A Study Design Design Process

Andrew J. Ko, Sally A. Fincher

Chapters in this Handbook provide foundations for conducting research, ranging from theories of learning and knowledge to methodological tools common to computing education research. In this chapter we discuss the research process itself. We deconstruct one skill, focusing on *study design*, a critical start to many kinds of computing education research.

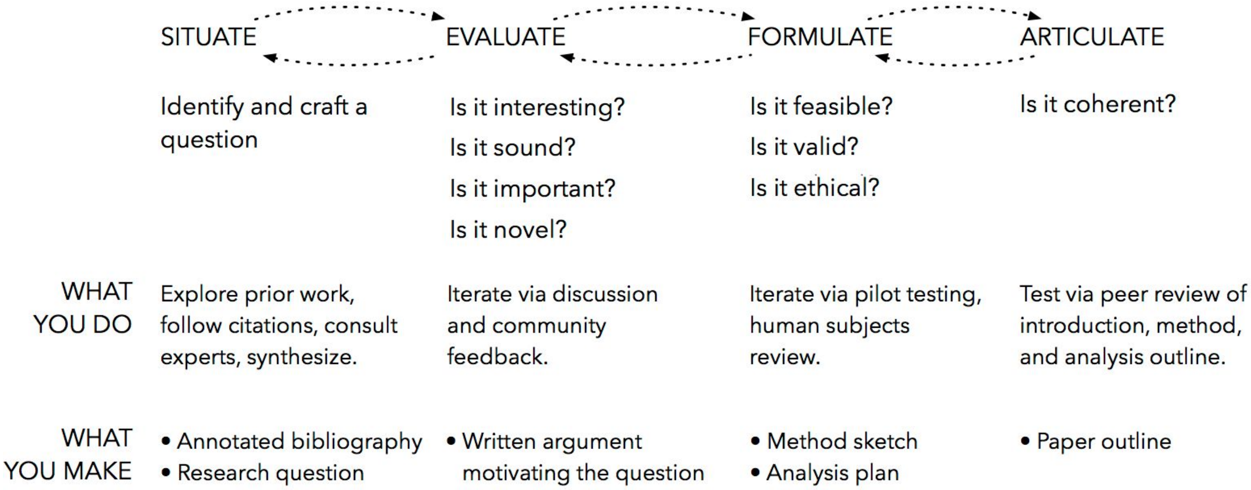
Sadly, a book chapter is inherently inadequate for actually learning to design studies. Becoming expert at anything, study design included, requires extensive deliberate practice, and this chapter cannot give you that practice. Instead, this chapter gives a framework to *structure* your deliberate practice, helping you to more effectively select what you practice, how you practice it, and how you seek feedback on it. This chapter is therefore scaffolding for skills that will take years to develop.

Our approach to structuring your practice is to frame study design as a *process*, and in particular, a *design* process (Lawson, 2006), involving iteration, feedback, prototyping, including divergent generation of new ideas and convergent refinement and selection of ideas. As in common with all design activities, the experience of designing studies is one that almost always begins with immense ambiguity and ends with clarity (in the case of study design, in the form of data collection instruments, procedures and analysis plans).

Note that **empirically**, **methodologically**, and **theoretically**, study design processes are not inherently different from the numerous other fields from which computing education research borrows its theories, methodologies, and empirical epistemologies. What *is* different is the knowledge and strategies researchers must use to design a successful study. For example, one way that study design is hard is that *what* you are designing is mostly invisible. Unlike designing cars, clothing, or devices, “studies” are not inherently visible or tangible. Therefore, a critical part of effective study design is making studies *visible*, by prototyping them in forms you can apprehend, critique, and refine. This is true of studies of anything, but in the design of studies about the learning and teaching of computing, the prototypes of studies we make are all domain-specific. Therefore, knowledge of the structure of computing education studies, and how to make these types of studies successful, is domain-specific.

In this chapter we will mostly talk about studies devised and executed by a single-researcher (or small team). However, one type of study often found in Computing Education is the multi-national, multi-institutional (MNMI) study, where data is gathered in many countries and many institutions, to investigate questions across different cultures and different teaching practices. The best-known example is “the McCracken study” (McCracken et al., 2001), which was replicated in 2013 (Utting et al., 2013), but many groups, especially those associated with the ACM ITiCSE conference use the model. (Fincher et al., 2005) provides an overview.

Finally, before embarking on a tour through elements of study design process, Figure 4.1 presents a map of the process we will discuss. This process includes key activities and questions to answer about the output of those activities, a checklist to guide study design practice from an initial idea to completion and report. The outcome of each step in the map is an artifact that will contribute to the overall design. After step one you should have the components of a literature review (perhaps written as an annotated bibliography). After step two you should have an argument motivating the question. After step three you should have a method sketch and pilot of your analysis. After step four, you should have a paper outline that allows you to execute your plan.



	SITUATE	EVALUATE	FORMULATE	ARTICULATE
	Identify and craft a question	Is it interesting? Is it sound? Is it important? Is it novel?	Is it feasible? Is it valid? Is it ethical?	Is it coherent?
WHAT YOU DO	Explore prior work, follow citations, consult experts, synthesize.	Iterate via discussion and community feedback.	Iterate via pilot testing, human subjects review.	Test via peer review of introduction, method, and analysis outline.
WHAT YOU MAKE	• Annotated bibliography • Research question	• Written argument motivating the question	• Method sketch • Analysis plan	• Paper outline

Figure 4.1 A study design design process spanning four iterative phases that come before executing a study.

Although we have presented this sequentially, be aware that study design is hardly ever an orderly process, don't be deceived by a neat representation. Good study design is an inherently iterative activity and you can expect to loop through stages, revisiting and refining, often many times. With practice, some parts may be collapsed and taken together, and with different kinds of study (as in software engineering methodologies) sometimes a different order makes better sense than others.

This necessary iteration, combined with the significant effort required to answer the questions, is why high-quality study design can take so long. Throughout this chapter, we will discuss strategies for streamlining all of these activities.

1. Situate: identify a research question

As portrayed in Figure 4.1, the first critical aspect of a study design process is identifying a research question by situating your thinking in prior work. And although it is often called "finding a question," as with any design process, the initial stage is largely about exploring and framing a problem. Before you can formulate a question, you have to have an *idea* for a question. This is an inherently *generative* activity, and like anything creative, can require inspiration from unexpected and diverse places.

Research—of any kind—is not a solitary activity, but constructed within a community. Therefore, ideas for questions can come from numerous places, people, and ideas. For example, you may find yourself observing someone try to solve a programming problem in a class through brute force, compelling you to ask why they've chosen that strategy over another. This might lead to you theorizing about strategy, which then leads you to dive into literature about strategy.

Novice study designers in computing education often begin with an idea for a *solution* to a problem. While this is usually premature, from solutions, one can derive questions, and even theories that underlie those questions. For example, imagine you were trying to teach basic ideas behind supervised machine learning with decision trees. Your first instinct might be to come up with an explanation for information gain and use a study to test whether that explanation is effective. Instead of leaping from a solution to a study design, first ask , "what is hard about the concept of information gain and what about the explanation would make it easier?" The answer to *that* question is a theory that explains why your explanation of information gain might work, which is a more rigorous way to devise predictions about the explanation's effects and ultimately test them. This is critical to understanding the causes underlying problems, and critical to designing a good study.

You can also find research questions in the "future work" discussion of a recent publication, a dissertation, or an interesting blog post about the learning or teaching of computing that poses, but does not answer, a question. Many research questions come from conversations with other researchers, such as those with colleagues, advisors, or other researchers, in PhD meetings, at conferences, or online. Whatever the source, ideas for research questions are unlikely to come purely from your own mind, but from the richness of the writing, people, and phenomena you surround yourself with.

To illustrate some of these circuitous paths to questions, here are two brief origin stories. In 2009, the first author of this chapter added a line to the list of interests on his faculty website that said something like "Designing playful interactions with code." A prospective doctoral student, Michael Lee, saw that statement, decided to apply, and ultimately joined the first author's lab. Now, the first author was not doing any work on "play," and he did not at that time have a vision for what it might mean, but was curious to find out. When Mike arrived, he had numerous conversations with the first author which generated hundreds of different ideas for research questions. After dozens of discussions, reading hundreds of papers, and extensively refining his thoughts, Michael eventually converged on the question, "*What is the effect of framing a compiler as a collaborator rather than an authority?*" That triggered a design process that ultimately led to Gidget (Lee et al. 2014), and, eventually, numerous published studies.

Another story. In 2003, the second author replied encouragingly to an email from a prospective PhD student. Matt Jadud started his doctoral studies later that year. At first, there was no clear direction to the work. Matt was interested in classroom interactions and had many hours of video, which we initially thought to analyse. Over the next few months, in many meetings and many discussions on many walks, the work moved away from focusing on what teachers were doing to support students—or what students were saying to each other, to support their programming practice—but to the *invisible interactions* they were having with the compiler. This led to Matt instrumenting the BlueJ environment and writing some of the very first papers that quantitatively identify novice students' compilation behaviours (Jadud, 2006).

There are three things to notice about these stories: one is that these were not activities conducted in isolation: we all brought a lot "into the room", in the form of observations, literature, and discussions between ourselves and with others. The second is that the process of identifying a research question was not systematic: it was highly social, highly iterative, and highly unpredictable. Third, the exploration in these stories was not random or haphazard, it was *informed:* by prior research, by the expertise of advisors and collaborators, and by the broader community in which these students were learning. This stage of research can be frustrating and feel fruitless (and it sometimes is), but informed, collaborative exploration is a necessary part of study design.

Research questions *can* be derived more mechanistically, most notably from theories. For example, one theory often cited in computing education research literature is Cognitive Load Theory (Sweller et al. 2011), which presents the concept of "extraneous cognitive load." Cognitive Load Theory claims that people have a fixed capacity for cognition, and if a task contains too much extraneous load, it is poorly learned as it leaves no room for acquiring the intrinsic and germane elements. By simply interrogating the *words* in these ideas, we can mechanically extract several potential research questions:

- Is the claim about extraneous load true?
- What is "load"?
- What counts as "extraneous" load?
- What determines a task's "capacity" for promoting learning?
- How can one measure "extraneous" load?
- How can we distinguish extraneous load from other types of load?
- How does extraneous load interact with prior knowledge?

New theories can also be a source of questions. Computing education research, as a distinct research area, needs to define domain-specific theories about how people learn computing, what it means to know programming languages and how to program, and what constitutes programming and software engineering expertise. All of these are important research questions. Questions may also arise from features of disciplinary practice—and computing is particularly rich in this regard. The development and deployment of new tools, computing environments and the compilation of ever-larger datasets in computing research are all rich sources of investigation. For example, what does it mean to understand machine learning sufficiently to build software that leverages machine learning? Theorizing about this knowledge, developing instruments to assess this knowledge, and designing new ways of helping people acquire this knowledge are fruitful areas for computing education research.

2. Evaluate: is it any good?

As we portrayed in Figure 4.1, identifying an idea and formulating a research question is just one part of a longer process. Not all questions are equal, some cannot be researched adequately, some are too broad, others too narrow, and some are beyond your skill or resource to answer. So, when you have a question, the next step is evaluative: is it any good?

2.1. Is it interesting?

One of the first things to evaluate is whether a question is interesting *to you*. This might seem like an odd criterion, but it's actually quite important: if you are not interested, why should anyone else care? If you're not interested, what is going to motivate you to do all of the remaining work that the research will require? Curiosity is the fuel of research, and so regardless of how important, timely, or novel the work is, it's really not worth *you* doing it if you don't really care about the answer.

Of course, interest and curiosity are not always so easy to judge. Perhaps you have multiple diverse interests. What if you might *become* interested in a question after further reading? Perhaps your interest is driven not by the content of a question, but more in the process of answering, or engaging with the community of people who might be interested in the answer. Because of these nuances, judging interest, and selecting one question over another, is often a very personal endeavor that requires reading, reflection, discussion with mentors and advisors, and engagement with research communities.

If it's interesting to you, is it interesting to anyone else? Research is never a private activity, it lives in interaction with a community, building on other's work and making itself available for use by others. When thinking about study design as a *design* process, it is instructive to think of the product: who is the audience for your work? It might be a supervisor, attendees of a particular conference, or readers of a particular journal. Each of these will have a different focus. Working out why they might be interested in your work, in what you have to say, is part of the design.

While we cannot prescribe a process for gauging interest, we can make some general statements about the arc of judging interest: reflect, ponder, and ruminate about your chosen area, and make sure to talk potential audiences about what they find interesting about the question. Without some feedback, and the necessary iteration on a question, you risk the question being uninteresting to anyone. If that happens, no matter how rigorously you answer it, you risk no one caring about what you have discovered.

Do not iterate on this step—or any other—forever. As we discuss in the following sections, there are many other factors to consider, and they may force you to abandon a question interesting to you or others.

2.2. Is it theoretically sound?

Suppose you found a question that’s interesting. The next step is to reflect on whether your question is *theoretically sound*. Soundness is fundamentally about whether the ideas in the question rationally leverage everything we already know about learning, education, and computing.

To illustrate, let’s consider a simple research question, “*Do debugging tools help students learn to code better?*” One process for critiquing a question’s soundness is to deconstruct the ideas in its words. For example, the question starts with “*Do tools help?*” and implies that the answer is either a “Yes” or a “No.” We know from prior work on learning that the answer is rarely binary, but rather “it depends.” Similarly, consider the phrase “help students.” What kind of help? There’s a difference between learning, finishing an an assignment, fixing a bug but not understanding the fix, etc. This question mostly ignores these factors, which means it ignores prior work, making it theoretically unsound.

Because theoretical soundness concerns how consistent a question is in respect to prior work, judging soundness means *knowing* prior work: the more you read, the more you’ll be able to judge. What can make this difficult in computing education research is that the number of sources for knowledge is vast. We have few dedicated research conferences, handfuls of practitioner conferences, and multiple journals. But there are dozens of venues in education research and learning science, all of which are also important to the theoretical soundness of our work. Many other areas of computer science are also adding ”education tracks” to their conferences. Judging soundness therefore requires monitoring many sources for new evidence, as well as mining many sources for existing evidence.

2.3 Is it important?

Another consideration is the extent to which your question is *important*.

Importance can be judged by many different stakeholders along many dimensions. For example, at the smallest level, is it important to you, your advisor, and your collaborators? It might be important because of funding tied to answering it, because of some personally important goal, or it may be of strategic importance to the lab or institution you are part of. There maybe a mismatch between student and advisor goals: because doctoral students depend on the advice, attention, and resources from their advisors, a supervisor's judgement of the importance of your question can matter a great deal in getting the resources you need to answer it. This can be more acute in computing education research, since there is less basic research funding for broad exploration and much more funding for the specific implementation of educational interventions, which can constrain what resources are available.

At a broader level, research communities judge importance against their values. For example, submitting a research paper to the ACM International Computing Education Research conference is different from submitting to the International Conference on Learning Sciences: the former values work specifically about the learning and teaching of computing, while the latter values foundational theories of learning, largely agnostic to domain. Your question might be important to one community but not the other.

Communities also judge research questions against their impact on knowledge in a research community. For example, a research question about how to effectively leverage a rarely-used feature of a rarely-used learning technology that is on the brink of becoming obsolete may be judged as far less important than a multi-national, multi-institution study of a widely-used, promising new pedagogy.

Finally, there are stakeholders of questions beyond academia. Would teachers, students, parents, policymakers, employers, or other groups find the questions important? Some questions that are important to researchers may be viewed as quite irrelevant to those in practice, and vice versa. That does not mean that you shouldn't follow your interests, but that you should know if your question's importance lies in basic or applied research. Knowing *who* you're trying to impact can clarify how you report and disseminate your discoveries in that it helps you empathize with your audience's interests.

2.4 Is it novel?

A good research question is *novel* when we do not yet have a robust answer to it, although there may be parts of answers. To evaluate novelty, you need to understand the body of evidence that already exists to answer it. We believe this is best done before thinking about study methods, data collection, or other matters we will discuss later, since prior work is full of good ideas about how to tackle a question, but you can also start with an idea for a study first, and verify its novelty later. Either way, this process will be iterative, reshaping your questions and whatever study ideas you've generated.

In the general case, understanding a body of evidence means reading everything you can find that provides some evidence for the answer. Because of the way research papers are written and archived, this is often a messy process: authors use different words for the same concepts, they publish in multiple different journals, conferences, and books, and scholarly search engines are not particularly good at organizing this mess.

Perhaps the fastest approach to understanding what we know about a question is to find an expert who's already read the work relevant to the question. This might be a highly-cited scholar or a fresh PhD student who performed a literature review on the topic recently. If you don't know which expert to ask, approach scholars who might know, negotiating the social network of experts until you find someone who does know. Make sure your approach is specific and polite, and you'll often get a reply. If you don't, move on; don't become a persistent pest. If no one knows of work that answers your question, it's probably novel! More likely, they will know of works that are loosely related to your question, which also reveals something about the novelty of your question. (Note that discussing research opportunities with others poses inherent risks of being scooped, but it can also lead to fruitful collaborations and replication, and so should not be feared).

In the absence of an expert, another good strategy is to find a survey, review, or "systematic literature review" article (hint: this book is full of them). These papers synthesize a large body of work for the purpose of conveying what we know and do not know with confidence about some phenomena. If you're lucky, someone has written one that shows you all of the work related to your question. More often, it will describe *much* of the work, but you will still have to do some searching and synthesizing of your own to characterize what is known about your question.

If you can't find an expert or a review article, you may need to do a literature search of your own, searching digital libraries for every possible relevant prior work. Think of all of the terms that might be relevant, all of the journals and conferences that might be relevant, and query a digital library search engine for relevant papers. Once you find a few related works, find papers that those papers cite, and—especially—what papers cite that paper, traversing the citation graph for all of the relevant work you can find. You'll know you're done when you stop finding new research.

While you're searching, you'll want to read the relevant work in detail to understand and extract the evidence that exists, along with any additional terminology which will help you in your search. If after this reading, you still do not see an answer to your research question, or the evidence is sparse, uncertain, or partial, so that we need more evidence, then your research question is novel. On the other hand, if the evidence is robust, perhaps your time is better spent investigating something else. Sometimes researchers undertake replication studies to strengthen poorly supported answers to research questions. But if the evidence is strong enough that your question will only add a negligible amount of increased confidence or knowledge, it may be worth abandoning it in favor of something that will contribute more to our body of knowledge.

One final consideration of novelty is whether all possible outcomes of a study would be novel, or just some of them. For example, imagine you designed a controlled experiment to evaluate the impact of some new learning intervention: if the study shows that it causes increases in learning, that may be a novel contribution, but what happens if the study does *not* show that increase? What if it shows a decrease? What if nothing changes? You could run to study and take the risk; you could try and formulate secondary question that result in discovery no matter what the outcome; you could also reframe the question so that no matter the outcome, the study reveals something we do not know with high confidence. For example, gathering data about what caused (or failed to cause) a difference would allow you to instead investigate the mechanisms of the learning intervention in addition to its downstream effects. Of course, replications can also be novel if they bolster our certainty about something.

By the time you have considered all of these points, you should have a good grasp of the space that your question will inhabit. You should make sure your notes are collated, as an annotated bibliography or the like, which will form the basis of the "related work" section of an eventual report.

1.5 Prototyping research questions

As you iterate on research questions, it is critical to continually express your research question in a form that allows you to observe, critique, and refine it. So far, we've mostly just listed the question ideas. However, the *form* that a research question ultimately takes is an *argument*, beginning with the state of the world and our knowledge, and ending with the question. Outlining these arguments is the best form of prototype to integrate all of the evidence about novelty, soundness, and importance that you have iteratively developed.

Let's look at an example of one such argument and then dissect it. We'll extract the argument from the paper that won the "Chairs" award[1] at ICER 2016, titled "*Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals, and Self-Efficacy on Performance*", authored by Alex Lishinski et al. (2016). The argument in their introduction is as follows:

1. Measures of self-regulated learning (SRL) predict students' academic outcomes.
2. Previous research in computer science (CS) education has examined motivation and subcomponents of SRL as possible predictors of success in introductory programming courses.
3. Previous research has also suggested that there are gender differences with respect to self-efficacy that may affect course outcomes.
4. However, previous studies have focused on examining the relationships between individual SRL and self-efficacy constructs and course outcomes, rather than investigating how multiple components of SRL (e.g., goal orientation and metacognitive strategies) and self-efficacy interact over time as students learn to program.
5. RQ: What are the relationships between self-efficacy, goal orientation, metacognitive strategies, and course outcomes in introductory programming students?

These four claims and question together represent an argument that substantiates all non-personal qualities of research questions we have discussed thus far. For *novelty*, claim (4) clearly states a gap in prior work, crisply stating that constructs have been studied in isolation rather than together. For *importance*, claim (1) states that self-regulated learning is an important indicator of student outcomes, implying we should understand it more deeply. For *soundness*, the references cited in the paper for all claims (1-4) substantiate each statement and tie the concepts used to theory. The research question itself uses the same concepts in the cited theory, further establishing soundness.

Arguments like the above are a powerful genre for capturing and evaluating a research question's quality: its motivation, theoretical grounding, and relevant prior work. The benefit to this form is that it's *low-fidelity*: it's much easier to discard a few sentences than an entire draft. Moreover, it's faster to iterate, because there is less text. It's also easier to get feedback on from collaborators, because they can quickly evaluate the merits of your argument.

[1] This award is given by the program committee chairs as a signal of rigor and excellence.

3. Formulate: can you research it?

Congratulations, you have a novel, theoretically sound, important, interesting question! This is no small feat: evaluating each of those criteria is challenging and slow. Unfortunately, you still need a way to ensure that your question is researchable. Methods are how we generate evidence to answer questions, and so you must devise a feasible method.

As Figure 4.1 shows, formulating a method is a creative, iterative, and often social process. However, research questions often imply requirements about the methods that are appropriate to use in addressing them. If you are comparing samples, or populations, then you will need to be able to make quantitative, statistical claims—that will affect your study design. If you are investigating the sense that students make of their education 10 years after they graduate, then you will need to talk to them: that will affect your study design. Method choice is also often informed by the repertoire of methods that a researcher is comfortable with executing, that they've used before, and know will provide evidence that satisfies them. For instance, it is unusual for someone who habitually does controlled studies that provide quantitative answers to take on a phenomenographic enquiry (or vice-versa). This is not necessarily a negative limitation: the more you use a method, the more you understand the nuance of its application and so, in turn, are able to design better studies.

To illustrate, let's consider the research question: "*What barriers exist to higher education faculty adopting alternative pedagogical methods in large CS1 courses?*" Let's deconstruct the requirements implied by this question:

- We know we're looking for a set of "barriers."
- We know we're going to focus on CS1 courses in higher education with a large number of students.
- We know that we're focused on pedagogy, and not other aspects of instruction.
- Our unit of analysis is faculty and not departments or universities.

From these four facts about the question, many methods are ruled out. For example, there's no comparison, so we're not running a controlled experiment. The type of data we're seeking is inherently qualitative (we're not looking for *how many* barriers exist, but rather characterizations of them, and characterizations are qualitative). And because we're looking for a *set* of barriers, the question implies that we'd like a *complete* set of those barriers, not a biased partial set with unknown barriers unaccounted for. This suggests that we may want to do some kind of survey, interview, or observation of a diverse population of higher education faculty focused on the various moments in which faculty are devising pedagogy to teach a course. We may consider a diary study, or longitudinal sampling. Whichever type of method we choose, the question also implies that we need to somehow access the decisions of faculty, since the barriers that exist are affecting those adoption decisions.

Generating ideas for methods partly requires knowing what types of methods exist, but also requires creativity. Taking a good social science research methods course can provide a basic foundation for this. You can also read one of countless books about research methods (Babbie, 2013; Bordens & Abbott, 2002; Creswell & Clark, 2007; Baxter & Jack, 2008; McMillan & Schumacher, 2014; Coolican, 2017). But this foundational knowledge isn't enough. You need to ideate, brainstorm, and generate method ideas, and that requires having seen many examples of methods. Having a large set of examples in your head will help you devise a particular method for your study.

One way to capture your method ideas is to *sketch* the set of facts we would normally report in a "Methods" section of a research paper. To help us illustrate how to critique and refine a method, let's sketch a method for the research question above (not the only possible one, of course):

- **Population**. All higher CS education faculty in the world.
- **Sampling**. We will obtain the faculty mailing lists of all CS departments in the world.
- **Recruiting**. We will write a message to all mailing lists explaining our research goals, asking for volunteers to participate in a 15 minute phone call, scheduling calls for all who reply.
- **Procedure**. Before starting the interview, we will explain the study to the faculty and obtain their consent to participate. During the interview, we will first ask the faculty member about their current pedagogical practices and where they learned them. Then, we will ask about their awareness of alternative pedagogy and experiences they have in considering the adoption of them. If they have considered new pedagogy, we will ask about their decisions of whether to adopt those pedagogy.
- **Data collection**. We will audio record all interviews and then have them transcribed.
- **Data analysis**. We will analyze each transcript for barriers to adoption of new pedagogy techniques, creating a set of barriers across the sample.

Let us put aside for the moment the content of the method and focus on its role as a *sketch*. Note that it is concise, it has structure, and most importantly, it describes what you and your research team would do to answer the question. As a sketch, it is not complete, but it allows us to see what is missing, what is ambiguous, what is invalid, etc. By prototyping the idea, we can now critique it and improve it.

Just as research questions have many criteria for quality, study designs must be feasible, reproducible, internally valid, externally valid, construct-valid, and ethical. In the rest of this section, we discuss how to use a sketch of a method to evaluate these criteria.

3.1 Is it feasible?

Unfortunately, the world is full of good research questions that we simply cannot answer. We may not have access to all the CS departments in the world, we may not know how to tell if populations are equivalent, we may not be skilled in the analysis that the question requires. Assessing feasibility is therefore key.

There are many aspects of feasibility to consider:

- Is there a method that will give the right sort of data to allow you to address the question?
- Do you have the skills to carry it out?
- Do you have time it will take?
- Do you have the resources (money, people, materials, etc.) that it will take?

Let's consider the method sketch we just completed above. There's definitely a method we can use to answer the question: but feasibility issues lurk. For example, to get the faculty mailing lists of all of the CS departments in the world, we need to know who to contact to get those lists, and then we need rights to post to each list. We may not be able to get these for all departments. Worse yet, even if we could get access to the lists, faculty may be so disinterested in helping that we would get too few replies for us to get a reasonable sample. None of these is guaranteed to be an issue, but they are risks to assess and mitigate. For example, mitigation strategies might be to target a small representative sample of faculty, writing personal email requests to each rather than broadcasting to mailing lists. This might take more work upfront, but may lead to a higher response rate overall.

What skills does this method require? Because it's an interview, it requires interviewing skills. But the sampling and recruiting also requires coordination, scheduling, and organizational skills. The analysis will require substantial experience in building categories from qualitative data. The tradeoffs in this are numerous. Perhaps you have these skills already, in which case, risk is low. Or, perhaps you want to learn them, in which case risk of poorly executing the method is higher, but the reward may also be greater, since you will acquire new research skills. One way to mitigate this risk might be to find a collaborator with good planning and organizational skills and delegate planning to them. You might still learn the skills through their mentorship, while mitigating risk.

Is there time to execute the method? There are many CS faculty, so if the response rate is high, that could be hundreds of hours of interviews. Perhaps a doctoral student focused on data gathering for months would be able to do this. If the key individual contributor is an undergraduate working five hours a week during full-time classes, it's probably not feasible. Mitigating these risks might involve using a different data collection method, such as a survey, which scales data collection and eliminates interviewing time, but this would be at the expense of the richness and depth of the data.

And finally, do you have the resources to do the study? There has to be money to pay for researcher's time, to pay for the transcription of the audio, and there may need to be some financial incentives to get faculty to reply. If there is no money, you might subsidize the work with your time or choose methods that take less time and resources. For example, perhaps instead of paying faculty to reply, there is a way of motivating them through altruism (recruiting colleagues that you and your collaborators have personal relationships with). This will bias the sample in several ways, but it will likely require fewer resources and lead to a higher response rate. Bias of some form is rarely escapable, and so this might be a worthy tradeoff if future studies can overcome the bias with a different method.

Most research methods are hard in some way. When there are too many hard or high risk things, it may be worth considering a different method, or possibly a different question altogether. That said, conservatism around the feasibility of methods is also one of the most significant barriers to progress in research: methods are simply tools for building knowledge, and if we only ever use the same tools, we will only ever build the same kinds of knowledge. Taking risks and learning new methods that a research community has not yet used could result in new knowledge.

One example we know of this is from psephology, the study and analysis of voting and elections. Amongst those who conduct exit polls there is one team of researchers in the UK led by Professor John Curtice from the University of Strathclyde, who regularly produce the most accurate predictions (BBC, 2017). One of the reasons for this is their use of method. Like all such researchers, they carefully sample and carefully control their sample so it is representative of the nation. However this team does not *ask* people how they voted, which is a sensitive subject and which may invite voters to given an expected answer rather than a truthful one. Neither do they question or interview voters. Instead “Participants are provided with a mock ballot paper that mimics almost exactly the one they have just completed, and are invited to place it in a mock ballot box, thereby ensuring that they do not have to reveal to the interviewer how they have just voted” (Curtice et al, 2017). By using a new method, they get better get more accurate predictions than self-report.

3.2. Is it (sufficiently) construct-valid?

Even if a method is feasible, it may not be valid. There are several aspects to validity that you will need to consider (Messick, 1995). One of the first kinds of validity to consider—construct validity—is to ensure that the way you are collecting data, including the measures and instruments you are using, actually represents a genuine phenomenon and faithfully reflects how it exists in the world.

Take, for example, the notion of a “barrier” to pedagogy adoption. Are there really such things as barriers in faculty’s heads that faculty figuratively “bump into” and cannot overcome? Or is it a problem of risk, where faculty encounter some alternative way of teaching, but they see a high risk for failure? Or perhaps it’s not risk, but rather a complete oversight of the existence of other ways of teaching? Will any of the interview questions posed in this study successfully access whatever form of factors affect faculty adoption of new pedagogy? Without a strong alignment between the theorized existence of “barriers” and a carefully designed way of observing them, our study design is not valid.

In computing education research, nowhere is this more important than in measures of *engagement* and *learning*. Without strong theoretical grounding of what we believe engagement or learning to be, and strong measures of those phenomena grounded in those theories, we cannot make valid claims about either. For example, is engagement best conceived of as a measure of sustained attention over time, or as a proxy for intrinsic motivation to learn? Or, is learning to code best conceived of as learning a well-defined systematic process for solving problems or as an art, shaped and informed by a community of practice? We don't have these theories yet. Instead, we have partially validated measures with rudimentary notions of learning like the FCS1 (Tew & Guzdial 2011) or SCS1 (Parker et al. 2016). These allow us to make some progress, but to continue to make progress, we need to build, validate, and use instruments across a community and over time, while we also develop the theories that support their validation.

3.3 Is it (sufficiently) internally valid?

Another validity consideration for a study method is its internal validity, which concerns the capacity for the design to relate cause and effect with some certainty. While this validity issue is most relevant to randomized controlled trials (also known as controlled experiments), it is actually an issue anytime a study design attempts to infer causality.

To illustrate, let's return to our faculty interview study. We wanted to understand what prevents faculty from adopting novel pedagogy. Prevention is about causality, and so our study is inherently about studying cause and effect. But can our interview study actually observe cause and effect? Not really: the only way to do that would be to somehow access the decision-making process a faculty member goes through in their mind (assuming it's a conscious process) and observe the factor that is preventing adoption. Unfortunately, we don't have direct access to people's minds, so have to be assured that our instruments gather what they're thinking. Most research on decision-making shows that most naturalistic decision-making is first emotional and then post-rationalized (Lerher, 2010). Our interview study is much more likely to capture the rationalizations of non-adoption, rather than the emotional contexts that actually prevented adoption.

In general, research methods are carefully evolved instruments for ensuring internal validity. Therefore, the easiest way to increase the internal validity of your study is to follow the best practices of a method. Those best practices might be expressed in textbooks, exemplary research papers, or occasionally in papers describing how to execute a particular method.

Internal validity is also quite challenging in computing education research, as many of the phenomena we hope to observe (such as learning, interest, identity, attitudes, and engagement) are not overtly visible in the world, but hidden inside learners' and teachers' minds.

3.4. Is it (sufficiently) externally valid?

Most study designs narrowly focus on a tiny fraction of a much larger population of people, places, or things. *External validity* (also known as "generalizability") is the extent to which the conclusions we draw about these tiny samples might generalize to other people, places, or things. Methods with high internal validity tend to have low external validity, and vice versa: this is because gaining certainty about causality usually requires one to look closely and in a controlled setting, with the tradeoff that you cannot account for the other contexts not being observed. Similarly, the more diverse the contexts you observe, the less control you have. One can overcome some of these tradeoffs at scale; for example, a randomized controlled experiment with 15,000 participants testing a new drug can account for a lot of variation. The variation in social contexts learning and the variation in individual learning, are harder to control and measure. Moreover, there are fewer resources in education than in health sciences to run large scale studies. This makes this tradeoff hard to overcome.

The external validity of our hypothetical interview study depends very much on how many faculty participate in interviews and *who* participates in those interviews. If just 10 of 10,000 faculty reply, it seems unlikely that the set of barriers those ten teachers report would reflect the set of barriers present in the broader population. Worse yet, if the 10 that replied were all from the same CS department, the external validity would drop even further. We can try to control external validity in this case by increasing the chances that we get a large diverse sample. Or we can try to measure external validity by comparing the sample we *did* get with the overall sample, showing that they are similar to each other, for example, by showing that we recruited faculty from most colleges and universities, from most research areas, and from most academic trainings.

External validity in computing education research is particularly challenging because learning and education are so strongly affected by culture and learners' prior knowledge. Moreover, controlling for these is hard since we have few valid measures of either. Thus, once again, we aim for *sufficient* external validity, in the hopes that our methods of characterizing external validity improve and that others replicate our work.

3.5 Is it ethical?

Another criterion is ensuring your study design is ethical. For most academic research there are legal protections for human participants against a wide range of harms, certainly including physical harm, but also emotional and psychological harms including stress that can come from challenging tasks or distress that can come from deception.

In computing education research, there are many settings in which these harms can arise, and in unexpected ways. Take stress, for example. When the first author was an undergraduate, he conducted a study of the learnability of three different integrated development environments for statistical hypothesis testing. One environment was particularly notorious for its steep learning curve, and the tasks the participants were asked to do only exacerbated the challenges that curve imposed. While the study had been approved by the university's institutional review board (IRB), and the first author had already engaged fifty participants, the fifty-first had a particularly negative emotional experience with the tasks, as she had just failed a class in which she had to demonstrate similar competence. Having to relive that failure and be reminded of its effects on her delayed graduation was so traumatizing, she complained to the IRB and the study was halted for further review. Neither of these outcomes—the trauma to the student or the delays to the first author’s research—was desirable. In the best case, IRB ethics reviews can help to prevent these sort of things from happening through some upfront investment in careful ethical reflection on your study design.

Ethical considerations are often especially complex in MNMI studies, where approval for the study has to be obtained at every location data is collected. Sometimes it is enough that approval has been granted at the lead institution. However, just as often, multiple approvals have to be sought, and sometimes in quite complex ways. In one study, a 17-site consortium had four “lead institutions” as that was the only way approval could be obtained from four of the sites. When designing MNMI studies, you should certainly build in extra time for this step.

2.6 Pilot to analysis

A key strategy for successful study design is to find a way to pilot as many elements as possible—even if only with one person. This can greatly improve your procedures, data collection, and data quality.

Many people pilot data gathering (be it interview protocols or testing survey designs) but it is important not to stop the piloting effort too soon. Gathering data is not the last step: analysis is. A key part of prototyping a study design is describing and implementing the actual procedures you will follow to clean, process, and analyze the data you gather. What steps will you follow? What scripts do you need to write? What statistics will you compute? Failing to plan for analysis can have a range of consequences. For example, a year-long diary study conducted by the second author failed to adequately consider analysis: the quantity of data received was so large that even reading it in a sensible time allowance was forbidding. So the data stays, sitting in a database, unused and unanalysed. In contrast, the first author ran a survey study with two students, omitted the analysis planning, and only realized after gathering data from over 4,000 people that the survey didn’t ask a key question of every respondent.

Piloting to analysis is especially important in two situations, one when you are unfamiliar with the methods, the other in MNMI studies. A colleague tells the story of taking data to a statistician after it was collected only to be told that the design was flawed for quantitative analysis (for various reasons) and couldn’t support the claim they were trying to make. “Had you consulted me earlier” said the statistician with a sad smile “you could have got it right”. In MNMI studies, data is always gathered from multiple sites by multiple researchers, and often analysed collectively. It is imperative that you are quite certain the main pieces of analysis are achievable.

Preventing these problems is all about verifying that the elaborate plans you’ve made for data collection can actually answer the question you posed.

As with research questions, an excellent approach to capturing, evaluating, and refining study designs is to prototype them into an articulation of the method and analysis plan as you might publish in the “Methods” section of a research paper.

Here's why:

- Having detailed documentation on the study plan can help you and others execute the plan consistently and reliably.
- If you wait to write it later, you may forget crucial details about the design that need to be captured to support reproducibility.
- In writing the full detail of a design, you'll inevitably find more flaws, ambiguities, and details to determine before running the study.
- By detailing exactly how you plan to analyze the data, you'll verify that you will have the data you need to answer the question, and that the types of analyses you have planned are appropriate for the data you're collecting.

4 Articulate: is it coherent?

By the time you are ready to actually execute your study, there should be minimal uncertainty about what you are going to do, and what you expect to happen. This doesn’t guarantee that you will get the results you want—this is research, after all, and if we knew the result in advance, we wouldn’t need to do it. But you should know that your question is interesting, sound, important, and novel, and you should have a detailed description of method and piloted analysis that you know will answer this question. Judging a study’s overall *coherence* is therefore a key final step before proceeding.

You can think of coherence as a thread of logic running through the entire chain of reasoning, from the first claim of the argument, to the research question, all the way through the data collection procedures and data analysis plan. Throughout this entire line of logical reasoning, every claim made should be substantiated, every question sound, every step of a method valid, every statistic consistent with its assumptions, and every interpretation congruous with the evidence gathered. This is ultimately a holistic judgement, and the same judgement made by other researchers during peer review, so doing this *before* you execute a study is a critical way to further eliminate flaws.

Let’s consider one example study and discuss it coherence. We will focus on the ICER 2017 paper by Danielsiek et al. (2017), who sought to build a validated instrument for measuring self-efficacy in introductory programming courses. Below is a set of claims that run through their paper; try judging the coherence of the claims:

- Introduction
    - Self-efficacy is the personal beliefs of one’s ability to succeed in a situation or task.
    - How a teacher teaches can influence self-efficacy.
    - Knowing the impact of teaching on self-efficacy requires measurement.
    - There is no measure of self-efficacy in introductory algorithms courses.
    - What is a conceptually valid measure of self-efficacy in algorithms courses?

- Method
    - We partnered with four institutions to measure self-efficacy.
    - We adapted a previously designed instrument to an algorithms course context.
    - We administered the adapted instrument in four classes, obtaining 130 responses.
- Results
    - We verified the data had sufficient sphericity to be factorizable.
    - We performed a factor analysis, finding four factors that explained 66% of the variance.
    - The factors were consistent with the instrument's intended measurements, indicating construct validity.
    - Cronbach's Alpha was 0.938, suggesting reliability.
    - The instrument did not correlate with measures of self-regulation, suggesting divergent validity.
    - The instrument did correlate with measures of personality traits, suggesting nomological validity.
- Conclusion
    - Therefore, the instrument we designed is a conceptually-valid measure of self-efficacy in introductory algorithms courses.

Note how each of the claims in this overarching argument are *all* in support of the final conclusion statement, achieving coherence. Viewing a study design from this perspective allows you to also assess the final conclusion based on the claims that support it. For example, one could quibble about whether the amount of evidence to support divergent validity is enough to claim that the measure is valid. Or, one could point out (as the authors do) that because this was the first study to instrument self-efficacy in this context there is no evidence to support convergent validity. This threatens the final claim that the instrument is valid, but cannot be tested until more studies are conducted in the same area.

In the first author's lab, students are expected to write arguments like the above to support such assessments. The best form of this task is to actually write the skeleton of the research paper reporting the results *before* running a study. That means writing an introduction with a series of claims leading to a question, all of the related work necessary to substantiate its soundness, importance, and novelty; a method description of information needed to execute the study, and an analysis plan that has been verified for feasibility and its ability to answer the question. Also writing a "limitations" or "threats to validity" section *before* executing the study will help brainstorm all of the flaws in the study that you might be able to eliminate before gathering data. With a skeleton of the paper, assessing coherence is then a matter of evaluating it as it will be assessed peer review, but without actual results or a discussion implications. This is not only great practice for reviewing, but also the best way to verify that your research plan is a quality one.

5. Execute

Once you're ready to execute a study, data collection and analysis should be straightforward, at least to the extent that you have sufficiently planned it. Of course, rarely (actually, never) does everything we have described in this chapter go perfectly. Time and resource limitations can lead to satisficing; lack of skills can lead to a wide range of mistakes and errors, some of which can be catastrophic to a study. The key is to learn the foundations in this book and and get as much feedback as possible before you gather data.

Treating study design as a design process furnishes you with a number of artefacts along the way:

- A literature review of the prior literature about your question
- A series of claims that establish the importance, soundness, and novelty of a research question, refined through feedback and further literature review
- A sketch of your method and analysis plan, refined through piloting into a paper outline

With these in hand *before* you execute your study plan, if execution and analysis goes well, all that is left is writing the results, adding a discussion of implications, and then sharing your discoveries with the community.

In this chapter we have presented a series of heuristics for considering study design as a design process. Like all heuristics, they offer "A process that may solve a given problem, but offers no guarantees of doing so" (Newell et al, 1957). Using them should give a better result than not using them, but achieving power and elegance in design, in this as other fields, rests with the skill of the practitioner.

References

Babbie, E. R. (2013). *The basics of Social Research*, 13th edn, Boston, MA: Cengage Learning.

Baxter, P. & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The Qualitative Report,* **13**(4), 544-559.

BBC (2017). Election 2017: methodology. http://www.bbc.co.uk/news/election-2017-40104373

Bordens, K. S. & Abbott, B. B. (2002). *Research Design and Methods: A Process Approach,* New York, NY: McGraw-Hill.

Coolican, H. (2017). *Research Methods and Statistics in Psychology,* Hove, UK: Psychology Press.

Creswell, J. W. & Clark, V. L. P. (2007). *Designing and Conducting Mixed Methods Research,* Thousand Oaks, CA: Sage Publications.

Curtice, J., Fisher, S., Kuha, J. & Mellon, J. (2017). On the 2017 exit poll - another surprise, another success. *Discover Society, Focus, Issue 46.* https://discoversociety.org/2017/07/05/focus-on-the-2017-exit-poll-another-surprise-another-success/

Danielsiek, H., Toma, L. & Vahrenhold, J. (2017). An instrument to assess self-efficacy in introductory algorithms courses. In *ACM International Computing Education Research Conference* (pp. 217-225). New York, NY: ACM.

Fincher, S., Lister, R., Clear, T., Robins, A., Tenenberg, J. & Petre, M. (2005). Multi-institutional, multi-national studies in CSEd research: Some design considerations and trade-offs. In *Proceedings of the First International Conference on Computing Education Research* (pp. 111-121). New York, NY: ACM.

Fincher, S., Tenenberg, J. & Robins, A. (2011). Research design: necessary bricolage. In *ACM International Computing Education Research Conference* (pp. 27-32). New York, NY: ACM.

Jadud, M. C. (2006). *An exploration of Novice Compilation Behaviour in BlueJ.* Doctoral dissertation, University of Kent, UK.

Lawson, B. (2006). *How Designers Think: The Design Process Demystified,* New York, NY: Routledge.

Lee, M.J., Bahmani, F., Kwan, I., LaFerte, J., Charters, P., Horvath, A., Luor, F., Cao, J., Law, C., Beswetherick, M., Long, S., Burnett, M.M. & Ko, A.J. (2014). Principles of a debugging-first puzzle game for computing education. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (pp. 57-64). New York, NY: IEEE.

Lehrer, J. (2010). *How We Decide,* Boston, MA: Houghton Mifflin Harcourt.

Lishinski, A., Yadav, A., Good, J. & Enbody, R. J. (2016). Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals, and Self-Efficacy on Performance. In *ACM International Computing Education Research Conference* (pp. 211-220). New York, NY: ACM.

Loksa, D., Ko, A. J., Jernigan, W., Oleson, A., Mendez, C. J. & Burnett, M. M. (2016). Programming, Problem Solving, and Self-Awareness: Effects of Explicit Guidance. In *ACM Conference on Human Factors in Computing Systems* (pp. 1449-1461). New York, NY: ACM.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B., Laxer, C., Thomas, L., Utting, I. &Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In *ITiCSE Working Group Reports* (pp. 125-180). New York, NY: ACM.

McMillan, J. H. & Schumacher, S. (2014). *Research in Education: Evidence-based Inquiry, 7th edn.,* London, UK: Pearson Higher Ed.

Messick, S. (1995). Validity of psychological assessment: Validation of inferences from persons' responses and performances as scientific inquiry into score meaning. *American Psychologist,* **50**, 741-749.

Nelson, G. L., Xie, B. & Ko, A. J. (2017). Comprehension First: Evaluating a Novel Pedagogy and Tutoring System for Program Tracing in CS1. In *ACM International Computing Education Research Conference* (pp. 2-11). New York, NY: ACM.

Newell. A., Shaw, J. C. & Simon, H. A. (1957). Empirical explorations with the logic theory machine. In *Proceedings of the Western Joint Computer Conference, Vol. 15* (pp. 218-239). New York, NY: ACM. [Reprinted in Feigenbaum and Feldman (1963), pp. 109-133.]

Parker, M. C., Guzdial, M. & Engleman, S. (2016). Replication, validation, and use of a language independent cs1 knowledge assessment. In *ACM International Computing Education Research Conference* (pp. 93-101). New York, NY: ACM.

Sweller, J., Ayres, P. & Kalyuga, S. (2011). *Cognitive Load Theory (Vol. 1),* Berlin, Germany: Springer.

Tew, A. E. & Guzdial, M. (2011). The FCS1: a language independent assessment of CS1 knowledge. In *ACM Technical Symposium on Computer Science Education* (pp. 111-116). New York, NY: ACM.

Utting, I., Tew, A. E., McCracken, M., Thomas, L., Bouvier, D., Frye, R., Paterson, J., Caspersen, M. Kolikant, Y.B-D., Sorva, J. & Wilusz, T. (2013). A fresh look at novice programmers' performance and their teachers' expectations. In *ITiCSE Working Group Reports* (pp. 15-32). New York, NY: ACM.