Polaris: Scaffolding the Creation and Evaluation of Design Arguments for Novice Designers

LEESHA V. MALIAKAL, Northwestern University
DANIEL REES LEWIS, Northwestern University
BOMANI MCCLENDON, Northwestern University
SAMEER SRIVASTAVA, Northwestern University
MATTHEW EASTERDAY, Northwestern University
DARREN GERGLE, Northwestern University
HAOQI ZHANG, Northwestern University

It is increasingly important to train novice designers to design effective solutions for today's complex problems. Despite existing approaches for training novices, students struggle to identify reasons why their designs may fail - a fundamental expert practice that novices must learn to lead creative work. We contribute Polaris, a learner-centered diagnostic tool that scaffolds novices to identify critical risks in their design argumentation. Polaris embeds expert knowledge and practice into templates and prompts that focus a novice's attention on structural issues in their design argumentation: gaps, lack of depth, and misalignment. Our study with 13 undergraduates leading 10 independent design-research projects demonstrates that students using Polaris identified structural issues in their design arguments that mentors considered critical and actionable for coaching. Our findings suggest that advancing tools that scaffold students to identify structural issues in their design argumentation can better help students focus their plans on mitigating critical design risks.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**.

ACM Reference Format:
Leesha V. Maliakal, Daniel Rees Lewis, Bomani McClendon, Sameer Srivastava, Matthew Easterday, Darren Gergle, and Haoqi Zhang. 2021. Polaris: Scaffolding the Creation and Evaluation of Design Arguments for Novice Designers. In *CHI '21, May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/xxxxxxx.xxxxxxx

1 INTRODUCTION

It is increasingly important to train novice designers in the skills they need to design and research solutions to today's most pressing problems. A fundamental skill novice designers must learn is to recognize where designs may fail and how to improve them. The ability to identify critical issues in *design argumentation*—or the underlying rationale that justifies a design [10, 12, 14, 26, 28]—allows expert designers to recognize flaws in a design and to focus their planned next steps on mitigating such issues, thereby improving their designs through iterative planning [5–8, 11, 14, 20]. In contrast, novice designers often struggle to focus on critical issues in their design argumentation when they evaluate

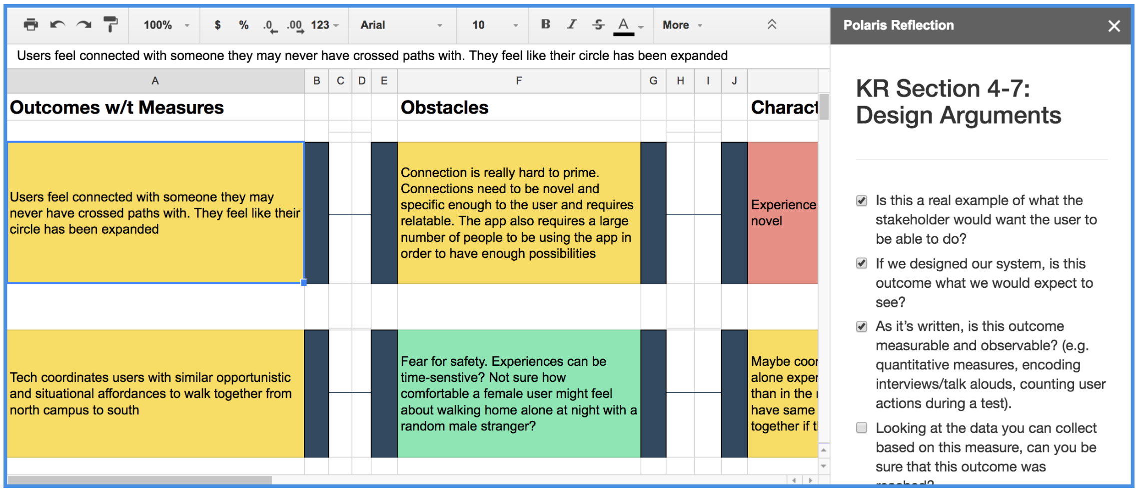

Fig. 1. We contribute **Polaris**, a learner-centered diagnostic tool that embeds expert knowledge and practice to scaffold novices to construct and evaluate their design arguments. Novices use the Design Argument Template (left) and the Reflection Prompts (right) to visually and procedurally evaluate their design arguments for structural issues.

their designs. This limits their ability to design effective solutions, as they are unable to recognize critical risks in their designs based on which to plan effective design iterations on their own [8].

The research literature and our own needfinding suggest that novices struggle with identifying critical issues in design argumentation because they lack the structural knowledge and diagnostic practices that expert designers have and use to evaluate designs [5–8, 14, 20]. Novices often focus their efforts on surface-level issues (e.g. improving the aesthetic of a website) and rely heavily on mentors to identify deeper structural issues in their design argumentation (e.g. justifying how the content of the website is serving a user need) [5–8, 23]. While existing pedagogy is adept at teaching design skills and processes such as storyboarding and design sprints, it often fails to train novices on how to improve their design argumentation by focusing on structural issues [8, 19, 20, 25, 29]. While expert design mentors are highly effective at coaching novices on recognizing and addressing such issues in their design argumentation, coaching is time-intensive and such experts are often limited resources in design learning communities [24, 25, 27, 29]. As a consequence, students often receive insufficient coaching to meaningfully improve their design argumentation skills.

To address this gap, we contribute *Polaris*, a learner-centered diagnostic tool that scaffolds novices to visually and procedurally identify key structural issues in their design argumentation (see Figure 1). Polaris embeds expert structural knowledge and diagnostic practice into a *design argument template* and *reflection prompts* that focus a novice's attention on three types of issues: gaps (i.e. are there components of the argument that are missing?), lack of depth (i.e. what is the depth required for each component to form a convincing argument?) and misalignment (i.e. what are the links required between and across components to form a cohesive argument?). The design argument template (Figure 1, left) structures the four components of a design argument: outcomes, obstacles, characteristics, and arguments, and visualizes the relationships between them via conceptual links. In contrast to existing tools that primarily help novices represent the components of a design problem (e.g. breaking it down into a problem statement and a solution space) [1–4, 9, 15–17, 21–23], Polaris' template helps novices visualize the relationship *between and across* components so that they can more easily detect structural misalignments (e.g., a proposed characteristic does not address the specific obstacle that it is linked to). To help novices focus on specific structural issues that design experts and mentors often catch, Polaris additionally provides reflection prompts (Figure 1, right) on lack of depth and misalignment issues for each component of a design argument (e.g., "Does the obstacle clearly describe how it prevents the linked outcome from being reached?"). This helps novices procedurally focus their evaluation on identifying core structural issues within and across the components of their design argument, and to avoid fixating on surface issues.

We hypothesize that Polaris' templates and prompts can help novice designers identify critical issues in their design arguments, and can inform how mentors coach students to devise plans to address them. To test this hypothesis, we conducted a pilot study with 13 undergraduates leading 10 independent research projects. Of the 91 issues that students identified with Polaris, 95.6% of the issues focused on core design argument components, and 96.7% of issues focused on three core types of structural issues: gaps, a lack of depth, or misalignment in their design argumentation. To explore the quality of the issues from the perspective of a mentor, we also asked mentors to generate their own issues and to evaluate the issues generated by 6 of the 10 teams (71 issues reviewed). Findings show that on average, mentors rated the issues that students identified as accurate (4.06/5) and severe (4.11/5). When comparing mentor and student issues, findings also show that students identified 55.17% of the critical issues identified by mentors. Moreover, mentors felt that student issues expressed a student's current understanding, and informed how mentors would coach students to overcome misunderstandings. These findings suggest that diagnostic tools such as Polaris can scaffold novices to identify structural issues in their design argumentation and to focus their plans on mitigating the most critical risks in their designs.

2 BACKGROUND

Our work explores how to scaffold novice designers to identify critical structural issues in their design argumentation. In order to understand how we might design such scaffolds, we first review how experts use design argumentation to improve their designs, and why novices struggle. We then review limitations to training design argumentation in current pedagogy, and the limitations of existing tools for scaffolding novices to evaluate and improve their design argumentation.

2.1 How Experts Use Design Argumentation

Experts evaluate and improve their designs by identifying issues in their *design argumentation*, or the underlying rationale that justifies their designs [10, 12, 14, 26, 28]. Experts use design argumentation to (1) build up an understanding of the problem space through problem representations, (2) articulate detailed and novel solutions to address those problems, and (3) construct arguments for why they believe the solution will address the underlying problems [10, 14, 28]. Prior literature underlines the importance of using design argumentation to plan out practical next steps in design work. To plan out iterations that advance designs in impactful ways, expert designers focus on evaluating and improving their underlying design arguments with each iteration [12, 26].

However, novices struggle to evaluate their designs through design arguments because they lack the structural knowledge and diagnostic practice that experts have [5, 6, 14, 20]. Literature on design expertise find that novices do not diagnose their designs in the ways that experts do [5, 7, 8, 18]. In contrast to experts who "focus on problematic areas" of the design, novice designers have "unfocused, non-analytical, and diffused" ways of evaluating when troubleshooting their designs and planning next steps [8]. Recent work suggests that novices struggle to diagnose their designs because they lack knowledge of (a) design problem structure to focus attention on key components, and of (b) potential issues to diagnose different risks in the design [6]. For instance, a novice might claim that their user's core obstacle is that "existing solutions don't have this feature that our design has" rather than explaining *why* existing solutions don't work. Often, this is because novices lack the structural knowledge to differentiate between why a user is actually struggling vs. the user not having the proposed solution that they think would help. Similarly, a novice may state an imprecise desired outcome because they don't yet recognize the need to have measurable outcomes when testing their designs. Without this knowledge of design problem structure and potential risks, novice designers will struggle to evaluate their design argumentation as experts do.

To train students to evaluate their design argumentation for critical risks that can drive meaningful iterations, we must first understand the types of critical issues experts find in design argumentation. We can then embed these expert models into effective scaffolds for novices. To address this need, our work characterizes three types of structural issues in design argumentation that experts often catch (gaps, lack of depth, and misalignment), and contributes a tool that scaffolds novices to focus on identifying such issues when evaluating their designs.

2.2 Limitations to Training Design Argumentation in Pedagogy

Current design education trains a wide range of design skills that make up expert practice. For instance, design practitioners and educators scaffold novice designers by introducing them to common design processes and practice [8, 20], such as needfinding, storyboarding, and prototyping. More recently, agile methodologies have been adopted in design practice (e.g., design sprints) to help innovators incrementally plan and monitor their progress [19, 25, 29].

However, existing pedagogy often fails to train novices on the design argumentation that is the backbone of effective design. For example, a novice designer may use a combination of user stories and prototyping techniques to develop a low-fidelity design that they believe will realize the user story. However, when a novice fails to argue for how the design addresses a root challenge that is preventing the user story from being realized, this opens up the possibility for a failed design solution that lacks justification. Similarly, while a novice designer may use methods like retrospectives and sprint logs to identify issues and plan next steps in their project, they may still focus on surface-level issues (e.g. improving the UI color scheme of an informational election voting website) rather than on deeper issues that an expert might notice (e.g. including content that voters need when visiting the site). While novices can improve aspects of their designs by employing such design process and practice, novices that struggle to construct and evaluate design arguments may continue to generate designs that lack mechanistic explanations, and are unlikely to be effective as a result [8].

While design experts can effectively coach students to focus on critical issues in design argumentation, such experts are often a limited resource in design learning communities [24, 25, 27, 29]. Prior work explores interventions to support mentors in coaching novice designers to identify critical issues in their projects and planning effective next steps to address them, for example by using diagnostic questioning to understand critical issues in students' research projects [29] and developing tools that surface a design team's plans to design coaches to facilitate feedback [24, 25, 27]. While effective, coaching interventions remain mentor-intensive, and mentors' time is still a limited resource in many design learning communities [25, 29]. Consequently, we've observed that when mentors spend significant time diagnosing issues during a coaching session, it can leave little time to coach students on devising plans to address them.

2.3 Scaffolding Novices to Evaluate and Improve Design Argumentation

Prior research has extensively explored how to scaffold novices to visualize and evaluate different parts of their design problem. For instance, many representational tools help novices visualize general problem and solution components that make up the structure of a design problem [1, 9, 11, 15–17]. More recently, business and design practitioners have adopted *canvases* that help break down the different components of a design problem, such as community partners, users, and value propositions [19, 21–23]. Other researchers have focused on designing diagnostic tools that provide novices with a set of heuristics and a process to help them evaluate their designs for common risks [5–8, 13, 14, 23], such as diagnosing the design of a parachute using heuristics like whether it has a vent at the top.

However, even with such scaffolding, novices still struggle to identify the deeper structural issues in their design argumentation that experts focus on most. While representational tools like canvases can externalize components of design problem structure (e.g. a problem vs a solution), they often fail to represent deeper structure, such as the way in which experts conceptually link these components to form aligned design arguments. Similarly, while diagnostic tools can externalize heuristics to help novices evaluate a design for common risks, existing tools either fail to capture the generalizable depth of argument that experts think about [7, 8], or provide so many heuristics that novices overlook critical issues in their evaluation [5, 6, 23]. For instance, heuristics for parachutes may be specific to conceptual knowledge about parachute design (e.g. the role of a vent), rather than highlighting general, structural issues in design argumentation (e.g. the measurability of a desired outcome). Without representing or prompting novices to reflect on the deeper structure of their design argumentation, novices can fail to learn the ways that experts identify such issues and focus instead on the surface-level issues that appear most salient to them.

While there are some expert systems for supporting design argumentation, these systems are limited in their usefulness, usability, and generalizability for novice designers. A review of these systems [26] found that they over-focus on argumentation logic, which distracts even experts from planning practical design implementations, thereby limiting the practical usefulness of such tools. Moreover, studies show that even expert designers face cognitive constraints in navigating such complex representations, suggesting significant limits to usability and usefulness of such systems for novice designers. In another vein, there have been some systems designed to scaffold novices to evaluate and improve their design argumentation for specific design problems such as kitchen design [12]. While studies show that such tools are usable and useful for novices, these interventions focus on conceptual issues that are domain-specific, rather than generalizable, structural issues in their design arguments.

Filling this gap, Polaris embeds the structural knowledge and diagnostic practice of experts into generalizable templates and prompts that scaffold novices to evaluate their design arguments for critical issues. Unlike prior approaches, Polaris scaffolds novices to focus their attention on the types of structural issues that experts generally find in design argumentation: gaps (i.e. are there components of the argument that are missing?), lack of depth (i.e. what is the depth required for each component to form a convincing argument?) and misalignment (i.e. what are the links required between and across components to form a cohesive argument?).

3 UNDERSTANDING EXPERT PRACTICE FOR EVALUATING DESIGN ARGUMENTATION

We first sought to understand how expert design mentors coach students to identify critical structural issues in their design argumentation. We observed planning meetings in an Agile Research Studio [29] led by one of the authors and another early career faculty mentor. In the studio setting, undergraduate students learn to lead systems HCI research with the support of faculty and graduate student mentors. For example, a student may design, implement, and study developer tools that help novice web developers make sense of real-world code. Students had been enrolled in the class and working on their projects between 1 and 5 quarters (3 to 15 months). As part of the studio structure, faculty and graduate student mentors coach students on their planning during weekly planning meetings. In this context, we conducted field observations across 25 student-mentor planning discussions in 10 planning meetings over the course of five weeks. A typical meeting lasts 1 hour, during which 2-3 different undergraduate research teams take turns getting feedback on their plans. This resulted in 10 hours of observation and 90 pages of written field notes that we inductively coded for themes around novice challenges and coaching strategies, which we synthesize below.

3.1 Student issues don't focus on design argumentation

We found that students often raised surface-level issues in planning meetings and struggled to identify critical issues in their design argumentation. Students often raised surface-level challenges, such as "tech is really hard to test" or "building tech took longer than I expected," rather than deeper structural issues in their argumentation. Even in instances where students did relate issues to their design arguments (e.g. "I struggle with chaining my design arguments"), issues were not specific enough to support a useful coaching discussion on how to address them. We also observed that mentors frequently asked students diagnosis-type questions to understand how the surface-level issues might relate to more critical issues in the student's design arguments, and redirected their attention to focus on argumentation. From these observations, we learned that tools would need to scaffold students to focus on evaluating their design argumentation, and to support them diagnosing critical structural issues in their arguments as their mentors did.

3.2 Mentors coach students to identify structural issues in design arguments

From our observations, we found that mentors spent most of their coaching efforts helping students identify critical structural issues in design argumentation before they could coach students through plans to address them. We highlight and focus on three types of structural issues that mentors often coached students to recognize during these meetings: gaps, a lack of depth, and misalignment. Mentors would often point out specific *gaps* in a student's design argument structure, such as not articulating the desired outcome of their design intervention. Mentors would also use probing questions to help students articulate details and *depth* they felt were lacking in their design rationale, such as suggesting outcomes that were not concrete enough to be measurable. Mentors also noted when they saw fundamental *misalignment* between parts of a student's design argument, such as noting when a proposed solution was misaligned with the user challenge they set out to overcome. More generally, novices experience challenges in identifying structural issues such as gaps, a lack of depth, or misalignment because they lack the structural knowledge and diagnostic practice that experts have and use, such as the components that compose design arguments, the ways in which components should be detailed, and the causal relationships between them [5–8, 14, 20]. By embedding these expert models into scaffolds, novices might be able to better argue for comprehensive solutions to problems that include the detail necessary for articulating underlying mechanisms of why they expect their designs to work.

4 POLARIS SYSTEM

To help novice designers construct and evaluate design arguments, we introduce *Polaris*, a learner-centered diagnostic tool that scaffolds novices to visually and procedurally identify key structural issues in their design argumentation (Figure 1). Polaris is composed of (1) a *design argument template* that embeds expert structural knowledge and (2) a series of *reflection prompts* that embed expert diagnostic practice for evaluating their design argument structure for gaps, a lack of depth, and misalignment. By embedding expert knowledge structures and diagnostic practice, we argue that Polaris can scaffold students to evaluate their own design arguments for critical structural issues they may have previously missed. We detail the features of Polaris below.

components of design argument template	**outcome w/t measures:** the specific, measurable, and attainable goal for a design if it is successful. The outcome is a practical design goal rather than theoretical research goal.	**obstacle:** the challenge your design is seeking to overcome in order to reach the outcome.	**characteristic:** the specific set of attributes that define your design and differentiate it from other designs.	**argument:** the reasons for why the characteristics will help overcome the obstacles to achieve the desired outcome.
example of student design argument	*"Tech coordinates users with similar opportunistic and situational affordances to walk together from north campus to south"*	*"Fear for safety. Experiences can be time-senstive? Not sure how comfortable a female user might feel about walking home alone at night with a random male stranger?"*	*"Maybe coordinate the walking home alone experience in the day? Rather than in the middle of the night. Or have same gender users walk together if the event stays at night"*	*"By having tech coordinate easy and opportunistic face to face meeting, a stronger possible emotional connection can be formed. And it's not a burden of the user's time"*
example reflection prompts	Is this a real example of what the stakeholder would want the user to be able to do?	Does this obstacle encompass a core user or stakeholder struggle?	Does this characteristic uniquely describe your approach to solving the problem? Does the characteristic clearly distinguish your design from other possible designs (especially existing designs that do not solve the problem)?	Is there existing evidence from knowledge of problem and users that supports this argument?

Fig. 2. How the Design Argument Template and Reflection Prompts in Polaris help novices construct and evaluate their design argument structures

4.1 Design Argument Template

The *Design Argument Template* is composed of four **components** that represent the structure for the design argument – Outcome with measures, Obstacle, Characteristic, and Argument (see Figure 2). To compose a design argument, a student first details the parts of their design argument through each component of the template, which correspond to components of an expert design argument structure (Figure 1, left). The design argument template allows students to represent multiple design arguments in different rows, which enables them to represent and distinguish between the different design arguments that together compose the full scope of their interventions. We argue that visually representing the components of an expert's knowledge structure will help novices identify specific gaps in their own mental model (e.g., not knowing the obstacle that is preventing an outcome from being reached).

The components of a design argument are connected via **conceptual links** that visualize the relationships between them. Students can create visual links between components (see navy links in Figure 1, left) as a way to emphasize the relationships between parts of the design argument. For example, when a student creates a link between a Characteristic component and an Obstacle component, it reinforces the idea that that a characteristic of the design is meant to overcome that specific obstacle that an end-user may face. We argue that providing links between components reinforces the causal relationships that experts use for mechanistic alignment between the components of their design arguments (e.g., this characteristic overcomes this obstacle to achieve this outcome because of this argument).

4.2 Reflection Prompts

The *Reflection Prompts* are a set of questions associated with each component of the design argument that scaffold students to evaluate the structure of their argument using heuristics based on expert diagnostic practice (see Figure 1, right). For example, when a student selects an Argument component, they will see an associated prompt that asks them: "Is there existing evidence from literature that supports this argument?" As a student evaluates each component of a design argument with the reflection prompts, they can check off each prompt that their current design argument satisfies. As more prompts are checked off for a component, the color of that component of that design argument changes from red to yellow to green, visually cueing students to recognize weak areas in their overall argumentation.

To build a set of reflection prompts that represent expert diagnostic practice, we coded the common probing questions that mentors asked across the 10 planning meetings we had observed that related to structural issues with respect to lack of depth and misalignment. See Figure 2 for some examples of reflection prompts. We argue that providing students with a set of *depth-focused prompts* will help them procedurally work through the ways in which their argument may be lacking the necessary details for a well-argued claim. Similarly, we argue that *alignment-focused prompts* give students a way to procedurally evaluate the relationships between components that must be established for a sound argument, and thus recognize where misalignment issues may occur. For components where we had too few reflection prompts, we further interviewed mentors and asked them to provide additional questions that students can use to identify critical issues. We then piloted the prompts with 2 research teams who helped us revise wording for clarity.

5 POLARIS PILOT STUDY

We tested Polaris in a pilot study with 13 undergraduate design-researchers who led ten HCI research projects, and three mentors. Our primary goal in this pilot was to understand how Polaris might scaffold students to identify critical issues in the structure of their design arguments, and how it might inform the ways in which mentors coach students to address them. In Part 1 of the study, undergraduates used Polaris to construct design arguments, and list any issues they identified. In Part 2 of the study, we asked mentors to review students' arguments, list their own issues, and then review the issues students identified. Our core hypotheses were as follows:

(1) Students will identify issues focused on design argumentation.
(2) Students will find structural issues within their design arguments such as gaps, lack of depth, and misalignment.
(3) Students will find issues that (a) mentors consider to be critical and (b) are similar to issues identified by mentors.
(4) Mentors will find student issues and design arguments useful for directing their coaching decisions.

We detail our pilot study methodology below.

5.1 Pilot Study Setting

We sought to test Polaris in a setting where students were both novices in constructing and evaluating design arguments, and had limited access to mentoring resources who could coach them in evaluating their arguments for structural issues. We tested Polaris in an Agile Research Studio [29] (same setting as our needfinding study, described above), in which students learn to lead systems HCI research with the support of faculty and graduate student mentors. Student participants were undergraduate Computer Science majors, and had been enrolled in the class and working on their projects between 1 and 5 quarters (3 month to 15 months), during which the students are introduced to design arguments. The three research mentors had between 3-7 years of experience mentoring HCI design research.

5.2 Part 1: Student Issue Identification

The first part of the study focused on understanding how Polaris can help novices evaluate and identify structural issues in their design argumentation. Students used Polaris to evaluate their current design arguments for their projects and documented any issues they found. To differentiate between the mechanisms by which the design argument template and the reflection prompts in Polaris helped students identify structural issues, we had students identify issues with just the design argument template, and then again with the reflection prompts alongside the template. The procedure was as follows:

(1) **Students construct Design Arguments (15 mins):** Undergraduates wrote their design arguments into the four sections of the design argument template, and created links between the sections.
(2) **Students identify issues with template only (10 mins):** Undergraduates were asked to assess their design arguments in the Polaris tool and write down any issues they found into a separate document. They were not allowed to use the reflection prompts during this part of the test.
(3) **Students identify issues after using prompts (10 mins):** Undergraduates were asked to open the reflection prompts feature and evaluate their arguments using the provided prompts, and again write down any issues they found in a separate document.

To evaluate how Polaris supported students in identifying structural issues within their design arguments, the primary author coded these issues per the three types of critical structural issues identified in needfinding: gaps, depth, and misalignment. To evaluate how different features in Polaris helped students identify issues across components of their design argument, we coded all student issues based on which components of the design argument the students referenced. We used 5 codes in this analysis, one code for each of the four sections of the design argument, and a final "other" code for issues which did not reference the design argument. Issues that referenced multiple components were given multiple codes. To understand the mechanisms behind how the design argument template and the reflection prompts supported students in identifying these structural issues, all coding was conducted blind to whether students identified these issues using the design argument template, or after using the reflection prompts alongside the template.

5.3 Part 2: Mentor Evaluation and Use of Student Issues

The second part of our study focused on understanding the quality of the issues students identified according to their mentors, and how mentors might use those issues to coach students through misunderstanding or through plans to address critical issues. Mentors evaluated the design arguments of a subset of students (6 out of the 10 teams) and documented any issues they found. We then asked mentors to rate the issues students identified, and generate coaching feedback per issue. The procedure was as follows:

(1) **Mentors identified issues (10 min per team):** Each mentor reviewed the design arguments of two of their student teams and generated a list of issues they found. Mentors then rated how critical their issues were on a 1-5 Likert scale (1=not critical; 5=very critical). Mentors did not see student generated issues before generating their own.
(2) **Mentors assessed student issues (10-15 mins):** We asked mentors to evaluate the issues that two undergraduate mentee teams identified, and rate them 1-5 on a Likert scale on four aspects: accuracy (i.e. whether mentors agreed on the issue), severity (i.e. criticality of issue to the design success), articulation (i.e. the expressiveness of the issue), and actionability (i.e. usefulness for coaching). Mentors could see both the issue list, and students design arguments.

(3) **Mentor wrote feedback per issue (20-25 mins):** We asked mentors to write their diagnosis and feedback on each issue their mentees wrote. Mentors could see both the issue list, and students design arguments.
(4) **Mentors wrote reflections on their use of Polaris (10-15 minutes):** We asked mentors to reflect upon their use of Polaris in comparison to their normal mentoring. This process generated the following data: (a) mentor ratings of student issues, (b) mentor diagnosis and feedback from issues, and (c) mentors written reflections on their experiences using Polaris in comparison to their mentoring meetings.

To evaluate the criticality of the issues that students found, we used mentor ratings of the issues students raised on the dimensions of accuracy and severity. To evaluate the degree to which students using Polaris were able to identify issues similar to their mentors, we compared mentor and student issues for the same design arguments that mentors had rated 3 or higher in severity. We then matched the student issues for each project to related mentor issues. Because student issues may vary in their degrees of similarity to a mentor issue, we defined the degrees to which a student issue was similar to their mentor's issue based on the structural similarity (0 = not related; 1 = referring to same component; 2 = same structural issue (e.g. gap, depth, alignment) but lacking specificity; 3 = same structural issue with some specificity; 4 = same structural issue with more specificity, including references to relevant project detail; 5 = near identical issue). The following example illustrates a match between a mentor and student issue with a similarity degree of 3. The mentor noted that a particular outcome for a design argument lacked depth and were not specific enough to be measurable: *"The outcomes [in outcome 2] identified in the design argument are too general, and therefore are not measurable."* Similarly, the student also noted that the same way in which this outcome lacked depth: *"For Outcome 2, difficult/ambiguous to measure because we don't have observable actions we'd be able to see if our design was successful"*.

To evaluate whether mentors could make sense of student understanding from the issues they raised, and whether they found issues useful for their own coaching, we had mentors also rate the subset of student issues (71 issues across 6 teams) on articulation and actionability for their own coaching. We then asked mentors to offer coaching feedback per issue, and note patterns in student misunderstanding. Finally, we qualitatively analysed the specific feedback that mentors gave students per issue, as well as the broader patterns of misunderstanding that mentors could identify. The primary author conducted multiple rounds of deductive coding on 71 instances of mentor feedback. We use two primary codes in our analysis: feedback focused on (a) planning next steps to address issues or (b) project or conceptual misunderstandings.

6 POLARIS FINDINGS

6.1 Students using Polaris identified issues focused on design argumentation

Students using Polaris raised 91 issues, 95.6% (87/91) of which were related to one or more components of their design argument, rather than surface level issues. 47 of the 91 issues were identified after using the template, and an additional 44 issues after using the reflection prompts. Student issues spanned all four components of the design argument; see Figure 3 for a component-by-component breakdown with examples of issues students raised. Issues students raised include recognizing mismatches between a stated outcome and what stakeholders would actually want; lacking reasoning behind why an obstacle exists; lacking details and clarity in characteristics; and missing explanation for how a characteristic overcomes an obstacle to reach a particular outcome in argument.

component	# of issues	examples
outcome	22	*"The Outcome for DA 2 is not a real example of what a stakeholder would want"*
obstacle	17	*"My obstacles should have reasons. As they're specified they really only state a vague problem, not reasons for that problem that I can actually address"*
characteristic	13	*"Our characteristic wasn't detailed enough for someone to be able to replicate our interface because we didn't explain what "points" or "pop-ups" were"*
argument	19	*"Need to explain better why the characteristic overcomes the obstacle to reach the desired outcome"*
other	4	*"Everything is too short"*

Fig. 3. A distribution of issues students found across components of the design argument

structural issue	# of issues	examples
gaps	13	*"We couldn't think of any characters to overcome or address our original obstacle (being able to detect the growth mindset in coding, both generally and automatically/passively)"*
misalignment	10	*"Inconsistency between parts of chain -> Parts of obstacles not relevant to ideal outcome, characteristics not addressing the obstacles stated"*
lack of depth	70	*"Need to specify what observable actions I'd be able to see if the design is successful"*
not structural issues	3	*"For the Characteristics section for DA 1, Other peers/mentors have not read my arguments"*

Fig. 4. The types of structural issues students found when evaluating their design argumentation

6.2 Students identified structural issues in design arguments

Students using Polaris identified three types of structural issues in their design argumentation: gaps in their argument, a lack of depth within components of their argument, and misalignment between components of their argument. Of the 91 issues students found, 96.7% (88/91) of the issues were related to at least one of these three structural issues; see Figure 4 for a breakdown across issue type with examples of issues students identified. Notably, students found many more depth issues than other types of structural issues, which we expected given the number of possible depth issues relative to gaps and misalignment. We analyze the issues students identified for each type of structural issue below.

6.2.1 Gaps in design argument structure. 9 of 10 student teams identified 13 instances of gaps in their design arguments after using Polaris. In analyzing these instances, we found that students identified three kinds of gaps: missing components, missing design argument, and mismatched components; see Figure 5. The most common type of gap that students identified using Polaris was when one of their design arguments was lacking one of the four fundamental components (outcome, obstacle, characteristic, argument). For example, one student issue noted *"We couldn't think of any characteristics to overcome or address our original obstacle."* We also saw cases where students missed a design argument for a core feature of their system (*"Missing arguments for situationally aware."*), and where students wrote the same thing for two different components, suggesting a gap due to mismatched component (*"...what actually is the outcome? The way I've written it, I'm talking about the characteristics of the spaces, not the actual outcome."*) These findings suggest that having students detail each component of each design argument in a template may also help them recognize when they are not distinguishing between parts of an argument, or are missing a design argument all together.

6.2.2 Lack of depth in design argument structure. 9 of 10 student teams identified 70 depth issues in different parts of their design arguments after using Polaris. We found that students identified 22 depth issues after using the template, and 27 additional depth issues after using the prompts alongside the template.[1] It's possible that students were able to identify many depth issues with templates alone because more experienced students may already be familiar with some depth issues that have come up in prior coaching meetings. Still, after using the prompts, students found many more depth issues, which suggests that the prompts helped students focus on specific heuristics that they had not considered when they initially evaluated their design arguments using just the template.

To evaluate whether depth-focused prompts helped students identify more detailed issues than with the template alone, we also looked for paired instances of issues where a student found a depth issue with just the template, and then used the prompts to articulate a more specific version of the issue. We found 4 instances from 4 of the 10 student teams that went from a less specific issue with the templates to a more specific issue after having used the prompts. Below is an illustrative example: after using the design argument template, a student wrote the issue *"I feel that my obstacles for DA 1 are not as specific as I'd like them to be, but I'm also unsure how to write them succinctly"*. The same student wrote this more specific version of the issue after using the prompts: *"In the Obstacle section for DA 1, I don't think I've captured both stakeholders"* Given that one of the Obstacle prompts maps directly to the detail in the issue ("Does this obstacle encompass a core user or stakeholder struggle?"), having the prompt likely helped the student more specifically articulate this issue. While there were not many paired instances, we see some evidence that the prompts may have helped students articulate more specifically what the depth issues were in their design arguments.

6.2.3 Misalignment in design argument structure. 6 of the 10 student teams identified 10 issues related to misalignment between components of their design argument. Within these issues, students noted 12 specific relationship links where they found misalignment, which we coded into three types of misalignment: (1) between two components, (2) across multiple components, and (3) general misalignment where no particular relationship was specified. 75.0% (9/12) of misalignment that students identified were between two components (see Figure 6). Of the 9 misalignment issues between components, 5 were about the relationship between outcomes and obstacles. A possible explanation for students catching so many misalignment issues between components, especially between outcome and obstacle, might

[1] When comparing depth issues found with templates and after prompts, we removed 21 issues about measurable outcomes when comparing template and prompt issues, because this depth heuristic for outcomes had been encoded in both the template header (Outcome w/t measures), and as reflection prompts.

gap type	# of issues	definition	examples
missing components	6	a component of the design argument is not represented	"For Outcome 1, we don't know what our core obstacle is" "We couldn't think of any characteristics to overcome or address our original obstacle"
missing design arguments	3	one of the design arguments for the intervention is not represented	"Missing arguments for situationally aware" "Incomplete set of design arguments"
mismatched component type	4	two different components are not distinguished from one another	"...what actually is the outcome? The way I've written it, I'm talking about the characteristics of the spaces, not the actual outcome." "Design Argument seems circular. The outcome seems to be the same as the task item"

Fig. 5. The types of gaps students found in design argument structure with illustrative examples

be that students have internalized the relationship between outcome and obstacle more than others. This might be because they practice aligning outcomes and obstacles most often in the early stages of their design-research projects as they conduct needfinding and develop user stories, a precursor to any solution development like proposing or prototyping characteristics.

Overall, our findings demonstrate that Polaris supported students in identifying three core types of structural issues in their design arguments: gaps, lack of depth, and misalignment. By providing novices with scaffolds that embed expert structures and heuristics in visual templates and procedural prompts, novice design researchers were able to better recognize when the structural state of their own arguments are lacking the rigor that one might see in an expert's model. This suggests that students may be able to effectively identify core structural issues across their design arguments independently, rather than relying on a mentor coaching session to surface the structural issues they may have missed or not focused on in their own evaluation.

6.3 Students identified issues that mentors found accurate and severe

Mentors generally considered student issues to be accurate and severe. When asked on a 5pt Likert scale the degree to which mentors agreed that a student issue was accurate, mentors on average rated student issues as 4.06 on accuracy (standard deviation = 0.97; between agree and strongly agree). Similarly, mentors on average rated student issues as 4.11 on severity (standard deviation = 1.02; between agree and strongly agree). We also found that mentors considered 71.83% (51/71) of student issues to be accurate (ratings of 4 or 5) and 73.24% (52/71) of student issues to be severe (ratings of 4 or 5).

misalignment type	# of issues	examples
between components	9	*"Need a stronger link of argument to characteristics. The more specific the better"* *"Inconsistency between parts of chain -> Parts of obstacles not relevant to ideal outcome, characteristics not addressing the obstacles stated"*
across multiple components	1	*"Argument needs a lot more specific language, and needs to map characteristics to overcoming obstacles to achieve outcomes, and just doesn't"*
general nonspecific	2	*"Mappings aren't clear between sections of the DA (i.e. Not explicit, not clearly implicit)"*

Fig. 6. The types of misalignment students found in design argument structure with illustrative examples

When comparing the general severity of issues that mentors identified to the issues students identified while using Polaris, we found that student and mentor issues were similarly severe. We first asked mentors to rate the severity of their own issues and then the severity of student issues. We found that the average mentor rating of severity for student issues (4.11 , n = 71) was similar to the average mentor rating of severity for their own issues (4.34 , n = 32). These findings demonstrate that not only did mentors consider student issues as generally severe, but that the average severity of student and mentor issues were relatively similar.

6.3.1 Students identified the majority of severe mentor issues. We found that 55.17% (16/29) of severe mentors issues were topically similar to issues that their students independently identified with Polaris. This finding supports our original hypothesis by demonstrating that students were able to identify issues similar to the majority of the severe issues their mentors found. However, 44.83% (13/29) of those mentor issues were issues that students did not find with Polaris (see Figure 7). Through further analysis, we found that 8 out the 13 severe mentor issues that students did not find were outside the scope of the Polaris design. These issues tended to be (a) project-specific details or (b) related to parts of the research argument beyond design arguments. For example, for (a), a mentor noted the following project-specific issue that would not generalize into a tool for any student: *"In first design argument on learning, the design argument does not mention learning at all (i.e. discuss how changing locus of control will produce greater learning)."* For (b), a mentor noted the following issue in another part of the research argument beyond the fundamental design argument: *"Mismatch in tech-specific outcome and design-based obstacle. Obstacle is about whether people want to do it, but outcome is written about what the tech can do?"* Here, the mentor is distinguishing between an obstacle in a general design argument, and a technical system obstacle in a technical argument for things like system framework or architecture designs, the latter of which was out of scope for our initial design and evaluation of Polaris. Because Polaris is designed to support students in identifying common structural issues in their design arguments for their projects, we would not expect students to find either of these issues that were project-specific or beyond the design argument.

comparison	% coverage
Severe mentor issues that students also found	55.17% (16/29) of mentor issues
Severe mentor issues that students did not find	44.83% (13/29) of mentor issues

Fig. 7. Student coverage of critical structural issues that mentors identified

comparison	% coverage
Severe student issues that mentors also found	40.38% (21/52) of student issues
Severe student issues that mentors did not find	59.62% (31/52) of student issues

Fig. 8. Mentor coverage of critical structural issues that students identified

6.3.2 Students identified severe issues mentors had not caught. We also found that 59.62% (31/52) of the student issues that mentors considered severe (rating of 4 or 5) were issues that mentors did not find on their own (see Figure 8). A possible explanation for this finding might be that mentors themselves struggle to recall all the possible structural issues and heuristics for evaluating design arguments, which could be due to a multitude of factors. For example, mentors in the experimental setup and in practice face real-world time constraints when evaluating the arguments that students articulate and issues they raise, and are likely unable to exhaustively review a student's design arguments for possible issues in the context of a planning meeting. Similarly, mentors may choose to omit less critical structural issues from their discussions and focus on more critical ones given limited mentoring resources. Alternatively, it's possible that mentors may struggle to recall specific project details that students are more regularly considering when actively working on their own project. These findings suggest that Polaris can not only support students in catching issues that mentors consider severe, but that it may also help students catch critical issues on their own that mentors might overlook or skip over in time-constrained feedback sessions.

6.4 Mentors found issues articulate and actionable for coaching students

Beyond accuracy and severity, mentors rated student issues as generally articulate and actionable for their own coaching. When asked on a 5pt Likert scale, mentors on average rated student issues as 4.14 on articulation (standard deviation = 1.25), falling between agree and strongly agree, and 3.72 on actionability, (standard deviation = 1.06) falling between neutral and easily. We also found that mentors considered 73.24% (52/71) of student issues to be articulate (ratings of 4 or 5) and 73.24% (52/71) of student issues to be actionable (ratings of 4 or 5). Further, mentors generally found more articulate issues to also be more actionable for coaching students. For the issues that mentors considered articulate (rating of 4 or 5), the average actionability of those issues was 4.15 (n = 52). In contrast, for the issues mentors considered to be less articulate (ratings of 3 and below), the average actionability was 2.53 (n = 19). A possible explanation for these findings is that a more precise articulation of an issue makes the issue much more actionable for the mentor because mentors have more detail that illustrates the student's current understanding with greater clarity, which informs how the mentor would coach the student.

6.4.1 Mentors focused coaching on planning next steps when they agreed with student issues. When mentors considered student issues to be both accurate and articulate (ratings of 4 or 5), mentors were able to offer suggestions for planning next steps to address those issues. Of the 25 issues that mentors provided feedback for and considered accurate and articulate, 72.0% (18/25) of mentor feedback suggested plans for next steps to address the issue. This is in contrast to issues that mentors considered accurate but *inarticulate*, for which only 22.22% (2/9) of mentor feedback was about planning. Below is an illustrative example of a mentor giving planning feedback for a student issue that the mentor rated as articulation = 5 and accuracy = 5:

Student Issue: *"Design arguments is bias towards my own expectation/guesswork as opposed to addressing the needs stakeholders"* **Mentor Feedback:** *"This is awesomely clear. I would ask the student to structure a test to see what learners can or cannot do using basic/existing tools and designs. I would though first discuss if this is the outcome we actually care about – and see if we can structure or initial test to get a list of goals/tasks learners struggle with, before focusing on arguments for reaching this particular outcome"*

These findings are consistent with our hypothesis that for student issues that mentors agreed with and considered articulate, mentors would focus their feedback for students on how to address those issues. These findings suggest that by seeing issues that mentors consider accurate and articulate, mentors are able to offer planning feedback to students that is oriented around next steps to address core issues.

6.4.2 Mentors focused on student misunderstanding when issues were inarticulate or inaccurate. We also found that for the issues that mentors considered either inarticulate or inaccurate, mentors focused their feedback on correcting or clarifying the student's understanding of the issue. For issues that were accurate but inarticulate, 66.67% (6/9) of mentor feedback was focused on correcting project or argument misunderstandings that students may have exhibited in their issues. For inaccurate but articulate issues, 80.0% (8/10) of mentor feedback was about correcting student's understanding of the project or general design argument structure. Below is an illustrative example of a mentor identifying a specific misunderstanding and offering feedback for inaccurate but articulate student issues. Here, the mentor offered feedback to address the student's project-specific misunderstanding:

Student Issue: *"Outcome 1 needs to be more concrete – actions that promote learning vs. just promoting learning"* **Mentor Feedback:** *"If outcome 1 was more concrete, it would just become outcome 3. This outcome is intentionally measuring something less direct than outcomes 2 and 3. The ultimate goal of this design is to promote learning, and we hypothesize that creating better collaboration and engagement will increase learning. We want to measure both."*

These findings suggest that even when students express issues poorly or inaccurately, Polaris can still help mentors catch problems with a student's conceptual understanding of the project itself—a precursor to forming better arguments and understanding critical issues. Still, well-articulated issues are helpful (even when inaccurate), as they provide mentors with more information that illustrate a student's understanding in their project or in the design argument structure.

7 DISCUSSION AND FUTURE WORK

Our findings demonstrate that Polaris supported novice designers in identifying issues across key components of their design rationale - outcomes, obstacles, characteristics, and arguments, rather than surface-level issues that novices often raise. Literature describes how experts focus on evaluating their design argumentation to identify the critical issues which drive the next steps they take to improve their designs [5, 14, 26]. By scaffolding novices to focus their evaluation on the argumentation that justifies their designs, students may be better equipped to plan out impactful iterations that improve the fundamental logic behind their designs.

Polaris has the potential to improve how students identify critical issues and improve their designs because it embeds the structural knowledge and diagnostic practice that experts use to evaluate their design argumentation into scaffolds for novices. Our findings demonstrate that the design argument template and reflection prompts helped students visually and procedurally evaluate their design arguments for key structural issues that mentors often focus on: gaps, lack of depth, and misalignment. By embedding such knowledge and practice into scaffolds for students, students can recognize and improve the fundamental structural patterns that make or break a design argument, as experts do.

Not only did Polaris support students finding these structural issues, the tool also helped students raise what mentors considered critical issues in their design arguments, the majority of which were topically similar to issues mentors found themselves. These findings suggest that scaffolds like Polaris may help students increasingly self-direct critical issue identification in their design argumentation. However, tools like Polaris are not meant to replace mentors. As evidenced by our findings, domain-specific project issues are critical issues that may not be represented in generalizable argument scaffolds like Polaris. By helping students identify general issues in argument structure, mentors with necessary domain expertise may be able to better focus their efforts on training students with the knowledge needed to identify domain-specific issues and design effective solutions. Further, our findings show that by externalizing a student's current understanding of project specifics and general design argument structure, Polaris directly informs the coaching strategies that a mentor may enact, from planning out next steps when in agreement on a key issue, to coaching a student through a misunderstanding that is now made apparent. By scaffolding students to raise critical issues that mentors generally agree with, limited mentoring resources can be spent supporting novice designers in their many other learning needs, such as learning to plan out iterations that address critical issues.

Future work may consider how to apply the way Polaris focuses on structural issues of gaps, lack of depth, and misalignment to other layers of argumentation that experts often use to justify technical designs and design research interventions that go beyond the design argument itself [10]. Findings show that mentors were able to identify critical issues in areas of design-research argumentation that were beyond the scope of design arguments and Polaris, such as distinguishing between a user obstacle in the design argument, and a technical implementation obstacle in a system argument. Future systems can adopt Polaris features that highlight generalizable structural issues (i.e. gaps, lack of depth, and misalignment) in the technical argumentation (e.g. justifying a particular interface and system architecture design) or the research argumentation (e.g. positioning their conceptual approaches relative to existing research) often used in technical design and design research work.

However, scaffolding a novice to construct and assess structural issues across different forms of design-research argumentation introduces its own technical challenges, and may not be as simple as generating a series of multiple templates and corresponding prompts. As prior literature suggests, complex structural representations and extensive sets of heuristics can hamper the practical usability and usefulness of such tools for improving arguments and designs [23, 26]. Moving forward, future argument scaffolds can explore how to help novices visualize, navigate, and conceptually link the different layers of argument structure that justify their overall designs (e.g. design vs. technical vs. research argumentation). For instance, future diagnostic tools may visualize multiple layers of argumentation at once with one argument layer actively in focus and related argument layers just out of focus to reinforce the conceptual links across layers of argumentation (e.g. how a general design argument is instantiated into an interface design, and how the interaction requires novel system architectures and frameworks). As another example, future diagnostic tools could implement visual cues that can explicitly guide novices to focus on the most critical areas across their overall intervention. Such affordances may better scaffold novices to construct and evaluate conceptual and narrative links across the layers of their argumentation.

Future work can also consider how to integrate these diagnostic tools with existing pedagogy and coaching techniques in ways that allow for increasingly scalable and effective training environments. As our findings demonstrate, design mentors on their own may not be able to catch and coach students through the many different structural issues that come up in their argumentation, particularly given the real-world time constraints of most coaching interactions. While scaffolds like Polaris can effectively scaffold novices to catch more of these generalizable structural issues within design arguments, they are not designed to surface the domain-specific conceptual issues that domain experts would catch, as seen in our findings. When integrated in practice, diagnostic tools like Polaris can supplement the efforts of coaches for more comprehensive scaffolding for novice designers, transforming how mentors coach novices in design training environments. For example, mentors could deploy suites of generalizable argumentation structures across many students at once, scaling their coaching efforts to focus on conceptual issues that only they have the domain expertise to recognise. Similarly, using generalizable argumentation structures could streamline coaching discussions by creating a shared language of expertise between students and mentors, enabling mentors to point out structural concerns that students can quickly recognise and work on, even within a feedback session. Future work should further consider the socio-technical integration of tools, pedagogy, coaching, and self-practice in forming effective learning ecosystems for design.

8 CONCLUSION

In this paper, we introduce Polaris, a learner-centered diagnostic tool that scaffolds novices to visually and procedurally identify structural issues in their design argumentation. Despite existing approaches for training novice designers, students still struggle to identify reasons why their designs may fail – a fundamental expert practice that novices must learn to lead creative work. To overcome these challenges, Polaris embeds expert structural knowledge and diagnostic practice into templates and prompts that focus a novice's attention on three key structural issues in design argumentation: gaps, lack of depth, and misalignment. Our pilot study demonstrates that Polaris helped novice designers identify structural issues in their design arguments as experts do. Further, the mentors considered student issues to be critical to the design, and actionable for coaching students on next steps.

By designing diagnostic tools like Polaris, we can scaffold novices to closely examine and improve the argumentation that justifies their designs. To build such tools, we need to first understand the expert knowledge and practice that current design pedagogy and scaffolding fails to train, and embed these deep expert practices into the tools themselves so that novice designers can develop the richer set of skills they need to practice independently. Further, diagnostic tools like Polaris can be coupled with existing pedagogy, enabling mentors to focus their coaching efforts on conceptual issues in the domain, or discussing strategies and plans for mitigating critical issues in designs. By supplementing coaching with such scaffolds, we can make better use of and compliment limited mentor resources for more scalable and effective training environments for novice designers.

REFERENCES

[1] Robin S Adams, Jennifer Turns, and Cynthia J Atman. 2003. Educating effective engineering designers: The role of reflective practice. *Design studies* 24, 3 (2003), 275–294.

[2] Philip Bell and Elizabeth A Davis. 2000. Designing Mildred: Scaffolding students' reflection and argumentation using a cognitive software guide. In *Fourth international conference of the learning sciences*. Erlbaum Mahwah, NJ, 142–149.

[3] Philip Bell, Elizabeth A Davis, and Marcia C Linn. 1995. The knowledge integration environment: Theory and design. (1995).

[4] Philip Bell and Marcia C Linn. 2000. Scientific arguments as learning artifacts: Designing for learning from the web with KIE. *International journal of science education* 22, 8 (2000), 797–817.

[5] Spencer E. Carlson, Leesha V. Maliakal, Daniel G. Rees Lewis, Jamie Gorson, Elizabeth M Gerber, and Matthew Wayne Easterday. 2018. Defining and assessing risk analysis: The key to strategic iteration in real-world problem solving. *Proceedings of International Conference of the Learning Sciences, ICLS* 1, 2018-June (1 1 2018), 352–359.

[6] Spencer E Carlson, Daniel G Rees Lewis, Leesha V Maliakal, Elizabeth M Gerber, and Matthew W Easterday. 2020. The design risks framework: Understanding metacognition for iteration. *Design Studies* 70 (2020), 100961.

[7] David P Crismond. 2011. Scaffolding strategies for integrating engineering design and scientific inquiry in project-based learning environments. In *Fostering human development through engineering and technology education*. Brill Sense, 233–255.

[8] David P Crismond and Robin S Adams. 2012. The informed design teaching and learning matrix. *Journal of Engineering Education* 101, 4 (2012), 738.

[9] Kees Dorst and Nigel Cross. 2001. Creativity in the design process: co-evolution of problem–solution. *Design studies* 22, 5 (2001), 425–437.

[10] Matthew W. Easterday, Daniel G. Rees Lewis, and Elizabeth M. Gerber. 2016. The logic of the theoretical and practical products of design research. *Australasian Journal of Educational Technology* 32, 4 (2016), 125–144. https://www.researchgate.net/profile/Daniel_Rees_Lewis/publication/305218923_The_logic_of_the_theoretical_and_practical_products_of_design_research/links/57bc5d7808ae8a9fc4c4c135.pdf

[11] Peggy A Ertmer, Donald A Stepich, Cindy S York, Ann Stickman, Xuemei Wu, Stacey Zurek, and Yuksel Goktas. 2008. How instructional design experts use knowledge and experience to solve ill-structured problems. *Performance Improvement Quarterly* 21, 1 (2008), 17–42.

[12] Gerhard Fischer, Andreas C Lemke, Raymond McCall, and Anders I Morch. 1991. Making argumentation serve design. *Human–Computer Interaction* 6, 3-4 (1991), 393–419.

[13] Xun Ge and Susan M. Land. 2003. Scaffolding students' problem-solving processes in an ill-structured task using question prompts and peer interactions. *Educational Technology Research and Development* 51, 1 (01 Mar 2003), 21–38. https://doi.org/10.1007/BF02504515

[14] Xun Ge and Susan M. Land. 2004. A Conceptual Framework for Scaffolding Ill-Structured Problem-Solving Processes Using Question Prompts and Peer Interactions. *Educational Technology Research and Development* 52, 2 (2004), 5–22. http://www.jstor.org/stable/30221193

[15] Colin M Gray. 2018. Narrative Qualities of Design Argumentation. In *Educational Technology and Narrative*. Springer, 51–64.

[16] Colin M Gray, Colleen M Seifert, Seda Yilmaz, Shanna R Daly, and Richard Gonzalez. 2016. What is the content of "design thinking"? Design heuristics as conceptual repertoire. *International Journal of Engineering Education* 32 (2016).

[17] Per L Halstrøm and Per Galle. 2015. Design as co-evolution of problem, solution, and audience. *Artifact: Journal of Design Practice* 3, 4 (2015), 3–1.

[18] David H Jonassen. 2004. *Learning to solve problems: An instructional design guide*. Vol. 6. John Wiley & Sons.

[19] J. Knapp, J. Zeratsky, and B. Kowitz. 2016. *Sprint: How To Solve Big Problems and Test New Ideas in Just Five Days*. Transworld. https://books.google.com/books?id=X_blCgAAQBAJ

[20] Bryan Lawson and Kees Dorst. 2005. Acquiring design expertise. *Computational and Cognitive Models of Creative Design VI. Key Centre of Design Computing and Cognition, University of Sydney, Sydney* (2005), 213–229.

[21] A. Osterwalder. 2008. Business Model Canvas. (2008).

[22] A. Osterwalder and Y. Pigneur. 2010. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley. https://books.google.com/books?id=UzuTAwAAQBAJ

[23] Daniel G. Rees Lewis, Jamie Gorson, Leesha V. Maliakal, Spencer E. Carlson, Elizabeth M Gerber, Christopher K Riesbeck, and Matthew Wayne Easterday. 2018. Planning to iterate: Supporting iterative practices for real-world ill-structured problem-solving. *Proceedings of International Conference of the Learning Sciences, ICLS* 1, 2018-June (1 1 2018), 9–16.

[24] Daniel Rees Lewis, Emily Harburg, Elizabeth Gerber, and Matthew Easterday. 2015. Building Support Tools to Connect Novice Designers with Professional Coaches. ACM Press, 43–52. https://doi.org/10.1145/2757226.2757248

[25] Daniel G. Rees Lewis, Matthew W. Easterday, Emily Harburg, Elizabeth M. Gerber, and Christopher K. Riesbeck. 2018. Overcoming barriers between volunteer professionals advising project-based learning teams with regulation tools: Overcoming advising barriers with regulation tools. *British Journal of Educational Technology* (2018). https://doi.org/10.1111/bjet.12550

[26] Simon Buckingham Shum and Nick Hammond. 1994. Argumentation-based design rationale: what use at what cost? *International journal of human-computer studies* 40, 4 (1994), 603–652.

[27] Natalia Smirnov, Matthew W. Easterday, and Elizabeth M. Gerber. 2017. Infrastructuring Distributed Studio Networks: A Case Study and Design Principles. *Journal of the Learning Sciences* (2017), 1–52. https://doi.org/10.1080/10508406.2017.1409119

[28] Jan Van den Akker. 1999. Principles and methods of development research. In *Design approaches and tools in education and training*. Springer, 1–14.

[29] Haoqi Zhang, Matthew W Easterday, Elizabeth M Gerber, Daniel Rees Lewis, and Leesha Maliakal. 2017. Agile research studios: Orchestrating communities of practice to advance research training. In *2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*.