottles, boxes), pharmaceutical companies instead focus first on testing whether or not a drug candidate can deliver on its target health outcome. Why develop the packaging if they can't build the drug? Teams should force-rank their assumptions defined in the Think It stage based on risk, with the greater-risk items having higher priorities. If two risks are gauged roughly equal, then the risk that is lower-effort to test should be prioritized higher to increase the speed at which teams can address these risks.

During the **Built It** stage, it's also important to develop the product from the start with a focus on quality: well-written code and robust, performant product components. However, this doesn't mean delivering the perfect product. The balance is in delivering a narrow product with the fewest possible features such that it delivers compelling value to the user, but built in a way that it won't catastrophically fail during usage. Once a small increment of working product is built, it's ready to be delivered to users.

### 3.6.3 *Ship It*

The **Ship It** stage is when teams start to roll out product to real users. To manage risk during software development, teams often rollout products in three phases of increasing penetration into the user base: alpha, beta, and generally available. This requires teams to listen to user feedback, and iteratively fix bugs while maintaining focus on achieving their previously defined product success criteria. Particular success criteria, like service uptime or customer satisfaction, can help identify whether or not there are new problems to address as more users adopt the product. While there are no standards for how many users to reach during each rollout phase, the rule of thumb is to generate sufficient evidence for the success criteria that the team can confidently rollout to more users.



### 3.6.4 *Tweak It*

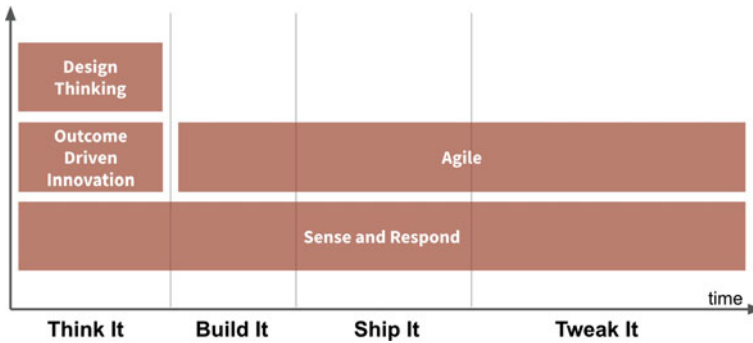
If success criteria continue to be met, and any performance issues that arose during rollout are addressed, the product or feature has achieved general availability when all target users have it available for use. At this point, the feature moves into the **Tweak It** stage. In this stage, the product or feature might be iteratively improved upon, or simply moved into a maintenance mode and supported by only occasional bug fixes. The operating costs in this stage decrease over time as product developers shift their focus to other projects. Products only leave this stage if they're fully deprecated or sunset.

In this model of product development, the operating cost (the cost it takes to build the product) is lowest during the Think It stage: there might only be a small team (2–3 people) developing prototypes, after which a larger team is hired to develop the product in the Build It and Ship It stages. However, the product risk is highest during the Think It stage as the team is not yet sure if they're solving the right problem or have the right solution. While this sounds like a grave challenge, in reality this presents a great opportunity to learn as much as possible (to reduce product risk) while the costs are lowest (before significant funds have been invested). In this way, by prioritizing research and development on the highest-risk assumptions, teams can incrementally collect evidence that helps them either change course or end the project entirely.

## 3.7 Core Concepts Summary

These core concepts and their corresponding frameworks are not mutually exclusive, and are often used in tandem during product development (see graphic below). For example, qualitative user interviews provide the foundation for formulating user need statements. These user need statements can be re-written as outcome statements following the Outcome Driven Innovation framework, which can then be tested to provide more quantitative data to guide the team. Assumptions about the product, user, and implementation plan can then be systematically identified, and a research plan developed using the Sense and Respond model. Next, Agile practices like iterative development and regular user testing can help keep the product relevant and focused on meeting the users' needs identified earlier. Finally, the “Think It, Build It, Ship It, Tweak It” framework helps us keep in mind that upfront testing of our hypotheses is very cost-effective, and even after a product has been implemented with its target users, a long tail “Tweak It” phase is often necessary to continue adjusting the product so it continues to meet its users' needs over time (Fig. 3.2).

Each framework contributes valuable methods to the innovation process. But it can be difficult to reconcile them into a cohesive model of effective product innovation. Together, these core concepts and the accompanying workshop exercises presented below provide both a high-level mental model for successful product development,



**Fig. 3.2** The five frameworks fit together across the product lifecycle

as well as the tactical detail necessary to help teams implement innovation best practices. The goal is to enable teams to zoom out to understand where they are headed, then zoom back in to take measurable steps toward achieving their goals.

### 3.8 Innovation Workshop

In general, innovation is usually seen as more art than science, but in reality the innovation process can be broken down into concrete steps. By following a best practice innovation process, teams might both increase their chances of success as well as better repeat those accomplishments, thus creating a sustainable cycle of innovation. This workshop presents a set of eight exercises designed to generate discussion and produce a diverse set of ideas used to develop an actionable research plan for validating new product ideas.

During the workshop, teams identify target users, describe and prioritize their users' unmet needs, and then brainstorm features that might fulfill those needs. Next, teams identify any underlying assumptions they've made and then rank those assumptions by level of risk should their assumptions prove invalid. Finally, teams develop a user and market research plan for testing their highest-risk assumptions. The exercises are presented together as a single cohesive workshop, but each exercise can optionally take place individually over an extended period of time. The eight exercises are as follows:

1. Define your objective: 10 min
2. Identify target users: 5 min
3. Define users' needs: 10 min
4. Prioritize desired outcomes: 25 min
5. Brainstorm features: 35 min
6. Decide: 10 min
7. Identify assumptions: 15 min

## 8. Define the research plan: 10 min.

Each exercise is covered in more detail below, and in the accompanying workshop slides (Moses 2018). The workshop is designed for 2 hours of active time for exercises, with an additional hour for facilitator-led instruction to teach participants the five core concepts, for a total of 3 hours to complete the workshop. If there is more than one team participating in the workshop, there is an optional 20 minute exercise at the end of the workshop for teams to take turns presenting their results.

These exercises are not only for those interested in starting a commercial business or working in a large enterprise, but are equally as applicable for improving global health interventions. While the workshop is framed around product innovation, a “product” can be anything that meets a user’s needs. Products need not have large user bases, nor do they need to be commercialized. A product might be physical, digital, or service-based. It can be proprietary or open source.

The workshop is designed to take place during the **Think It** stage of product development to give teams a foundation to kickoff new product innovation with the greatest chance of success. However, if teams have already progressed to prototyping, or have an existing product or service in use by their target population, they can still benefit from these exercises. Product functionality and product-market fit can always be improved. These exercises can direct more advanced teams to systematically assess what desired outcomes their product should be meeting, identify outstanding assumptions, and determine a research plan to test those assumptions.

The workshop is best conducted for a team of 5–7 participants. No prior entrepreneurial or product development experience is required. While more than one team can participate in the same workshop, each team is best served by a single dedicated, trained facilitator who is familiar with the instructions. Each team should have ample colored sticky notes, 5–7 sharpie markers, blank printer paper, and a large whiteboard for assembling their Outcome Map. For the Solution Sketches, it is helpful to use larger, 6 × 8 in. sticky notes. If that is not available, participants can fold a piece of printer paper in half. Slides for facilitating the workshop are provided in open-source format on Github (Moses 2018).

## 3.9 Workshop Output

The output of the workshop is organized into an outcome-based product roadmap, or Outcome Map. The Outcome Map is a simple visualization that combines a target user’s desired outcomes, potential features that might meet those outcomes, assumptions about the user, product, or market, as well as ideas for testing those assumptions. It enables teams to collaboratively identify and refine a series of practical steps for effective product innovation (Fig. 3.3).

The Outcome Map is prioritized first by “Opportunity” (a measure of user’s needs), then by risk. Teams may choose to prioritize the Outcome Map in other ways, such as by a measure of financial sustainability or technical feasibility. However, even if

DESIRED OUTCOMES	FEATURES	ASSUMPTIONS	RESEARCH PLAN
Highest priority	Two orange sticky notes	Two pink sticky notes	Two yellow sticky notes
Medium priority	One orange sticky note	Three pink sticky notes	Two yellow sticky notes
Lowest priority	One orange sticky note	Two pink sticky notes	Two yellow sticky notes

**Fig. 3.3** The outcome-based product roadmap, or Outcome Map, that teams will develop in this workshop. The map can be assembled on a whiteboard or table using colored sticky notes

a product or feature is technically feasible, it might not be financially sustainable if there is not strong demand, i.e. if we do not target our users’ significant unmet needs. For this reason, teams focus first on prioritizing by Opportunity to ensure they’re meeting their user’s greatest unmet needs.

To save time during the workshop and aid in presenting instructions to participants, it is helpful for facilitators to prepare the basic four-column structure of the Outcome Map on a whiteboard before the workshop begins. The facilitator might also provide a completed example Outcome Map to give participants better direction for how to define items within each column.

### 3.10 Workshop Scenario

The workshop provides participants the following shared scenario:

#### Laboratory studying interventions for treating frailty

Imagine we’re working in a lab studying interventions to improve health outcomes for frail individuals, and preventing frailty among adults 65 years and older. Frail individuals have less ability to cope with acute stressor events like surgery or emergencies, which can lead to worse health outcomes than those who are not frail. Specifically, we’ve found that a group exercise program improves health outcomes, and we’ve published our results. A review article was recently published that cited our paper and backed up the evidence with additional articles supporting our findings. Our intervention is comprised of 5 upper- and lower-body exercises, done for 60 seconds each, 3 times per week, for 12 consecutive weeks. After reading an interesting article in Forbes describing significant business opportunities in this space, our principal investigator sets an objective to guide our team to productize our intervention:

Launch a product or service that improves functional capacity and mobility for frail adults 65 years and older.

**Table 3.2** Good and bad objective statements

Good objective statement	Bad objective statement
Launch a product or service that improves functional capacity and mobility for frail adults 65 years and older	Build a mobile app for seniors that delivers a workout plan that tailors 5 exercises for the user, and allows them to schedule and track their performance over time

## 3.11 Workshop Exercises

### 3.11.1 Exercise 1: Define Your Objective

Conventionally, in many companies product planning happens in a Waterfall fashion: the leader defines the vision and a 3–5 year product roadmap, and his or her leadership team translates that into each of their respective team’s goals. The problem with this approach is that it doesn’t take into account the inherent nonlinearity of product development, caused by the iterative nature of building, testing, and learning from users as well as changes in the market that might come from new regulations or the entry of a new competitor. Teams need a way to navigate these challenges without being constrained to any particular solution, but are provided enough guidance that they can remain focused on delivering their target user outcome. This can be accomplished with a well-defined Objective statement, which describes the outcome to be achieved but not how the team will achieve it (Gothelf and Seiden 2017) (Table 3.2).

### 3.11.2 Exercise 2: Identify Target Users

Imagine you were going to give a talk at a conference. When preparing, you might ask yourself, “Who is my audience?” Similarly, we need to understand our audience—our user when developing a product. In this workshop, we provide the following simplified user persona as part of our scenario:

#### **Frail 85-year old after surgery**

I just had surgery and am returning home. Because I’m frail, I might have worse health outcomes. My goal is to recover as quickly as possible and to return to my previous state of health.

Participants should reference this user persona for the remaining exercises.

### 3.11.3 Exercise 3: Define Users' Needs

Participants are asked to brainstorm *desired outcome* statements for the target user persona defined above. For example, these statements might be:

Desired outcome statements

- Minimize the time it takes to recover from my surgery
- Minimize the likelihood I'm injured when exercising
- Maximize the amount of fat I lose when exercising

In reality, a user might have 100–200 desired outcomes, but for the purpose of this workshop, participants are asked to generate as many as they can in 5 minutes. Facilitators are urged to read Ulwick's book (Ulwick 2016) for more detail on this methodology.

### 3.11.4 Exercise 4. Prioritize Desired Outcomes

In Outcome Driven Innovation, user need statements are assembled into a survey, which is then fielded to hundreds of target users to provide more quantitative evidence to help teams understand which needs are most unmet. This is achieved by asking survey respondents to rank on a Likert scale their level of agreement with two key questions for each desired outcome:

1. When [doing the job or activity], how **important** is it for you to [achieve desired outcome]?
2. When [doing the job or activity], how **satisfied** are you with the way you currently [achieve desired outcome]?

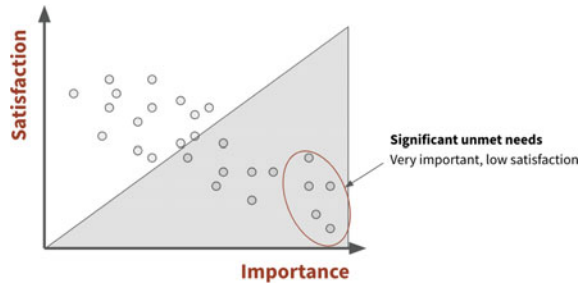
For instance:

1. When recovering from surgery, how **important** is it for you to recover quickly?
2. When recovering from surgery, how **satisfied** are you with how fast you're currently able to recover from surgery?

By aggregating responses to these two questions, teams can visualize each desired outcome as a coordinate in a two-dimensional plot of Importance vs. Satisfaction. Those desired outcomes that rank very important but very dissatisfied reveal key opportunities for innovation (Fig. 3.4).

In lieu of conducting a large survey to collect quantitative data on users' ratings of satisfaction and importance, we use a quicker, easier method during the workshop. First, facilitators ask teams to force rank the desired outcomes they brainstormed in order of estimated importance to the user, from high to low. The facilitator records the rank order of each outcome, then asks the team to re-order the outcomes by estimated user satisfaction, from low to high. By adding together these two rank orders, and using the importance rank to break any ties between two equal sums, teams

**Fig. 3.4** A plot of desired outcome statements can reveal significant unmet needs



can approximate a relative “Opportunity Score.” The desired outcome statements can now be placed on the Outcome Map in ascending order, where the smaller the Opportunity Score, the greater the users’ unmet need, and the higher on the map the outcome will be placed. If they haven’t already, facilitators ask the team to write the desired outcome statements on sticky notes and then place them on the left side of the Outcome Map in one vertical line.

At this stage, the participants have defined their user’s needs and estimated the opportunity for innovation for each need based on how important and dissatisfied the user is in achieving that need. Next, we take these results and brainstorm features that might meet our user’s needs.

### 3.11.5 Exercise 5: Brainstorm Features

What features will help our users meet their desired outcomes? Facilitators should remind participants to develop features that would help the user achieve their desired outcomes—not just the features they like. This section is broken down into three key exercises, adapted from the Design Sprint method developed at Google (Knapp et al. 2016), and described in more detail in the accompanying slide deck:

1. **Brain dump:** write down anything that comes to mind on a sheet of paper
2. **Crazy 8’s:** rapid low-fidelity sketches of eight concepts, or a user flow of 1–2 concepts
3. **Solution Sketch:** more detailed sketch of a single concept.

Once the Solution Sketches are complete, facilitators ask teams to place them on the Outcome Map next to the desired outcomes they help achieve. If a feature achieves more than one desired outcome, then move that outcome statement into the same row.

### **3.11.6 Exercise 6: Decide**

Time permitting, facilitators can setup a “Gallery Walk”, where participants observe each other’s features, then “dot vote” on the best ones. In this way, teams can narrow down the number of potential features they’ll need to consider in the remainder of the workshop (and later, build during product development). This workshop does not detail this method, but readers are referred to Chapter 10 in the Sprint book (Knapp et al. 2016) for an explanation of this process. In this workshop, participants simply present their Solution Sketches to team members and vote on which one or two designs to move forward with.

At this stage, teams have defined their user’s needs and brainstormed potential solutions for the most significant unmet needs given their estimated Opportunity Scores. The result of this exercise is to build a shared understanding of other team members’ Solution Sketches, and to decide which one or two designs will become the focus for the remainder of the workshop. For instance, if a team of five produces five Solution Sketches, the team will vote to move forward with only one or two of the designs. In this way, teams can focus their efforts for the remainder of the workshop, where they exhaustively identify their assumptions, prioritize those assumptions by risk, and outline a research plan to test these assumptions.

### **3.11.7 Exercise 7: Identify Assumptions**

As a group, have participants discuss the risks associated with delivering their chosen design(s). These risks might include:

- What challenges could the team face during product development? During implementation?
- What must be true about:
  - Our users?
  - Our technology?
  - State or federal regulations and policy?
  - The business model? Market?

Participants record assumptions on sticky notes and place them on the whiteboard, then force rank each assumption by risk. To rank by risk, ask: “How bad would it be if we got that assumption wrong?” For example, if your feature relies on access to user data, can you feasibly gain access to this data, and if so, how? How bad would it be if the technical approach you planned to use to access this data didn’t work? If access to user data is critical to the functioning of your application, then this is a high risk and should be prioritized towards the top of your list. If your team also planned on a particular revenue model, but have not yet tested it, then that presents another risk. However, in comparison to the technical risk of your product not functioning if your team is unable to access user data, then this revenue model risk would be



prioritized lower. In this case, by addressing that particular technical risk first, at least your team would have a functioning product, and perhaps could find a new revenue model if the one originally planned did not work out.

### ***3.11.8 Exercise 8: Define the Research Plan***

There are a plethora of user, market, and product research methods available to the team to validate their assumptions. Participants are asked to write one experiment per sticky note to address each assumption on the whiteboard. While the details of these methods are out of the scope of this book and workshop, a number of methods are listed in the accompanying slide deck for this exercise (Moses 2018). For example, teams might need to better understand their users, and could decide to run an ethnographic study or diary study to learn more about how their target users work. If a team needs to learn more about their product offering, they might build technical prototypes to investigate technical feasibility, or design a clickable prototype of screens to test the usability of a new product workflow.

## **3.12 Workshop Wrap Up**

By the end of the workshop, teams have assembled an outcome-based product roadmap, or Outcome Map. Potential solution ideas are prioritized by users' greatest unmet needs, and the research plan is prioritized by risk. At this point, teams are still in the Think It stage of product development. Armed with a research plan, teams have greater direction for what to do next. However, facilitators should caution participants to carefully consider how much research they conduct. While teams will never achieve zero risk, they should strive to collect enough evidence to confidently decide whether to end the project, pivot to a new idea, or move forward with product development of their proposed solution.

In this workshop, teams focused primarily on solving for users' unmet needs, but participants are urged to think more holistically about a few additional factors when designing real-world applications. To successfully launch a product, even for global health and grant-funded projects, teams should carefully consider all of the following factors (Gerber 2019):

- **Desirability**
  - What significant unmet needs do our users have?
- **Feasibility**
  - What assets do we have (people, data, skills)?
  - Can we build it?
  - How long will it take to build it?

- **Viability**

- Can we find a group of users or sponsors willing to pay?
- Does the revenue or sponsorship cover the cost to build, sell, and service this product?

For example, even if we were to identify a large market of potential users and employed a team capable of building and delivering the product, there is risk no one adopts it if the product doesn't solve the users' significant unmet needs. By weighing these considerations, teams can better prioritize what to build.

### 3.13 Summary

Innovation is not a linear path, and there is no one right way. However, we *can* break down the process into faster, lower-cost chunks and then thoughtfully reflect on the outcomes along the way. As taught in this chapter and workshop, it is valuable to identify all high-risk assumptions upfront, then design experiments to test potential critical failures early on in the product development process. The Outcome Map is never complete, and is best treated as a living document. Similarly, product development never ceases—that is, until the product has achieved its goals and it comes time to deprecate it.

Building local innovation capacity by encouraging and teaching innovation best practices is only part of the puzzle. During debriefs with workshop participants at the 2018 Khon Kaen University Datathon in Thailand, the author learned that a cultural shift is also a critical component of progress. This is perhaps the greatest difficulty for a team, organization, or society to accomplish. In medicine, cultural change can be at the root of implementation challenges for evidence-based medical practices, despite clear results of positive outcomes (Melnik 2017; Best et al. 2013; Rice et al. 2018). The reasons might seem intractable: limited resources, competing priorities, the need for leadership support and training, hierarchical relationships, a culture of shame for failure of new initiatives, among other factors.

But there is hope. In Estonia, the small Baltic country in former-Soviet Union, half of its citizens did not have a telephone line 30 years ago, yet today Estonia represents one of the most digitally-advanced societies in the world (A.A.K. 2013). Toomas Hendrik Ilves, the fourth president of Estonia, has said that Estonia's success in technological innovation has not been so much about using new technologies, but about “shedding legacy thinking” (A.A.K. 2013).

By rethinking how high-income countries might best catalyze global health improvements, we might derive new, effective, low-cost approaches. Building local innovation capacity through transferring knowledge, processes, and encouraging a culture shift about risk taking and failure offers an alternative. Given the inherent difficulties of culture change, local innovation capacity might be further supported by training teams in strategies for organizational change management techniques

(Kotter and Schlesinger 2008; Kotter 2014), in addition to innovation best practices. As a global community, we are still learning how best to catalyze global health improvements. Skill building in innovation best practices and organizational change management offers an alternative, low-cost approach to driving global health improvement.

## References

- A.A.K. (2013). How did Estonia become a leader in technology? *The Economist*. Retrieved 3 July 2019.
- Barry, M. (2010). *POV Madlibs*. Hasso Plattner Institute of Design at Stanford University. Retrieved 25 April 2019.
- Beck, K., Grenning, J., Martin, R. C., Beedle, M., Highsmith, J., Mellor, S., van Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham, W., Kern, J., Thomas, D., Fowler, M., Marick, B. (2001). Manifesto for Agile Software Development. Agile Alliance. Retrieved 15 Nov 2018.
- Best, A., Saul, J., Willis, C. (2013). Doing the dance of culture change: Complexity, evidence and leadership. *HealthcarePapers*, 13(1).
- Brown, T. (2009). *Change by design: How design thinking transforms organizations and inspires innovation*. HarperBusiness.
- Dam, R., & Siang, T. (2019). What is design thinking and why is it so popular? Interaction Design Foundation. Retrieved 24 April 2019.
- Dam, R., & Siang, T. (2019). Stage 2 in the design thinking process: Define the problem and interpret the results. Interaction Design Foundation. Retrieved 25 April 2019.
- Donaldson, K. (2014). 5 innovations in global health: Maybe not what you were expecting. *D-Rev Blog*. Retrieved 3 July 2019.
- First Round Review. Six steps to superior product prototyping: Lessons from an Apple and Oculus Engineer. Retrieved 15 Nov 2018.
- Gerber, J. (2019). How to prototype a new business. IDEO U. Retrieved 3 July 2019.
- Gibbons, S. (2019). User need statements: The 'define' stage in design thinking. Nielsen Norman Group. Retrieved 25 April 2019.
- Gothelf, J., & Seiden, J. (2017). *Sense and respond: How successful organizations listen to customers and create new Product*. Harvard Business Review Press.
- Juma, C., & Yee-Cheong, L. (2005). Reinventing global health: The role of science, technology, and innovation. *The Lancet*, 365(9464), 1105–1107.
- Knapp, J., Zeratsky, J., & Kowitz, B. (2016). *Sprint: How to solve big problems and test new ideas in just five days*. Simon & Schuster.
- Kniberg, H. (2013). How spotify builds products. Retrieved 15 Nov 2018.
- Koch, A. (2011). Expert reference series of white papers: 12 advantages of Agile Software Development. Global Knowledge Training, LLC.
- Kotter, J.P. (2014). *Accelerate: Building strategic agility for a faster-moving world*. Harvard Business Review Press.
- Kotter, J. P., & Schlesinger, L. A. (2008). Choosing strategies for change. *Harvard Business Review*. Retrieved 3 July 2019.
- Lucidchart Content Team. (2017a). What is Agile methodology? *Lucidchart Blog*. Retrieved 5 July 2019.
- Lucidchart Content Team. (2017b). The pros and cons of Waterfall methodology. *Lucidchart Blog*. Retrieved 5 July 2019.
- Melnyk, B. (2017). Culture eats strategy: Why nursing and other health professions haven't fully embraced evidence-based practice. *Medscape*. Retrieved 3 July 2019.

- Moses, C. (2018). Innovation Workshop. Github repository: <https://github.com/cmoses8626/innovation-workshop>.
- National Academies of Sciences, Engineering, and Medicine. (2017). Global health and the future role of the United States. In *Catalyzing innovation*. The National Academies Press.
- Rice, H. E, Lou-Meda, R., Saxton, A. T., et al. (2018). Building a safety culture in global health: Lessons from Guatemala. *BMJ Global Health* **3**, e000630.
- Sauro, J. (2014). 10 essential user experience methods. MeasuringU. Retrieved 25 April 2019.
- Sauro, J. (2015). 5 types of qualitative methods. MeasuringU. Retrieved 25 April 2019.
- Syed, S. B., Dadwal, V., Martin, G. (2013, August). Reverse innovation in global health systems: Towards global innovation flow. *Globalization and Health* **9**, 36.
- Ulwick, A. W. (2016). *Jobs to be done*. Idea Bite Press.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

