Refining the LITES Codebook Using a Discussion and Consensus-based Approach

Anu Singh, The Ohio State University

Anu Singh is a Postdoctoral Scholar in the Department of Engineering Education at The Ohio State University. She earned her Ph.D. in Engineering, with a specialization in Engineering Education Research from the University of Nebraska-Lincoln. Her research focuses on engineering students' critical thinking skills, argumentation, and epistemological beliefs, with the goal of supporting their development of competencies that extend beyond technical expertise.

Dr. Elif Miskioglu, Bucknell University

Dr. Elif Miskioglu is a mid-career engineering education scholar and educator. She holds a B.S. ˘ in Chemical Engineering (with Genetics minor) from Iowa State University, and an M.S. and Ph.D. in Chemical Engineering from Ohio State University. Her early Ph.D. work focused on the development of bacterial biosensors capable of screening pesticides for specifically targeting the malaria vector mosquito, Anopheles gambiae. As a result, her diverse background also includes experience in infectious disease and epidemiology, providing crucial exposure to the broader context of engineering problems and their subsequent solutions. These diverse experiences and a growing passion for improving engineering education prompted Dr. Miskioglu to change her career path and become a scholar of engineering education. ˘ As an educator, she is committed to challenging her students to uncover new perspectives and dig deeper into the context of the societal problems engineering is intended to solve. As a scholar, she seeks to not only contribute original theoretical research to the field, but work to bridge the theory-to-practice gap in engineering education by serving as an ambassador for empirically driven educational practices.

Dr. Kaela M Martin, Embry-Riddle Aeronautical University - Prescott

Kaela Martin is an Associate Professor of Aerospace Engineering at Embry-Riddle Aeronautical University, Prescott Campus. She graduated from Purdue University with a PhD in Aeronautical and Astronautical Engineering. Her research interests in engineering education include developing classroom interventions that improve student learning, designing experiences to further the development of students from novices to experts, and creating engaging classroom experiences.

Dr. Adam R Carberry, The Ohio State University

Dr. Adam R. Carberry is Professor and Chair in the Department of Engineering Education at The Ohio State University (OSU). He earned a B.S. in Materials Science Engineering from Alfred University, and received his M.S. and Ph.D., both from Tufts University, in Chemistry and Engineering Education respectively. He recently joined OSU after having served as an Associate Professor in The Polytechnic School within Arizona State University's Fulton Schools of Engineering (FSE) where he was the Graduate Program Chair for the Engineering Education Systems & Design (EESD) Ph.D. Program. He is currently a Deputy Editor for the Journal of Engineering Education and co-maintains the Engineering Education Community Resource wiki. Additional career highlights include serving as Chair of the Research in Engineering Education Network (REEN), visiting École Nationale Supérieure des Mines in Rabat, Morocco as a Fulbright Specialist, receiving an FSE Top 5% Teaching Award, receiving an ASEE Educational Research and Methods Division Apprentice Faculty Award, receiving a Frontiers in Education New Faculty Award, and being named an ASEE Fellow.

Refining the LITES Codebook Using a Discussion and Consensus-based Approach

Introduction

This methods full paper is grounded in our prior work on engineering intuition. Intuition, which is also colloquially referred to as a gut feeling or a sixth sense, has been conceptualized by researchers as a less conscious [1] or unconscious cognitive process [2]. Resulting intuitive decisions are involuntary, automatic, and almost effortless [3]. Intuition use in the literature is sometimes described in conflicting ways. Some authors emphasize that intuition involves both cognitive and emotional elements [4] and describe intuition as an "unconscious judgment based on feelings" [5], whereas others define intuition as an "information-based and lawful process" [6], which draws on stored knowledge [7]. These variations in the description of intuition highlight its complexity and indicate different implications for researchers [8], which enables further investigation of intuition across contexts.

Across disciplines (e.g., business and medicine) and STEM fields [9], researchers have explored the role of intuition and decision making in managerial decision-making [2], clinical decision-making [10], and strategic thinking [11]. The role of intuition in engineering is comparatively less examined and has been explored in the context of solving complex, ill-structured problems [12] that often require decision-making to solve [13]. Existing studies provide limited guidance or tools for identifying intuitive processes despite these scholarly works highlighting the role of intuition in the workplace.

One of the reasons for the limited availability of tools is that intuition is difficult to capture. Individuals often struggle to explain their engagement with intuition in words [14], [15]. Although theoretical models such as recognition-primed decision making [16] and dual-process theory [17] describe the cognitive process involved in the use of intuition, they do not provide indicators or tools to describe the use of intuition during the decision-making or problem-solving process. The Leveraging Intuition Towards Engineering Solutions (LITES) framework [12] and complementary LITES codebook, provide a data analysis tool for exploring the use of intuition in solving ill-structured problems.

The LITES codebook originated from a qualitative study of engineering practitioners guided by grounded theory and phenomenography that aimed to define engineering intuition [12]. The codebook development and subsequent use focused on a consensus approach to support quality [18]. The definitions and clarity of the codebook were cross-checked by members of the team who were not involved in the codebook development directly, but were involved in the data collection, as another measure to assure quality.

Recent work surprisingly revealed that coders with no prior experience with the research on intuition interpreted and applied the codes from the LITES codebook differently from the experienced team members. These coding discrepancies in the use of the LITES codebook suggested that the codebook was not user-friendly for researchers outside the original research team. This result points to a limitation of and need for a greater clarity within the codebook, while also raising questions about consistency in the use of the codebook, i.e., inter-coder reliability (ICR) [19].

The need for a refined codebook that supports greater consistency in code application shaped the purpose of this work, which was completed in two phases. The first phase (P1) involved refining

the LITES codebook to minimize misinterpretation and improve the accuracy and consistency of code application. The second phase (P2) investigated ICR, the quantitative level of agreement between coders when applying codes to a text segment [20], with the refined codebook using Cohen's Kappa. Examining ICR across the original (v1) and refined (v2) LITES codebook aimed to assess codebook improvement and subsequently support which codebook to use for future studies of engineering intuition. The following complementary research questions (RQ) guided our study:

RQ 1. How did discrepancies in the application of the LITES codebook emerge, and what was the nature of these discrepancies?

RQ 2. How do the ICR values compare when coding with the original versus the refined LITES codebook?

While the LITES codebook is intended to be used in consensus-based coding, improving ICR across the v1 and v2 codebooks would support wider usability of the LITES codebook, amplifying a useful tool in the study of engineering intuition.

Methods

The two phases of the study involved different sets of participants, data collection, and data analysis processes. P1 focused on refining the code definition of the LITES codebook (RQ1), while P2 investigated the reliability of the v2 codebook (RQ2). The following sections describe the methods used in each phase.

Phase 1 (P1) Data Collection and Analysis

Data Collection: This study used transcripts obtained from three interviews with engineering experts. Detailed information on the interview protocol has been published in our previous work [21]. The transcripts were generated using Zoom's transcription feature, cleaned, deidentified, and checked for accuracy.

Data Analysis: Each of the three transcripts was coded independently by six coders using the v1 codebook. Among the six coders, two were experienced coders (E1 and E2) who were involved in the development of the v1 codebook, while the remaining four were new coders with no prior experience with the v1 codebook. Among the four new coders, one coder had considerable previous experience with qualitative coding, one had limited experience with deductive coding, and the remaining two coders had no experience with qualitative coding.

The new coders were not provided with codebook-specific training to better mimic the experience of an external scholar using the published codebook. The new coders compared and discussed their coded transcripts with the experienced coders. The new coders were asked to explain why they used a specific code for a given statement and their understanding of the codes. During this discussion, the experienced coders documented discrepancies that emerged from the new coders' interpretation of the codes. Later, both the experienced coders and the new coder with considerable coding experience discussed and documented discrepancy points for each code in the codebook and refined the code definition through consensus. Discrepancies emerged across all nine primary codes in the v1 codebook (Constraints, Individual Mindset, Experience, Holistic Prediction, Data-Based Assessment, Sensibility Check, Physiological, Individual Conditions, and Outcome); sub-codes were not considered. Our prior work provides examples of

how coding discrepancies emerged and how they were addressed for selected codes [22]. ICR was not calculated in prior work since the v1 codebook was developed and used through consensus coding. Consensus discussions were also used during P1 to identify code discrepancies and refine code definitions, so ICR was not calculated for P1.

Phase 2 (P2) Data Collection and Analysis

The reliability of the v2 codebook was examined by adding two external coders (N1 and N2) with no prior experience with the LITES codebook. N1 and N2 were selected using purposeful sampling and had previous experience in qualitative deductive coding. N1's primary background is in engineering education research, while N2's is in technical communication with some prior collaborative work in engineering education. N1 and N2 were not provided with formal training on the codebook use. The intention was to allow N1 and N2 to directly implement the codebook definitions rather than being influenced by the interpretation of codes by the research team.

Data Collection: One of the three interview transcripts in P1 was chosen for P2. The transcript was truncated by removing text not related to solving the underlying problem, such as introductions and discussions of other problems. To finalize the truncated version of the transcript, the research team members initially prepared their own shortened version of the transcript before coming to a consensus on the final truncated version to give to N1 and N2. The research team ensured that the final truncated version provided sufficient opportunities to apply codes from both the v1 and v2 versions of the codebook. Line numbers were assigned to the truncated transcripts for analysis. N1 and N2 coded only the primary codes. N1 and N2 received the truncated transcript, detailed coding instructions, and codebooks through email.

The data collection process was completed in two rounds. To ensure independent evaluation of both codebook versions by the research team, N1 and N2 were provided with one version of the codebook at a time. N1 and N2 applied the v1 codebook during the first round. A month after the completion of round one, N1 and N2 received the v2 codebook. A month was chosen so that N1 and N2 would not explicitly remember the definitions of codes in v1 or the transcript's content. In the second round, N1 and N2 coded the same transcript as in round one, but now with the v2 codebook. The expert coders (same E1 and E2 as in P1) coded the truncated transcript only once using only the v2 codebook.

Data Analysis: The coded transcript was analyzed to calculate the ICR of the revised codebook. Coding of the transcript by N1 and N2 was compared with the expert-coded version completed by E1 and E2. Each transcript line was entered into a spreadsheet to conduct line-wise analysis. A line was included in the reliability calculation if at least one of the four coders assigned a code to the line. Lines with no codes assigned by any of the coders were excluded. N1 and N2 applied codes differently across the two coding rounds, so the set of lines included in the analysis for v1 and v2 was different. Later lines were binary coded for the presence or absence of each code in the line to examine intercoder reliability for the codebook using Cohen's Kappa coefficient [23]. Cohen's Kappa coefficient was calculated separately for both versions of the codebook across different coder pairs. The kappa values calculated using SPSS (v. 27) were later compared to evaluate the difference in reliability of code use across the two versions of the codebook.

Code occurrences were unevenly distributed across the coders, indicating the presence of the prevalence effect [24]. Under such conditions, kappa values are often low and indicate poor reliability [25]. The frequency count for each code from E1 and E2 was used as the reference

point for determining whether to calculate kappa values for those codes. A code was considered sufficiently frequent if both E1 and E2 applied that code at least 11 times in the transcript. Three codes of the total nine codes (Holistic Prediction, Sensibility Check, and Physiological) fell below the threshold (Table 1). ICR values for these codes were not calculated.

Table 1: Frequency of each Code used by E1E2 while coding the truncated transcript

Codes	Frequency of code use	
	E1	**E2**
Data-Based Assessment (DBA)	45	52
Individual Mindset (IM)	44	33
Constraint (C)	29	41
Experience (E)	29	32
Outcome (O)	26	44
Individual Conditions (IC)	15	18
Sensibility Check (S)	15	10
Holistic Prediction (H)	10	3
Physiological (P)	2	1

Results & Discussion

Phase 1 (P1): Emergent Code Discrepancies in v1 Codebook

P1 resulted in the v2 codebook (Table 2; changes in italics). The discrepancies identified during the consensus codebook revision process revealed recurring patterns across multiple codes. The identified discrepancies were grouped into four different themes to capture commonalities.

Theme 1: Multiple interpretations of key terms used in code definitions

Discussion between the new[1] and experienced coders indicated differences in how certain terms in the code definition of the original codebook were interpreted differently by both groups. These discrepancies were observed in the definitions of the Holistic Prediction (H), Sensibility Check (SC), Individual Conditions (IC), and Outcome (O) codes. For example, when applying the O code, new and experienced coders assigned different codes to the following excerpt, where the participant discussed the resolution status of an intermediate problem that occurred while they were working towards solving the primary problem described.

> So, that was another issue <u>that we worked through</u>.

The new coders did not code the underlined segment of the above excerpt as Outcome, whereas the experienced coders did. During the discussion, the new coders explained that they considered the excerpt as an intermediate step in the complete problem-solving process, rather than a final solution to the primary problem. To address this discrepancy and improve the clarity of the code definition, the experienced coders incorporated the word “intermediate” into the original definition to read as “any final or *intermediate* results.”

[1] Note that "new" coder in P1 refers to four coders with different levels of qualitative experience coding and are different from the N1 and N2 coders of P2.

Similar discrepancies emerged in coding for H, SC, and IC. The word *action* embedded in the description for the H code was identified as the source of the discrepancy. The new coders interpreted *action* in a narrow and literal sense and considered *action* as specific steps carried out to address a problem. Experienced coders interpretation of the word, which was different from the new coders included a plan, proposed idea, or proposed solution. The discrepancy was addressed by adding *solution* along with *action.* The SC code discrepancy involved the terms *solution*, *conscious*, and *subconscious*. The new coders interpreted *solution* as a proposed plan or idea that is under discussion and not yet implemented, which was addressed by including *actions* in the code definition. The new coders' limited understanding of applying the criterion of *conscious* or *subconscious* review of the *solution/action* resulted in discrepancies, which was addressed with the addition of a note on the SC code application to indicate the possibility of co-coding the code with H, Data-Based Assessment, Physiological, and Experience codes. For the IC code, a discrepancy emerged with different interpretations of *ability* by the new and experienced coders. The new coders interpreted *ability* as whether the participant is able or not to solve the problem because of certain factors, whereas the experienced coders interpreted the word to also capture participants' engagement in different strategies or processes to solve a problem. This discrepancy was addressed by replacing *ability* with *approach*.

Theme 2: Lack of clarity on whose perspective a code definition applies to (participants or others)

Lack of clarity on whose perspective is considered emerged during the application of Constraint (C), Individual Mindset (IM), Physiological (P), and Individual Conditions (IC). Discussion between the new and experienced coders addressed whether these codes should be applied only to the participant's perspective or could also include perspectives, reactions, or conditions associated with others described by the participant.

For example, regarding the use of the P code in the following excerpt, where the participant described team members' reaction (panic) to a particular problem:

> And so when that happened, they sort of panic, and they did exactly the wrong thing

The new coders coded the above excerpt as Physiological, but the experienced coders did not assign the same code. During the discussion, the experienced coders identified that the original definition of the code does not explicitly mention that the code is to be used only to capture the participant's reaction. To eliminate this discrepancy, the experienced coders included *participant* in the refined definition of the code (Table 2). Similarly, for the C and IM code, *participant* was added to the refined code definitions to ensure clarity of the code definition for the new coders and for consistent application. For the IC code, *individual* in the original definition was confusing for the new coders. For example, the new coders assumed that *individual* referred to both participant and team member, which was different from the experienced coders' interpretation. This discrepancy was addressed with the replacement of *individual* with *participant* to make it explicit that any condition that influences only the participant's problem-solving process would be coded as IC.

Theme 3: Challenges in differentiating the Individual Conditions code from the Constraint Code

This theme captures instances indicating the need for a clear distinction between the use of the Constraint (C) code from the Individual Conditions (IC) code. Discussion between the new and

experienced coders revealed the new coders' struggle in distinguishing the application of the IC code from the C code. For example, in the following excerpt, the participants said:

> So, I understand that. But the safety is something we cannot negotiate.

The excerpt indicates the participant's priority for considering the safety factor in the related problem-solving process. The new coders coded the participant's excerpt as C, and the experienced coders coded it as IC. During the discussion, the new coders explained that they interpreted *safety* as a limiting factor in the problem-solving process, whereas the experienced coders explained that participants' emphasis on the safety factor was due to their own personal experience based on the transcript context. The C code applies solely to the problem itself. Any problem solver would experience it, whereas IC varies from person to person. To address this discrepancy, the experienced coders added a detailed note on the C and IC code application (Table 2). Both notes highlighted differentiating elements related to the application of C and IC codes, including phrases such as *irrespective of problem solving* and guiding questions to clarify the difference between the C and IC codes.

Theme 4: Limited operational guidance for code application

The final theme involves the cases where the original definition of the code was clear and easily understood by the new coders, but the descriptors or code application notes needed changes for Data-Based Assessment (DBA), Individual Mindset (IM), and Experience (E) code.

For example, in the following excerpt pertaining to DBA, the participant discussed a rule of using 32.2 in unit conversion for certain calculations:

> So, you know, the rule of thumb was, you just start multiplying and dividing by 32.2 until you get to the right order of magnitude, right, you know.

The experienced coders used DBA for this excerpt, while the new coders did not. During the discussion, the new coders noted that the participant's discussion seemed like a standard process for doing things in a discipline, which is an expected process and not a specific approach the participant used to solve the problem. Additionally, the new coders mentioned that the DBA code definition and description did not provide any information on whether or not to consider standard operating procedures as DBA. Considering the new coders' explanation, the experienced coders included "*following a standard procedure or stepwise process for the discipline, no research or analysis, no reasoning or justification*" in the descriptor for the DBA code (Table 2), while leaving the original definition unchanged.

For the IM and E codes, code applications notes were added. The new coders often struggled with the IM code when participants referred to standard disciplinary knowledge. For example, in a systems engineering context, transcript excerpts describing structural design trade-offs, well-established safety risks, and known performance tests are part of the standard disciplinary knowledge. The new coders coded this standard knowledge as IM, which was not correct. To address this discrepancy, the experienced coders added an application note to the code, "*does not apply to perspectives that are general knowledge within the field (e.g., known safety risks described as dangerous).*"

Table 2: Refined version (v2) of the LITES codebook (italicized text shows change to the codebook)

Codes	Definition	Descriptors
Constraints (C)	Anything described *explicitly or implicitly by the participant* that limited or restricted problem solving or that amplified existing constraints	Time, resources, money, outside influence, ambiguity, risk, politics, specifications, goal, contract
Notes on C Application	*Applies to anything that limits the problem itself irrespective of who is solving the problem. Distinct from Individual Conditions which are unique to the problem solver (but may similarly "constrain" problem-solving). Key question to differentiate between Constraints and Individual Conditions: If another engineer were in the participant's position, would they experience the same limitation/restriction? If yes, Constraints; if no, Individual Conditions*	
Individual Mindset (IM)	Description of a perspective or attitude *participant had/has about anything including of situation*, themselves or others	Attitude, Passion, opinion, as a women, as a men, I see engineering as…
Notes on IM Application	*Does not apply to perspectives that are general knowledge within the field (e.g. known safety risks described as dangerous).*	
Experience (E)	Knowledge acquired through observation or doing relevant work (outside of academic education) or knowledge missing from a lack of such experience	Relevant to job/work/career path, applicable encounters, life-lessons, never seen before
Notes on E Application	*Includes participant's descriptions of leveraging other's experience (or lack thereof).*	
Holistic Prediction(H)	Anticipation and/or assessment (prediction) of the effects of potential actions *or solutions* often through use of a "big picture" perspective	Holistic perspective, systems view, big picture thinking, if we do X then Y, impact
Data-Based Assessment (DBA)	Evidence- or knowledge-focused approaches to problem solving or a lack thereof	Evidence-based, justification, analysis, gathering information, *following a standard procedure or stepwise process for the discipline, no research or analysis, no reasoning or justification*
Sensibility Check (S)	Conscious or subconscious review of potential solutions*/actions* to assess whether the solution*/action* is reasonable in the context	In the ballpark, what I should expect, does that answer make sense, make sure
Notes on S Application	*Often (but not always) co-coded with another Problem-Solving Approach (Holistic Prediction, Data-Based Assessment, Physiological, Experience). Occurs after identification of a proposed solution/action.*	
Physiological (P)	*Participant's* indescribable, subconscious (physical or emotional) reactions that suggest a possible solution/action or that occur in response to a proposed or actual decision/action	Gut-feeling, doesn't feel right, I don't know, I just do, internal response, something not seeming right despite data, initial reaction, instinct
Individual Conditions (IC)	Any condition described *explicitly or implicitly* by the *participant* that uniquely affected their problem-solving *approach*	Fear, embarrassment, hesitation, stress, confidence, trust, liking, motivation, comfortable, empowerment, tech as a tool
Notes on IC Application	*Applies only to conditions affecting the participant themselves, not others (such as team members). See Constraints for note on key questions to differentiate Individual Conditions and Constraints.*	
Outcome (O)	Any final or *intermediate* results from problem-solving actions or *inactions*	Outcome, result, effect

For the E code, the new coders did not apply the code when participants discussed using other people's experiences (e.g., colleagues, seniors, team members). The experienced coders addressed this discrepancy with a note on the application of the E code to the v2 codebook specifying that the E code *"includes participant's descriptions of leveraging others' experience (or lack thereof)."*

P1 resulted in the v2 codebook (Table 2). P2 leveraged the new codebook to expand this study by investigating how consistently N1 and N2 applied the refined codes to the same transcript.

Phase 2 (P2): Improved ICR with v2 Codebook

An overall improvement in ICR was observed with the use of the v2 codebook for the external coder pairs (Table 3). According to Landis and Koch's [26] benchmark, Cohen's Kappa value reported in this study can be interpreted as *slight agreement* for kappa values between 0.0-0.20, *fair* agreement for 0.21- 0.40, *moderate* agreement for 0.41-0.60, *substantial* agreement for 0.61-0.80, *perfect* agreement for 0.81-1.0, and values below zero indicate *poor* agreement. It is important to note here that our focus is not on the ICR values obtained with v1 and v2 codebooks, but rather on the differences between these values. The agreement between the external coders (represented as N1N2) increased notably from *slight* to *moderate* ($\kappa = 0.158$ with v1 and $\kappa = 0.460$ with v2), showing that their application of codes was more consistent while using v2. The increased kappa value (agreement) for N1N2 with the v2 codebook is also closer to the expert coder pair (E1E2).

For N1N2, the high kappa value with the use of v2 suggests that the refinements to the codebook improved the clarity of the definitions and supported more consistent use of the codes by N1 and N2, hence improving the reliability (reproducibility) of the codebook [19]. Although E1 and E2 coded the transcript only once, the results (Tables 3 and 4) show two different kappa values for v1 and v2 due to differences in the number of lines that have applied codes across all four coders. These results reflect that methodological decisions on average slightly influenced the ICR values, not that agreement improved between E1 and E2.

Table 3: Cohen's Kappa value for each coder pair across two versions of the codebook

Codebook Version	N1N2	N1E1	N1E2	N2E1	N2E2	E1E2
Original (v1)	0.158	0.214	0.299	0.261	0.372	0.512
Refined (v2)	0.460	0.354	0.442	0.330	0.321	0.516

Across the six individual codes that met the minimum count threshold (Table 1), Cohen's Kappa value improved for N1N2, except for the IM and O codes (Table 4). For each code, the change in agreement level across v1 and v2 was assessed using Landis and Koch's criteria [26]. Improvement was considered when the N1N2 agreement shifted to a higher category (e.g., from *slight* to *moderate* or from *fair* to *moderate*).

When the agreement level remained unchanged, improvement was evaluated by examining whether N1N2 agreement moved closer to E1E2 agreement. For N1N2, the change in level of agreement was noteworthy for both the C and IC codes (from *slight* to *moderate*), suggesting that the revised definitions and the addition of application notes with C and IC in v2 made these codes easier for N1 and N2 to interpret and apply more consistently. For the DBA code, agreement between N1 and N2 improved from *slight* to *fair* with the use of v2, but remained lower than E1E2 pair. This result suggests that further improvement in agreement is more likely

to come from coder training. The level of agreement for the code E remained *moderate* with the use of v2, but the revised definition reduced the difference between N1N2 and E1E2 agreement levels, suggesting more consistent interpretation across coders with different levels of experience.

Table 4: Cohen's Kappa value for six codes across both versions of the LITES codebook

Code	Codebook Version	N1N2	N1E1	N1E2	N2E1	N2E2	E1E2
C	v1	0.158	0.214	0.299	0.261	0.372	0.512
	v2	0.460*	0.354	0.442	0.330	0.321	0.516
IM	v1	0	0.027	0.013	0	0	0.432
	v2	-0.008	0.035	-0.010	-0.030	0.027	0.433
E	v1	0.431	0.355	0.524	0.242	0.564	0.614
	v2	0.545†	0.389	0.422	0.513	0.650	0.629
DBA	v1	0.023	0.283	0.232	0.366	0.606	0.599
	v2	0.383*	0.527	0.648	0.352	0.407	0.682
IC	v1	-0.009	0.218	0.346	-0.01	-0.01	0.502
	v2	0.558*	0.292	0.247	0.530	0.387	0.503
O	v1	0.194	0.022	0.283	0.222	0.201	0.527
	v2	0.176	0.447	0.543	0.214	0.212	0.551

* Change in agreement level. † No change in agreement level but moved closer to the expert.

In contrast, the level of agreement for the IM and O code reduced with the use of the v2. For the IM code, the kappa value was less than zero, which means *poor* agreement [26] between N1 and N2. Further investigation revealed that N1 and N2 identified a low number of instances of the IM code in the transcript for both the v1 (N1=10; N2=0) and the v2 (N1=1; N2=3) codebooks, compared to E1 and E2 (E1=44; E2=33). These numbers indicate that N2 identified more instances of the code with v2, but those instances were not the same as those coded by the other coders.

The mismatch in how the code was applied by N1 and N2 with v2 led to more chances of disagreement between the coders, which reduced their overall agreement and lowered the kappa value [27]. For the O code, agreement between N1 and N2 did not improve with the use of v2. For the IM and O codes, the difference between N1N2 and E1E2 agreement levels remained large, suggesting that IM and O codes are harder for new coders to apply consistently without guided support from experienced coders.

Although v2 improved the agreement levels for most codes between N1 and N2 coders, a noticeable difference in kappa values between N1N2 and E1E2 remained. For example, the agreement for N1N2 for the DBA code with v2 (0.383) was lower than for E1E2 (0.682). This difference in kappa values may be due to differences in the level of familiarity with the data between the two coder groups, which influenced their code application and agreement levels. E1 and E2 were more familiar with the transcripts because of their involvement in data collection (interviewing participants), which provided them with more context to support their interpretations of the data. Also, E1 and E2's engagement in the development and refinement of the codebook through consensus discussion may have resulted in more consistent application of the codes. For the E and IC code, N1N2 agreement were closer to those for E1E2, suggesting

more alignment in code application between the new and experienced coders. This reduced difference between N1N2 and E1E2 suggests that the revised definitions supported consistent interpretation of codes by N1 and N2, despite their different disciplinary backgrounds.

Implications

With an improved inter-coder reliability, the v2 codebook results in a more consistent application of the codebook to capture the use of intuition in the engineering problem-solving process. This work provides methodological advancement by outlining a discussion and consensus-based approach that involves the engagement of new coders in discussions about code application. These discussions revealed the new coders' understanding of code definitions, implementation challenges, and provided insights into where modifications could be made. The discussion and consensus-based approach resolves discrepancies in code interpretation and improves consistency in code use by the coders beyond the research team. The approach can be applied to other areas where researchers aim to develop a codebook for broader applications.

Conclusion

This two-phase study was conducted to make the LITES codebook more robust for researchers within and beyond the original team. P1 revealed discrepancies associated with each code in the codebook through detailed discussion with coders who had no prior experience using the codebook. These discrepancies were addressed through targeted revisions of definitions and descriptions in the addition of notes clarifying code application, using a discussion and consensus-based approach. P2 investigated the intercoder reliability of both versions of the codebook with two new external coders. The v2 codebook resulted in improved ICR values overall. The v2 codebook showed greater consistency in code use among N1 and N2, indicating more stable and aligned code use. These findings show that a discussion and consensus-based approach grounded in new coders' feedback (with no experience with the LITES codebook) can meaningfully improve a codebook. The study provides a revised LITES codebook as a more robust tool for studying engineering intuition, while demonstrating a useful approach for refining codebooks in general.

Acknowledgement

This work is made possible by the National Science Foundation (Grant Nos. 2325523, 2325525, and 2434698).

References

[1] T. Hardman, "Understanding creative intuition," *J. Creativity*, vol. 31, 2021. https://doi.org/10.1016/j.yjoc.2021.100006

[2] E. Dane and M. G. Pratt, "Exploring intuition and its role in managerial decision making," *Acad. Manage. Rev.*, vol. 32, no. 1, pp. 33-54, 2007. https://doi.org/10.5465/amr.2007.23463682

[3] D. Kahneman and G. Klein, "Conditions for intuitive expertise: a failure to disagree," *American Psychologist*, vol. 64, no. 6, 2009. https://doi.org/10.1037/a0016755

[4] M. Sinclair and N. M. Ashkanasy, “Intuition: Myth or a decision-making tool?,” *Management Learning*, vol. 36, no. 3, pp. 353-370, 2005. https://doi.org/10.1177/1350507605055351

[5] Y. Wang, S. Highhouse, C. J. Lake, N. L. Petersen, and T. B. Rada, “Meta-analytic investigations of the relation between intuition and analysis,” *J. of Behavioral Decision Making*, vol. 30, no. 1, pp.15-25, 2017. https://doi.org/10.1002/bdm.1903

[6] J. A. Effken, “Informational basis for expert intuition,” *Journal of Advanced Nursing*, vol. 34, no. 2, pp. 246-255, 2001. https://doi.org/10.1046/j.1365-2648.2001.01751.x

[7] G. A. Klein, *Intuition at Work: Why Developing Your Gut Instincts Will Make You Better at What You Do*, Doubleday, 2003.

[8] N. Khatri and H. A. Ng, “The role of intuition in strategic decision making,” *Human Relations*, vol. 53, no.1, pp. 57-86, 2000. https://doi.org/10.1177/0018726700531004

[9] E. E. Miskioğlu, K. Martin, and A. Carberry, “The critical role of intuition in problem solving,” in *Ways of Thinking in STEM-based Problem Solving*, 1st ed., London: Routledge, 2024, pp. 62–76. doi: 10.4324/9781003404989-5.

[10] L. Rew, “Acknowledging intuition in clinical decision making,” *J. of Holistic Nursing*, vol. 18, no. 2, pp. 94-108, 2000. https://doi.org/10.1177/089801010001800202

[11] G. Henden, *Intuition and its role in strategic thinking*. Handelshøyskolen BI, 2004

[12] E. E. Miskioğlu, C. Aaron, C. Bolton, K. Martin, M. Roth, S. M. Kavale, and A. Carberry, “Situating intuition in engineering practice,” *J. Eng. Educ.*, vol. 112, no. 2, pp. 418–444, Apr. 2023. https://doi.org/10.1002/jee.20521

[13] N. S. Nisrina, M. Melinda, Y. Anwar, and D. Kurniawan, “Interplay of problem-solving and decision-making in STEM education with water pollution problem,” *International Journal of Research in Education and Science*, vol. 11, no. 2, pp. 396-408, 2025.

[14] G. P. Hodgkinson, J. Langan-Fox, and E. Sadler-Smith, “Intuition: A fundamental bridging construct in the behavioural sciences,” *British Journal of Psychology*, vol. 99, no. 1, pp. 1-27, 2008. https://doi.org/10.1348/000712607X216666

[15] S. Teerikangas and L.Välikangas, “Exploring the dynamic of evoking intuition,” in *Handbook of Research Methods on Intuition,* London: Edward Elgar, 2014, pp. 72-87.

[16] G. A. Klein, “A recognition-primed decision (RPD) model of rapid decision making,” in *Decision Making in Action: Models and Methods*, Ablex Publishing, 1993, pp. 138-147.

[17] J. S. B. Evans, “Dual-processing accounts of reasoning, judgment, and social cognition,” *Annu. Rev. Psychol.*, vol. 59, no. 1, pp. 255-278, 2008. https://doi.org/10.1146/annurev.psych.59.103006.093629

[18] J. Walther, N. W. Sochacka, and N. N. Kellam, "Quality in interpretive engineering education research: Reflections on an example study. *Journal of Engineering Education*, vol. 102, no. 4, pp. 626–659, 2013. https://doi.org/10.1002/jee.20029

[19] J. L. Campbell, C. Quincy, J. Osserman, and O. K. Pedersen, "Coding in-depth semistructured interviews: Problems of unitization and intercoder reliability and agreement," *Sociological Methods & Research*, vol. 42, no. 3, pp. 294-320, 2013. https://doi.org/10.1177/0049124113500475

[20] K. S. Kurasaki, "Intercoder reliability for validating conclusions drawn from open-ended interview data," *Field methods*, vol. 12, no. 3, pp. 179-194, 2000. https://doi.org/10.1177/1525822X0001200301

[21] N. Ugenti, J. Busato, E. Miskioglu, and K. Martin, "Using cognitive task analysis to observe the use of intuition in engineering problem solving," in *ASEE Annual Conferences* 2024, *Portland, Oregon, June 23-26*, doi: 10.18260/1-2--48229

[22] A. Singh, E. E. Miskioğlu, K. Martin, and A. Carberry, "WIP: Improving Codebook Application for Studying Engineering Intuition," in *IEEE Frontiers in Education Conference, 2025, Nashville, Tennessee, November 2-5.*

[23] K. Gwet, *Handbook of Inter-rater Reliability.* Gaithersburg, MD: STATAXIS Publishing Company, 2001.

[24] K. A. Hallgren, "Computing inter-rater reliability for observational data: An overview and tutorial," *Tutorials in Quantitative Methods for Psychology*, vol. 8, no. 1, 2012.

[25] T. Byrt, J. Bishop, and J. B. Carlin, "Bias, prevalence and kappa," *Journal of Clinical Epidemiology*, vol. 46, no. 5, pp. 423-429, 1993. https://doi.org/10.1016/0895-4356(93)90018-V

[26] J. R. Landis and G. G. Koch, "An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers," *Biometrics*, pp. 363-374, 1977. https://doi.org/10.2307/2529786

[27] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: The kappa statistic," *Family Medicine*, vol. 35, no. 5, pp. 360-363, 2005.