The Logic of Effective Iteration in Design-Based Research

Daniel G. Rees Lewis, Spencer E. Carlson, Christopher K. Riesbeck, Kristine J. Lu, Elizabeth M. Gerber, and Matthew W. Easterday
daniel-lewis@northwestern.edu, S.C.@u.northwestern.edu, c-riesbeck@northwestern.edu, k.lu@u.northwestern.edu, egerber@northwestern.edu, easterday@northwestern.edu
Northwestern University

Abstract: Educational Design-Based Research (DBR) methodologists stress that DBR teams should iterate in order to create desirable designs and impactful theory. Effective iteration helps DBR teams understand stakeholder goals and quickly reject both designs stakeholders do not want and less impactful theory. However, we in the learning sciences have not defined the logic of iteration in DBR, specifically: (1) What are the core elements of iteration? (2) When should DBR teams iterate? (3) If DBR teams aim to iterate rapidly, how can they organize DBR to do so? (4) How can teams choose their focus when they iterate? We propose four key principles that make up the logic of iteration in DBR: (1) Building-testing cycles, (2) iterating early and frequently, (3) slicing, and (4) risking. This paper helps us organize DBR projects and train learning scientists to better support stakeholder goals, develop more impactful theory, and positively impact policy and practice.

Keywords: Design-based research, Methods

Iteration process and challenges in education design research

The goals of DBR are to support stakeholder goals and produce impactful theory (Collective, 2003; Collins, Joseph, & Bielaczyc, 2004; Easterday, Rees Lewis, & Gerber, 2017; Sannino, Engeström, & Lemos, 2016). DBR methodologists stress that DBR teams should "in parallel, iterate toward better designs and add to foundational understanding of learning processes and how to support learning" (DiSalvo, Yip, Bonsignore, & Carl, 2017). Design-based research (DBR) teams might have projects that support student learning and flourishing, support educator practice, contribute to the local community, and produce novel and impactful theories for the research community. However, it is all too easy for DBR teams to dedicate a tremendous amount of time and effort to create designs that stakeholders do not want (Carlson et al., 2018.; Penuel & Gallagher, 2017), or theory that the research community does not want (Dede, 2004); effective iteration can reduce the wasted work by helping DBR teams to more quickly reject designs and theories that will not support stakeholder goals or advance theory.

Iteration is a deliberate set of activities to (a) learn more about stakeholder goals, (b) learn more about the extent to which the design is effective in meeting stakeholder goals and researcher goals, and then (c) change the design and theory in accordance with what is learned (Adams, Turns, & Atman, 2003; Easterday et al., 2017; Rees Lewis et al., 2018; Schön, 1983). A DBR team might iterate by showing a draft curriculum to parents and students, learn that it doesn't resonate with them, and change the design accordingly. Theory also develops through iteration. One type of theory that DBR can produce are novel design arguments or local instructional theories—these are theories about a design to bring about given learning in a given context (Easterday, Rees Lewis, & Gerber, 2016; Gravemeijer & Cobb, 2013). Through iteration, when the DBR teams find that their design, guided by theory, doesn't function as intended, they change the design and theory accordingly. Another type of theory that DBR can produce are ontological innovations—novel categories that change the way the DBR team understand the project (DiSessa & Cobb, 2004), such as a new type of knowledge or social processes relevant to meeting stakeholder and researcher goals. Through iteration, DBR teams find ontological innovations when the design functions and the stakeholders act in a way that is very different than they expected (e.g. Yackel & Cobb, 1996).

Iteration is important because DBR teams face a highly ill-structured problem. At the start of the project many aspects of the project unclear, such as (a) what the stakeholder goals are, (b) what the designs might work, (c) what learning processes the team should focus on, (d) what knowledge the team needs, and (e) what designs and theory the team should create (Jonassen & Hung, 2015). Through iteration DBR teams can gain greater certainty about these unclear aspects of the project (Adams et al., 2003; Easterday et al., 2017; Gravemeijer & Cobb, 2013).

Despite broad agreement that iteration is critical to successful DBR projects (Collective, 2003; Collins et al., 2004; Easterday et al., 2017; Gravemeijer & Cobb, 2013; Sannino et al., 2016), we have not yet defined the logic of iteration in DBR—by which we mean the set of principles that guide the activity of iteration in DBR

teams. Specifically, in the learning sciences, we have not yet articulated: (1) What are the core elements to iteration? (2) When should DBR teams iterate? (3) If we aim to iterate early and frequently, how can we organize DBR to do so? and (4) How can teams choose their focus when they iterate? By defining the logic of iteration, we can help DBR teams to iterate effectively towards designs that better support stakeholder goals, produce better theory, with less wasted work (Dede, 2004).

Without a principled approach to iteration, DBR teams can easily create designs that stakeholders do not want, create unimpactful (or no) theory, and spend months or years of wasted work. DBR teams might iterate too late, finding that the stakeholders do not want the design, or the theory doesn't work after spending months or years of work on the project. DBR teams may iterate with no clear goals in mind and have no clear plan to collect data that informs the team about the stakeholder goals and the design. Perhaps worse, DBR teams might avoid interacting with stakeholders at all until late in the project. This can lead to the team failing to understand messages from the stakeholders about their goals (Penuel & Gallagher, 2017), or to an "elephantine effort result[ing] in the birth of mouse-like insights in their contribution to educational knowledge" (Dede et al., 2004, pp. 107). Herrington and colleagues (2007) note that given the danger that DBR projects might produce no useful designs and theories, some researchers advise PhD students to avoid conducting DBR during their PhD.

Imploring DBR researchers to iterate does not in itself help them to iterate more effectively. Some articulations of the DBR design process in the learning sciences helps us understand the elements that go into iteration (e.g. Easterday et al., 2017; McKenney & Reeves, 2018). However, stating that iteration involves moving between activities such as understanding stakeholders, defining project goals, creating designs, and testing designs does not support DBR teams to reason about how to iterate. This is also the case for processes that stipulate timelines with phases that students should move between.

We illustrate the challenges DBR teams have with iteration with an example DBR educational technology project called *Assist*. In project *Assist* the team worked to support a K-12 teacher professional development training network. The network conducted professional development with hundreds of teachers across over 20 school districts by training teachers to adapt their lesson plans and teaching practices to be more equitable and effective (Debarger et al., 2017; Lewis, Perry, & Murata, 2006). The stakeholders in the network included the steering committee, professional development facilitators who worked with teachers, district county admin, university faculty, teachers, students, and parents. Through negotiation with stakeholders in the network, the DBR team scoped a broad line of inquiry (McKenney & Reeves, 2018): how to create technology and related practices to increase the quantity and quality of feedback that K-12 teachers receive from university researchers when adapting curriculum in a geographically distributed network. Supporting more regular feedback to teachers emerged as the goal because (a) the geographically distributed nature of the network meant that university faculty who volunteered to give feedback didn't know when to give feedback, and (b) the teachers reported they wanted more feedback as part of a challenging curriculum adaption process. Based on this situation, what should the DBR team do? Stating that the team should iterate, or understand stakeholders, or define goals, is not enough to help them know an effective course of action to take. We will return to this example to illustrate the logic of iteration throughout the paper.

Our claim is that while some experienced DBR teams conduct effective iteration, we in the DBR learning sciences community have not fully articulated the logic of effective iteration. A lack of explicit logic to iteration in DBR hampers our ability to teach DBR to students, and more broadly to conduct more effective DBR in the learning sciences that impacts practice, theory, and policy. Consequently, we ask: *what is the logic of iteration in educational design-based research (DBR) that can increase the chances we better support stakeholder goals, produce more impactful designs and theory, and reduce wasted work?*

The logic of effective iteration

To define the logic of iteration, we must define the core elements of iteration, as well as addressing the challenges that arise when we seek to iterate more rapidly during DBR. Consequently, we ask four sub-questions to outline the logic of effective iteration: (1) What are the core elements to iteration? (2) When should DBR teams iterate? (3) If DBR teams aim to iterate rapidly, how can they organize DBR to do so? (4) How can teams choose their focus when they iterate? In the first two questions we build upon more established elements of iteration in the learning sciences, namely that we should look to build and test early (e.g. (McKenney & Reeves, 2018). In the final two questions we address how DBR teams can achieve the goal of building and testing early to support stakeholder goals and build impactful theories; in these final two questions we contribute the ideas of slicing and risking.

Building and testing cycles

The first question to ask is: *What are the core elements to iteration?* Building and testing is the core of iteration. In this section we further define this principle. Building and testing helps DBR teams in the face of project uncertainty. The goal of iteration is to reduce that uncertainty so the DBR team can make better decisions to meet stakeholder goals and build impactful theory. DBR teams must avoid two traps (a) only building, and so operating blind without any reduction in uncertainty, and (b) only analyzing without creating their design.

Table 1: The key principles that make up the logic of effective iteration in DBR are: (1) Building-testing cycles, (2) iterating early and frequently, (3) slicing, and (4) risking

Question	DBR Team Challenge	Key Principles
1) What are the core elements iteration?	It can be challenging to elicit goals and constraints through abstract conversations with stakeholders. Furthermore, teams might not make continual progress in creating designs and building theory.	**Building-testing cycles**: Ongoing attempts to create designs that stakeholders recognize as useful and that are in fact useful when implemented. Testing these designs with stakeholders to inform what to build in the future.
2) When should DBR teams iterate?	DBR projects can easily fail to support stakeholder goals (e.g., teachers, community members, research community) and not discover key goals until too late.	**Iterating early and frequently**: DBR teams should seek to iterate early and often by conducting frequent build-test cycles. The speed of the build/test cycles can nearly always be profitably increased, especially early in the project.
3) How can DBR teams iterate early and frequently?	The DBR team cannot build and test the whole design right away. Therefore, the team faces the challenge of how they iterate to test the most important aspects of the design.	**Slicing**: Choosing a small enough version of the design to build and test, which could feasibly support the goals and practices of stakeholders within each cycle.
4) What can teams choose their focus when they iterate?	There is an often-bewildering plethora of things that DBR teams could iterate on. For example, DBR teams might ignore critical stakeholders until too late into the project, causing the project to fail because it does not support stakeholder goals.	**Risking**: Defining which aspect of the project is most likely to make the project fail to make the desired impact because it fails to support stakeholder goals. The team seek to reduce risks during the build and test cycle.

To avoid these traps, DBR teams should continually conduct building and testing cycles (Gravemeijer & Cobb, 2013; McKenney & Reeves, 2018). Building and testing is the cornerstone of effective iteration (and effective design practice more broadly) and involves both gaining understanding of stakeholder goals, the context, and understanding more about what design the team should build (Adams et al., 2003; Schön, 1983). Building and testing cycles can incorporate all aspects of a design process (Easterday et al., 2017; McKenney & Reeves, 2018), but critically involve creating and testing a some aspect of the design. Building and testing involves creating and empirically testing a design that instantiates a theory about how to support the desired learning processes in a given context—what we might call a design argument, a local instructional theory, or conjecture (Easterday et al., 2016; Gravemeijer & Cobb, 2013; Sandoval, 2014). In testing, the DBR team define questions that if answered help them gain more certainty about stakeholder goals, the context, and the extent the theory instantiated in the prototype is meeting the stakeholder and researcher goals (McKenney & Reeves, 2018). Through empirical work the DBR team might also learn something unexpected that changes the project (DiSessa & Cobb, 2004).

Building and testing helps DBR teams to communicate with stakeholders, and continually work towards supporting stakeholder goals. Through building and testing, DBR teams are able to avoid communicating in abstract terms; it is all too easy for DBR team and the stakeholders to appear to agree, but in fact imagine significantly different designs (Ehn, 1988). DBR teams can build and test through creating parts of the design at different levels of fidelity, including prototypes and mockups that are tangible representations of an idea that are quicker to build and test (Ehn, 1988). While DBR teams might not consider much of what they have created in a cycle as "finished", teams should aim to create prototypes that are intended to be valuable for the stakeholder at

the point of testing—both in perception and actuality. Penuel and Gallagher (2017) argue that in research-practice partnerships researchers should have questions that stakeholders recognize as supporting their goals—which is achieved through both better communication but also through asking the right questions. In the same spirit, here we argue that in each build and test cycle, stakeholders should perceive what is built and tested actually supports their goals. In continually seeking to build and test tangibly valuable designs for the stakeholders, the DBR team and stakeholders can better see the extent to which the design is supporting stakeholder goals (Ehn, 1988).

Project assist: Building and testing cycle

So how does this help the project *Assist* DBR team to iterate effectively? As we will now illustrate, the principle of building and testing only supports project Assist to some extent. The team created technology and related practices to provide teachers online feedback as they adapt their curriculum to make the more effective. A build and test cycle would involve creating part of the design that can be immediately useful to at least one stakeholder type. The team learned from conversations with teachers and that they wanted more support for the curriculum adaption process they were undertaking; this was one of the reasons that the team chose to pursue an online feedback system so that university faculty might support teachers through online feedback. The team wanted to better understand: (a) The extent teachers valued online feedback of their lesson plans, and (b) if teachers would use the feedback to improve their lessons. To answer these questions, the team sought to build a feedback interface with features drawn from the online feedback literature. The team tested the interface with a few teachers to see (a) if teachers found the feedback and interface desirable, and (b) if the teachers used the written feedback to improve their lesson plans. Here, the team are trying to create something of immediate value to the teachers, because providing feedback to the teachers could have been useful to them. In doing so, the team are tested part of their design argument—that the interface they designed—backed by principles from the online feedback literature—was desirable and effective for teachers in the given context. The DBR team found that the teachers they tested with unanimously report that receiving online feedback on their lesson plans was highly undesirable. Teacher found the online feedback to be impersonal and untrustworthy, and reported they would never use such a system or apply the feedback to their lessons. Through showing the teachers the feedback interface, the team were able to communicate with the teachers in more concrete terms; in previous discussions with the team, five teachers had reported that they wanted to receive feedback on the lesson plans. However, through working together the teachers and team were able to more precisely define what types and modes of feedback teachers (didn't) want.

The team learned that, as conceived, their design did not support stakeholder goals. The team benefited greatly from building and testing; in that they were able to rule out a design approach they were planning to take. Furthermore, the team learned that the design would need to support trust-building with teachers. However, the principle of building and testing cycles leaves many questions about effective iteration unanswered: the principle does not help the team know when, how, or what to build and test.

Iterating early and frequently

We now turn to the question of: *When should DBR teams iterate?* We argue the answer to this question is to iterate early and frequently in the project. DBR teams should seek to iterate early and frequently because stakeholder goals might not be surfaced early in the project; it is all too easy for teams find out late in the project their design doesn't support stakeholder goal or the desired learning. Iterating early and frequently allows the team to gain greater certainty about these initially unclear aspects of the project, and to more quickly understand what will and won't work (Jonassen & Hung, 2015). If teams discover that their design does *not* support stakeholder goals, they have reduced wasted work compared to if they had not tested and pursued an ineffective design. Furthermore, the team are better informed to design something else that will support stakeholder and researcher goals. Over years of mentoring less experienced DBR teams we have found that there is sometimes resistance to iterating early and frequently because it takes time to test, and teams don't feel their designs are "ready". We have found that there is nearly always a way to iterate earlier and more frequently that helps the DBR teams to develop better designs and theory. We argue DBR teams can find ways to increase the frequency of their building and testing cycles through *slicing*, which we discuss next.

Project assist: Iterating early and frequently

We previously described how the *Assist* DBR team tested their online feedback interface, created to support teachers adapt their curriculum to be more effective. The team saw that the teachers found the online feedback in the interface impersonal and untrustworthy, and so they didn't find the system desirable and didn't use the feedback to improve their lessons. No matter when the teams conducted this build and test cycle, it would have been tremendously useful; the build and test cycle allowed the team can move away from a design that a key

stakeholder does not want, and a theory about an approach that probably can't be impactful in the education system. However, it is clearly better that the team conducted this build and test cycle earlier, as it both helps them understand the goals of the teachers better, and they will avoid wasted work. We also argue that these tests can be done frequently. Conducting the build and test cycle we describe above with just a handful of teachers over a couple of 1-2 weeks is feasible for many DBR teams. The first two principles conducting (1) build and test cycles (2) early and frequently in turn begs the question of how to build and test in this way, and what to build and test.

Slicing

Teams can rarely create and test a complex design in one build test cycle, which leads to the challenge of how to iterate early and frequently. So, we now turn to the question of: *If DBR teams aim to iterate rapidly, how can they organize DBR to do so?* To meet this challenge, DBR teams must *slice*, a term we borrow from software design (Rasmusson, 2010). Slicing involves choosing a small enough version of the design argument to build and test, which could feasibly start to support the goals of stakeholders within each cycle. DBR teams cannot simply do more work once committed to iterating frequently, but instead should reduce the scope of their work within a given cycle. Because it is easy for DBR teams to waste time and effort creating designs and theory that do not support stakeholder goals, the goal of slicing is to create a prototype that can help the team to understand the extent they are supporting stakeholder goals; that is, in slicing, the DBR team should aim to create a design that stakeholders can use to support their goals. By continually creating designs, the DBR team can make steady progress and avoid rushing out a design that does not support stakeholder goals or produce impactful theory. A DBR team might test a critical lesson in a curriculum to see if it encourages certain practices, or demonstrate the use of technology to students, teachers and admin to see if it is desirable.

Slicing is *incremental* and *panoramic*. Slicing is *incremental,* in that the DBR teams should first build and test more valuable parts of the design in higher fidelity. Slicing incrementally allows DBR teams to test in higher fidelity, and so the team can conduct more accurate tests. Slicing is also *panoramic,* meaning that while some parts of the design are built in higher fidelity the design team still envisage all aspects of the design. DBR teams should also work panoramically by thinking through and creating, even at low fidelity, all the aspects of the design they envisage. By slicing panoramically, the DBR team can better reason about what should be included in the slice and what will be included elsewhere in the design. A DBR team might slice incrementally by creating and testing one of the most important lessons of a given curriculum and panoramically by sketching out rough content across the whole curriculum.

Part of slicing is building and testing in a way that allows the DBR team to gather useful measures. In slicing, DBR teams should gather (1) measures that stakeholders recognize as relevant to their goals, and (2) measures that are as far down the causal chain towards researcher and stakeholder goals as the team can manage within the cycle. In slicing, DBR teams must make decisions about what to build, which in turn impacts what data they can collect. First, DBR teams should gather measures that are as close to supporting stakeholder goals as possible; the DBR team should aim to show the measures to relevant stakeholder goals. This might involve working closely with stakeholders and including stakeholders in the DBR team to understand what they view as evidence that their goals are being supported (DiSalvo et al., 2017; Gutiérrez & Jurow, 2016; Sannino et al., 2016). Second, DBR teams should gather measures that are as far down the causal chain of supporting stakeholder goals as possible. That is, the measures should be as close to the final outcome that the stakeholders care about. It is easy for DBR teams to measure a lot of activity, but not what is important to their theory of why the design can support the goals of the stakeholders and research team. It is worth noting that the measures stakeholders recognize as relevant to their goals and measures that are further down the causal chain may or may not be the same, so DBR teams should gather both. In sum, slicing helps DBR teams organize cycles in projects to focus on regularly building, testing, and measuring outcomes to support stakeholder goals.

Project assist: Slicing

We previously described how the *Assist* DBR team tested their online feedback interface that they created to support teachers adapt their curriculum to be more effective. So how might the DBR team create and test something within that build and test cycle that stakeholders could perceive as valuable and stakeholders can use to support their practice? The following slice took a team of two researchers 2-weeks.

The team asked five teachers for their lesson plans that they were adapting. They then asked one of the university faculty with expertise in math teaching and equity to provide feedback on the lesson plans. The team then created the feedback interface with the graphical elements that they intended to use based on the online feedback literature and included the written feedback from the university faculty. The interface was a prototype without all of the intended functionality that they were able to quickly create—for example the university faculty couldn't actually enter the feedback into the system. The team then tested the interface with the five teachers by

showing it to them and asking the teachers to (a) make any changes to their lesson plan they saw fit based on the feedback, and then (b) asking the teachers about their experience with the interface, including how desirable it was. In this way, the team push towards a higher fidelity, as they created an interface largely as they intended and included actual feedback from university faculty on the teacher's actual lesson plan. Here the slice is incremental, as the team created one part of the design in higher fidelity. The slice is also panoramic, in that the team still sketched out other parts of the system, such as how the university faculty would give feedback. We find that slicing tends to help the team start to answer many other questions. For example, in engaging the university faculty to give feedback, the team got a sense of the types of feedback they might give and were able to change their criteria for what counts as a quality lesson.

In this example, the team gathered measures that were relevant to stakeholder goals. The measures were relevant to teacher goals, as they sought to measure how desirable and useful the system was in supporting their curriculum adaption process. The measures were also relevant to the partner organization goals, in that they sought to capture the extent the design and feedback improved the curriculum (not at all in this case). The measures were also relevant to the DBR team's goals, as the team sought to create theory for how to provide effective online feedback to help teachers improve curriculum.

The team also sought to gather measures that are as far down the causal chain towards researcher and stakeholder goals as the team possible within the cycle. When implementing technology to elicit feedback, the DBR team could seek to measure many things, including but not limited to: (a) if teachers like the idea of getting feedback, (b) if teachers like written online feedback from an expert, (c) if feedback was elicited in an online system, (d) the quality of the feedback, (e) if teachers use the feedback to revise the lesson plans, (f) the quality of the revisions of the lesson plans, (g) the teaching practices enacted in the lesson, (h) the students reactions to the lessons and which students reacted in what ways, (i) changes in student learning and which students learning is impacted, and (j) the extent that the achievement gap was reduced on the state test. Within one build and test cycle, the DBR team couldn't gather measures at the end of this causal chain—that is, a more equitable outcomes on the state math test. However, they were able to measure (e)—if the teachers used the feedback in the feedback interface to change in the lesson plans (not at all in this case). If the teachers had changed the lesson plans, they would also have been able to measure (f)— the quality of the revisions they made. We find that by slicing, DBR teams can nearly always create designs and gather measures that are further down the causal chain towards what both the stakeholders and DBR team care about. It is nearly always possible for a team to move beyond more superficial measures that some inexperienced design teams are drawn to. However, while slicing helps the teams understand how to both iterate early and frequently, it does not tell the team what they should focus on slicing.

Risking

Given that slicing supports early and frequent iteration, DBR teams must constantly decide the focus of the build and test cycles. So, we ask: *How can teams choose their focus when they iterate?* DBR teams must choose what to focus on in their slice by *risking*: choosing their slice to address the project's biggest risks. While in this paper risking is the last principle we discuss, risking is in fact the first consideration in effective iteration. A risk is anything that presents a threat that the design could fail to support stakeholders' goals or create useful theory due to the team's incomplete or incorrect understanding of critical aspects of the project (Carlson et al., 2018). Each cycle, the DBR team should identify the biggest risk(s) and seek to mitigate those risks through building the part of the design that enables them to test whether their intended design overcomes the risk. Risks might come from many aspects of the project, for example the DBR team might not understand stakeholder goals, the intended design might address a goal the stakeholder does not have, or the design might not achieve the learning it is intended to support. Risking also means that the DBR teams will not build and test the first part of the design as experienced by stakeholders; rather, the DBR team should create and test the parts of the design associated with the biggest risk. If the team have no evidence that they have parents buy-in, and that parents play an important role in the design, the team might need to build and test the part of the design that relates to parents.

All DBR projects have multiple risks and must prioritize those risks, which is a value laden activity (Gutiérrez & Jurow, 2016). We do not make pronouncements in this paper over exactly how to prioritize risks. In our DBR teams we follow a priority as described in Carlson et al. (2018) and Penuel and Gallagher (2017) who prioritizes gaining stakeholder buy-in, understanding stakeholder goals and then meeting stakeholder goals above all other risks in DBR.

Project assist: Risking

At the start of the *Assist* project, there were a number of risks that the team had to choose between to drive effective iteration. In the build and test cycle we described above, the team learned teachers didn't trust the feedback in the online interface; the risks the DBR were testing was that the design wouldn't meet the teachers' goals in that it

would not be a desirable and effective way to help teachers improve their curriculum. The team faced many other risks, including (a) the partner organization, a teacher professional development network, would no longer perceive the design and research as valuable, (b) even with effective feedback, the teachers might not adapt the curriculum to students' needs in a way that better supports student learning, (c) the DBR team's research questions might not be novel, and (d) the university faculty designated to provide feedback might not provide any high quality feedback.

In the *Assist* project, the DBR team focused on building and testing to address the risk of teacher desirability and use of the feedback. They chose this risk because (a) if this did not work than the team's whole design approach would not support any stakeholder goals, and (b) the team had no evidence that online feedback would help teachers to improve their curriculum, beyond initial positive remarks from teachers about wanting feedback. Other risks were part of this prioritization. For example, another risk the team discussed was if parents would find the process desirable. The parents were involved in the adaptation process as they observed public lessons that teachers had adapted (Lewis et al., 2006). The team chose not to focus on this risk for this build and test cycle because (a) parents had already expressed positive attitudes about the adaption process and the public lesson during short interviews the team did when asking for IRB consent, and (b) the partner organization also informed the DBR team that parents found the adaption process desirable and wanted to change the status quo. Another risk the team discussed was if the adaption process would bring about higher quality lessons, more student learning, and reduce the achievement gap on the state test. The team chose not to focus on this risk because (a) there is considerable research showing that adaptation process supported improve student engagement and learning (e.g. Debarger et al., 2017; Penuel, Gallagher, & Moorthy, 2011), and (b) their partners had an impressive track record in guiding teachers through an adaption process that both increased student learning and reduced the achievement gap on the state test. This is not to say that the DBR team should not conduct build and test cycles later that would address these risks, which would involve engaging and working with other stakeholders. Rather, the team decided that for that particular 2-week build and test cycle the biggest risks related the extent the online interface supported the teachers' goals.

In applying the logic of effective iteration, in just two weeks the project *Assist* team were able to avoid wasting time on a solution what would not support stakeholder goals or build impactful theory. As a result, they were able to quickly focus a more successful design and theory.

Conclusion

In educational DBR, it is all too easy for projects fail to support stakeholder goals or produce no new theoretical insights. To help DBR teams increase the likelihood that they support stakeholder goals, we propose a logic of iteration that answers 4 questions: (1) What are the core elements to iteration? (2) When should DBR teams iterate? (3) If DBR teams aim to iterate rapidly, how can they organize DBR to do so? (4) How can teams choose their focus when they iterate? We sought to answer these questions through the introduction of four key concepts: Building testing cycles, iterating early and frequently, slicing, and risking. (1) Building-testing cycles should aim to create designs that support stakeholder goals and establish how the designs fail to support these goals. (2) Iterating early and frequently suggests that DBR teams should conduct more rapid and early building and testing cycles in order to surface goals and understand the extent the design is supporting these goals. (3) Slicing suggests that DBR teams should build and test a small enough instantiation of their design argument that can be tested every cycle. (4) Risking suggests that the DBR team should focus on their biggest risk as a way to decide what to focus on. This methodological development can help the learning sciences better support projects that align with stakeholder goals, develop more impactful theory, and ultimately positively impact policy and practice.

References

Adams, R. S., Turns, J., & Atman, C. J. (2003). Educating effective engineering designers: The role of reflective practice. *Design Studies*, *24*(3), 275–294. https://doi.org/10.1016/S0142-694X(02)00056-X

Carlson, S. E., Maliakal, L. V., Lewis, D. G. R., Gorson, J., Gerber, E. M., & Easterday, M. W. (2018). Defining and Assessing Risk Analysis: The Key to Strategic Iteration in Real-World Problem Solving. In J. Kay & R. Luckin (Eds.), *Rethinking Learning in the Digital Age: Making the Learning Sciences Count, 13th International Conference of the Learning Sciences (ICLS) 2018* (Vol. 1, pp. 352–359).

Collective, D.-B. R. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, *32*(1), 5–8.

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design research: Theoretical and methodological issues. *The Journal of the Learning Sciences*, *13*(1), 15–42.

Debarger, A. H., Penuel, W. R., Moorthy, S., Beauvineau, Y., Kennedy, C. A., & Boscardin, C. K. (2017). Investigating purposeful science curriculum adaptation as a strategy to improve teaching and learning.

Science Education, *101*(1), 66–98.
Dede, C. (2004). If design-based research is the answer, what is the question? A commentary on Collins, Joseph, and Bielaczyc; diSessa and Cobb; and Fishman, Marx, Blumenthal, Krajcik, and Soloway in the JLS special issue on design-based research. *The Journal of the Learning Sciences*, *13*(1), 105–114.
DiSalvo, B., Yip, J., Bonsignore, E., & Carl, D. (2017). Participatory design for learning. In *Participatory Design for Learning* (pp. 3–6). Routledge.
DiSessa, A. A., & Cobb, P. (2004). Ontological innovation and the role of theory in design experiments. *The Journal of the Learning Sciences*, *13*(1), 77–103.
Easterday, M. W., Rees Lewis, D. G., & Gerber, E. M. (2016). The logic of the theoretical and practical products of design research. *Australasian Journal of Educational Technology*. https://doi.org/10.14742/ajet.2464
Easterday, M. W., Rees Lewis, D. G., & Gerber, E. M. (2017). The logic of design research. *Learning: Research and Practice*, 1–30. https://doi.org/10.1080/23735082.2017.1286367
Ehn, P. (1988). *Work-oriented design of computer artifacts* (PhD Thesis). Arbetslivscentrum.
Gravemeijer, K., & Cobb, P. (2013). Design research from a learning design perspective. In T. Plomp & N. Nieveen (Eds.), *Educational design research* (pp. 72–113). Netherlands Institute for Curriculum Development (SLO).
Gutiérrez, K. D., & Jurow, A. S. (2016). Social design experiments: Toward equity by design. *Journal of the Learning Sciences*, *25*(4), 565–598.
Herrington, J., McKenney, S., Reeves, T., & Oliver, R. (2007). Design-based research and doctoral students: Guidelines for preparing a dissertation proposal. *EdMedia+ Innovate Learning*, 4089–4097. Association for the Advancement of Computing in Education (AACE).
Jonassen, D. H., & Hung, W. (2015). All problems are not equal: Implications for problem-based learning. In A. Walker, H. Leary, C. Hmelo-Silver, & P. A. Ertmer (Eds.), *Essential Readings in Problem-Based Learning* (pp. 7–41). Purdue University Press.
Lewis, C., Perry, R., & Murata, A. (2006). How should research contribute to instructional improvement? The case of lesson study. *Educational Researcher*, *35*(3), 3–14.
McKenney, S., & Reeves, T. C. (2018). *Conducting educational design research*. Routledge.
Penuel, W. R., & Gallagher, D. J. (2017). *Creating Research Practice Partnerships in Education.* ERIC.
Penuel, W. R., Gallagher, L. P., & Moorthy, S. (2011). Preparing teachers to design sequences of instruction in earth systems science: A comparison of three professional development programs. *American Educational Research Journal*, *48*(4), 996–1025.
Rees Lewis, D. G., Gorson, J. S., Maliakal, L. V., Carlson, S. E., Gerber, E. M., Riesbeck, C. K., & Easterday, M. W. (2018). Planning to Iterate: Supporting Iterative Practices for Real-world Ill-structured Problem-solving. *Rethinking Learning in the Digital Age: Making the Learning Sciences Count*, *1*, 9–16. London, UK: International Society of Learning Sciences.
Sandoval, W. (2014). Conjecture Mapping: An Approach to Systematic Educational Design Research. *Journal of the Learning Sciences*, *23*(1), 18–36. https://doi.org/10.1080/10508406.2013.778204
Sannino, A., Engeström, Y., & Lemos, M. (2016). Formative interventions for expansive learning and transformative agency. *Journal of the Learning Sciences*, *25*(4), 599–633.
Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York: Basic Books.
Yackel, E., & Cobb, P. (1996). Sociomathematical norms, argumentation, and autonomy in mathematics. *Journal for Research in Mathematics Education*, 458–477.

Acknowledgments

We thank the Delta Lab for continual support for this work. This work was funded by the National Science Foundation (US) grants IIS-1530833 and IIS-1530837.