A Variant Q-Sorting Methodology for Building Diagnostic Trees

Sahar Sabbaghan ©, Cecil Eng Huang Chua, and Lesley A. Gardner

Abstract—Diagnostic theories are fundamental to information system (IS) practice and are represented as trees. While there are approaches for validating diagnostic trees, these validate the overall performance of the tree rather than identifying ways incorrect diagnoses can occur. It is important to fully validate diagnostic trees because even if the tree gives the correct decision “most of the time,” it is possible for incorrect decisions traveling down little-used branches of the tree to result in catastrophic decisions. In this article, we describe the process of using a variant of q-sorting to validate diagnostic trees. In this methodology, diagnostic trees that independent experts develop are transformed into a quantitative form, and that quantitative form is tested to determine the inter-rater reliability of the individual branches in the tree. The trees are then successively transformed to incrementally test if they branch in the same way. The results help researchers not only identify quality items for use in a diagnostic tree but also facilitate diagnoses of problems with those items and facilitate the reconciliation of discrepant trees by experts. The methodology validates not only the whole tree but also its subparts.

Index Terms—Diagnostic theories, diagnostic-tree, inter-rater reliability, q-sorting, tree.

I. INTRODUCTION

DIAGNOSTIC theories are fundamental to information systems (IS) practice. A diagnostic theory is used to identify why a particular situation occurs. Diagnostic theories have wide applicability. For example, MYCIN [1] and other expert systems [2], [3] often diagnose errors using diagnostic theories. Beyond expert systems, diagnostic theories are useful for identifying the root cause of a phenomenon. As an example, a diagnostic theory was used to improve a production line by highlighting possible areas of inefficiency [4]. Finally, diagnostic theories are integral to follow-up customer satisfaction surveys. When customers indicate they are dissatisfied, we may send a second survey to identify the source of dissatisfaction. This second survey often has an embedded diagnostic theory. Examples include surveys that identify the reasons for poor online purchase experiences [5] or surveys that unveil the relative ranks of the different web strategies to build trusting beliefs [6].

Many diagnostic theories are optimally represented as a tree, i.e., a diagnostic tree (DT), where intermediate nodes represent information important for the decision-making process and leaf nodes represent the optimal decision. The decision-making process begins with a generic problem (e.g., users do not find the technology useful) and as additional information is obtained, the system navigates through the branches of the tree corresponding to the obtained information until sufficient information is available to make a decision (e.g., why users do not find the technology useful). All the above-highlighted examples are of DTs.

The validation of DTs has been little investigated. Existing methodological approaches to validation (e.g., the use of expert judgment [7]) focus on the overall validity of the DT, without considering whether parts of the tree may not be valid. Existing quantitative techniques (e.g., the use of edit distance [8]) are not integrated into methodological approaches, and thus, it is not clear how their results can be applied to improve DTs.

Nevertheless, systematic approaches to validity are necessary. In a DT that diagnoses diseases, for example, it is not sufficient to say the DT generally outperforms the human doctor. It is possible for the situations where the DT does not outperform the human doctor that the tree prescribes fatal medicine. We must validate not only overall tree performance but also the performance of individual tree branches.

Manual assessment of the tree can be very challenging, as the problem of assessing every branch is time-intensive—the growth of branching is exponential. There are no automated ways of testing, as this requires domain knowledge. Furthermore, validation is not just simply identifying that a DT or a branch of the tree is correct or incorrect. We also want to know why the tree is incorrect or what systematic problems exist. For example, if there are numerous errors in the lower branches of the tree, these could all be caused by a systematic structural error at the top of the tree. This article introduces a methodology for validating DTs that validates not only the overall tree but subparts of the tree as well.

The rest of this article is organized as follows. We first introduce DTs and demonstrate the gaps in existing validation techniques. Our methodology relies on the use of two independent experts to develop independent DTs which we then assess. Correspondingly, we need to categorize the possible differences between the two DTs. We, thus, follow our literature review with a taxonomy of such errors. Following this, we present a modified
q-sorting approach alongside an example of its use to develop an Instagram self-efficacy DT. Finally, Section VII conclude this article.

II. DIAGNOSTIC TREES AND THEIR PROPERTIES

DTs are designed to explore potential root causes of a phenomenon, where large constructs are unpacked to explore and identify the different dimensions. Each dimension is represented as one or more questions and their corresponding answers, which provide information. The information is arranged in a tree structure, where questions lower in the tree more closely relate to a potential root cause. The decision process involves performing a depth-first traversal of the tree, where each stage of the traversal involves choosing a branch representing particular information. As one navigates deeper down the tree, more information is accumulated. This continues until either the end of the tree or a particular depth of the tree is reached, whereupon a decision can be made.

DTs have particular properties. First, all items except the penultimate child have at least two children. An item with only one child has no “branching,” and hence, no accumulated information and no choice; clearly, this cannot be allowed. Second, items higher up the tree are formatively defined by items lower down the tree. Hence, items concerning higher level concepts are mapped to items with greater precision. The items, thus, have a parent–child relationship. This means items higher up in the tree are more important than those lower in the tree. Thus, it is important to first validate the top of the tree, then the next level, and so on. This saves effort as there is no point developing subitems for a poorly defined item.

A. Diagnostic Tree for Assessing Instagram Self-Efficacy

As an example, consider a DT on perceived skill using Instagram (i.e., Instagram self-efficacy), designed to identify systematic reasons for users’ failure to engage with various Instagram functions. Self-efficacy is key for understanding how individuals adopt new tools and develop skills in the use of those tools. It is a pivotal concept for understanding technology acceptance, implementation, and use [9]–[11]. Self-efficacy is, “People’s judgments of their capabilities to organize and execute courses of action required to attain designated types of performances. It is concerned not with the skills one has but with judgments of what one can do with whatever skills one possesses” [12, p. 391]. Because people who have low self-efficacy in Instagram are less likely to use it in the future as compared to those with a high degree of self-efficacy [10], [13], it is important to consider improving Instagram self-efficacy as part of a strategy to promote use.

The DT designer has determined there are the following five principal issues (overarching constructs) that users could have.

1) Linking Instagram to other social media accounts, where users have difficulty connecting their Instagram account and content such as posts and stories to their other social media accounts (such as Twitter, Tumblr, and Facebook).

2) Social information management, where users experience difficulty in managing social information (e.g., posts or stories) produced by others on the platform. Users may find certain functions such as tagging, retagging, copying, and replying challenging [14], [15].

3) Content creation, where users have difficulty creating new content to be consumed by others.

4) Interactions, where users have difficulty interacting with other users, such as on chats or video calls.

5) Account management, where users have difficulty configuring their personal account settings.

These were identified as a result of an inter-rater-based q-sort identified in Step 2 of our methodology detailed in the following. Each overarching construct in turn unpacks to hundreds of possible specific causes. For example, “editing” can be unpacked to items such as choosing a filter, adjusting certain settings (i.e., light, color, and contrast), using effects, using photo props, and using additional photo editing apps with Instagram. Fig. 1 presents the example of Instagram self-efficacy DT.

Fig. 1. DT for Instagram self-efficacy.

III. EXISTING APPROACHES TO DIAGNOSTIC TREE VALIDITY

Existing approaches to validating DTs essentially take two forms: methodological and quantitative techniques. Most methodologies do not integrate the more advanced quantitative techniques, and quantitative techniques are often employed in isolation.

A. Methodological

The major validation methodologies used for validating DTs are the following.

1) Expert Review: One common approach to validating DTs is the expert review, where experts are given the tree and methodologically examine it for errors [16]. Expert reviews can provide valuable insight into usability problems and can be beneficial in evaluating a DT across all stages of development [17]. One example is the efficient machine fault diagnosis expert system, where validation was done by developing DTs with the use of multiple domain experts who identified different issues within the system [3]. Another is the healthy habits system, where three experts with 5–15 years of research experience in their individual fields of expertise reviewed the DT and individually identified strengths and shortcomings [16].

2) Comparison: Much research structures the DT validation as a competition or test [18]. In some situations, the test is against a dataset with known properties, the correct diagnosis is known, and the DT is evaluated as fit for purpose if it correctly
diagnoses a certain number of cases (precision) and does not fail to misdiagnose a certain number of cases (recall). For instance, discrete-event systems are often tested by comparing them to a known dataset [19].

In other situations, experts are considered the proxy for the correct answer, and the DT is evaluated to determine how close the DT’s answers are to the experts’ [3]. Air handling units are often tested in this way [20].

The principal problem with comparison approaches is only the overall performance of the tree is evaluated. It is possible for the tree to overall perform better than other systems, but perform worse on particular problems. Also, in some cases, DTs perform better at comparisons because of factors unrelated to the quality of the tree. For example, DT systems can outperform human experts simply because they are more consistent, do not suffer from fatigue, and are immune to cognitive biases like framing [7].

B. Quantitative

Various quantitative techniques have been proposed for validating DTs. However, these quantitative techniques are generally applied in isolation, rather than being incorporated into systematic methodologies. Most quantitative techniques also suffer from problems in their measurements. Applicable quantitative techniques include edit distance, factor analysis, and cluster analysis.

1) Edit Distance: When an edit distance algorithm is applied to a DT, one assumes two DTs built for the same purpose, for example, two separate trees developed by two independent experts. The edit distance algorithm is applied to identify how dissimilar the two trees are with the implication this dissimilarity indicates problems with one or both trees. Edit distance algorithms calculate the number of changes (typically identified as insertions, deletions, and updates) necessary to transform one tree into another [21], [22]. There are a number of limitations of edit distance algorithms for facilitating DT validation. First, edit distance algorithms produce summary statistics on the overall similarity of two trees. These do not facilitate a diagnosis of errors. They do not, for example, tell us that most errors occur at the top of the tree (very bad) or at the bottom of the tree (not so serious), or tell us that most of the errors are occurring in the children of node 1. Second, edit distance measures do not take into account the sample size. Clearly, if there are two trees, each having 50 nodes, where ten changes are required to transform one into the other, this is different from two trees, each having 500 nodes where only ten changes are required. In statistical thinking, we want to compare the statistic to some probability distribution to standardize results according to “sample size.” We then calculate confidence intervals or p values of significance, where the threshold (typically 0.05 or 0.01) is sample size independent. The tree edit distance literature has no equivalent analog.

2) Factor Analysis: Factor analytic techniques have also been proposed for DTs [23]. Factor analysis works by grouping together items that are highly correlated or testing a proposed model of how correlated items are to the actual correlation between items [24]. Factor analysis is difficult to apply to DTs because traditional factor analysis was designed for tabular, rather than the nested data found in trees. As a result, factor analysis can only be applied to very simple trees of low nesting [25]–[27]. In addition, highly nested trees have statistical properties that confound factor analysis. In a good nested DT, the children of a particular item are highly orthogonal (i.e., uncorrelated). However, these children are moderately correlated with their parents. For example, consider the items “1) I can edit a photo using Instagram,” “2) I know how to crop my photo,” and “3) I know how to apply effects to my photo.” 2) and 3) should relate to 1), because 1) is their parent, if you know how to crop or apply effects, you know something about editing photos. But 2) and 3) are orthogonal to each other—knowledge of cropping should not impact knowledge of effect application and vice versa. Hence, they should not have a strong correlation with each other. Traditional factor analytic measures such as factor loading, “which is the correlation between the original variables and the factors, and the key to understanding of the nature of a particular factor” [24, p. 89] or structural-equation model-based confirmatory factor analysis [28], [29] have limited applicability to such situations.

3) Cluster Analysis: Cluster analysis employs techniques similar to factor analysis. In many cases, cluster analysis employs similarity measures distinct from correlation [24], [30]. DTs that employ cluster analytic techniques include those in customer relationship management, where customer reviews are mined for commonalities which are then employed to diagnose product weaknesses [31] and in bankruptcy prediction [32].

Cluster analysis algorithms employ their own specific distance measures to the cluster. Examples include single linkage clustering [28] and weighted pair groups clustering [29]. These distance measures assume particular properties of the underlying data, which are not always true. As a result, many cluster analytic techniques generate spurious results [35], [36].

The main problem with quantitative techniques generally is the garbage-in-garbage-out problem. Quantitative techniques establish the validity of the results based on assumptions about the numeric properties of valid DTs. Often, valid DTs do not have those numeric properties. For example, they have a sample size in which the technique was not calibrated for (edit distance) or have correlational properties not assumed by techniques such as factor analysis or cluster analysis. It is well recognized that quantitative techniques should only be applied when the technique user has a thorough grasp of the numeric properties of the data. However, missing from the conversation on applying such techniques to DTs is an understanding of how the technique user can obtain sufficient understanding for ascertaining the viability of a quantitative technique.

IV. ERRORS ACROSS PAIRS OF DIAGNOSTIC TREES

Definitions: We are proposing a methodology based on comparing DTs developed independently by two experts. In the following, we formally define terms employed in the remainder of this article, based on definitions in [37].

1) Root is the node with no parent. The root of the tree is on level 0.
Fig. 2. Presents problematic item A for the DT. (a) Hierarchy movement. (b) Level movement. (c) Diagonal movement.

2) *Top level* is first-degree descendant of the root. The top level of the tree has a level of 1.
3) *Descendant* is the *n*-th degree child of a parent. A first-degree descendant of a node is also called a child node.
4) *Parent* is a node with descendants.
5) *Branch* includes the descendants that share the same top-level node.
6) *Level* is the distance of a node from the root. A node is on the *n* + 1 level of its parent node. As an example, a node located on the third level is three levels below the root node and its parent is on the second level. Levels closer to the root are considered higher levels and levels further from the root are considered lower levels.

V. TAXONOMY OF DIFFERENCES ACROSS TREES

For our methodology to work, we first need to identify possible ways two DTs can be structurally dissimilar. Given two DTs *T* and *T’*, the following are the ways the trees can be arranged differently. In each arrangement, the difference in position of item A represents a certain type of error. Fig. 2(a)–(c) presents these separate arrangements.

Hierarchical Movement: In Fig. 2(a), A is found in the same branch of items of the tree but is placed at different levels. In the figure, A across both trees is a descendant of item 1. However, while in tree *T*, A is directly mapped to item 1, A in tree *T’* is directly mapped to item 5. We call this a hierarchy movement.

Level Movement: In Fig. 2(b), A is from completely different families of items but is found at the same level. In tree *T*, A is a direct child item of item 1, while in tree *T’* it is a direct child of item 2. In both cases, A is on the same level. We call this a level movement.

Diagonal Movement: In Fig. 2(c), A is both in a different level and family of an item of the tree. We call this a diagonal movement.

Swap: In Fig. 3, there is another item B where A and B have changed places. We call this a swap. Swaps are considered because they reflect a single cognitive difference between two experts. There are effectively three kinds of swaps (hierarchical, level, and diagonal). In the contrasting examples of Fig. 3, A and B have swapped places.

Missing data points: Finally, it is possible for one person to be unable to map an item into the tree, while the other was able to do so. In total, there are, therefore, seven kinds of errors comprising three kinds of movements, three kinds of swaps, and a situation where one person put an item in the tree, but the other did not.

VI. MODIFIED Q-SORT APPROACH

Our modified q-sort methodology is intended to ensure a constructed DT is consistent across experts (i.e., has validity). We cannot use a traditional q-sort [38] because it only has one level of branching, where you group all items into a set of buckets. Our methodology not only assesses the validity of the tree but also diagnoses where in the tree the inconsistencies are between experts. Our q-sort methodology requires three key components.

1) Access to knowledge sources, such as a library, or online forums: The knowledge sources must be rich enough that the items for the DT can be obtained from these knowledge sources. A variety of knowledge sources may be used. These include posts in online forums, videos, tutorials, reviews, letters, biographies, speeches, reports, books, [39] interviews, focus groups, and observations of the phenomenon. What a relevant knowledge source is will differ depending on the DT domain and, in most cases, a collection of knowledge sources may be required.

2) At least two experts. Experts must be blind and independent of the study: Most studies have indicated that two experts are sufficient for assessing inter-rater reliability
However, in cases where the context is new, then more than two experts may be required [42]. Experts should have the following characteristics. One, experts must be experts in the domain. Two, experts must have a strong command of the language of the topic and DT. A strong command of the language is necessary because the hierarchical layout of DT items means experts must understand words that clue a reader into whether an item is more or less specific. We found individuals with poor command of the English language fail to comprehend such hierarchy-related words as “overall” or “generally.”

3) **Access to and comfort with the use of the following pieces of equipment.**
   a) Spreadsheet software, such as Microsoft Excel, to be able to transfer the DT into a table, where one column represents the parent items and the other the child items. This is important to analyze the correspondence between the experts’ trees.
   b) Statistical software, such as SPSS, to calculate Goodman and Kruskal’s Lambda, Cohen’s Kappa, and Goodman and Kruskal’s Gamma [43] to analyze experts’ trees.
   c) Digital boards that allow one to manipulate and save a tree structure.
   d) Nondigital tools such as stationery (paper, cards, or sticky notes), cutting tools (scissors, cutters, or clipper), and writing tools (such as pencil, pen, or marker). These are used to create cards to capture each item in the DT item bank. Experts physically drop the cards into cardboard boxes in a manner similar to a traditional q-sort [38]. The boxes equivalent to the number of top-level constructs in the tree (see Step 2).
   e) Collaborative software (such as Microsoft Team or Zoom) allows simultaneous work among the experts, which is useful in reconciling items of the DT.

We have found it important to have tools capable of both representing DTs in a digital (digital boards) and physical (items on paper sorted into boxes) environment. Some examples of ideal digital formats are digital whiteboards or large screens. Digital formats allow experts to visualize the overall tree, save their work, and be able to move items around by using tools such as cut and paste. Saving work is important because experts often want to restore their work from a prior point. In addition, paper formats are useful as they can be worked on in different locations (i.e., at home and office) and can be spread over a large area (e.g., the floor).

As illustrated in Fig. 4, our q-sort methodology has the following steps.

1) **Build items for the DT.**
2) **Identify top-level constructs for the DT.**
3) **Incorporate distractors.**
4) **Select one top-level construct.**
5) **Sort and map process.**
6) **Revise the tree.**
7) **Calculate, evaluate, and interpret inter-rated scores.**
8) **Address structural discrepancies.**
9) **Create the final DT.**
10) **Empirically validate the DT.**

The components required for each step are identified by color-coding the steps. Specifically, blue indicates the use of knowledge sources, green indicates the use of blind and independent experts, and orange means both the use of technology and nondigital formats. To illustrate our q-sort methodology, we employ an example of developing a DT to elicit problems users have when using Instagram.

**Step 1: Build the items for the DT.** This step is supposed to identify all items that will be employed in the DT, as shown in Fig. 5. By the end of this step, the items should capture the entire scope of the problem domain of the DT. The concept of scope encompasses both 1) that all subdimensions of the concept captured in the DTs are represented in the items, and 2) for any possible subdimension, all possible actionable causes of that subdimension are captured in the items. Step 1 comprises two substeps which iterate: (a) build the item bank and (b) assess the item bank.

**Step 1(a). Build Item Bank.** For step 1(a), We examine each knowledge source, which provides a distinct understanding of possible root causes of the domain. For example, existing journal articles capture information about Instagram and Instagram self-efficacy [10], [44]–[46]. Similarly, social media sources, such as reviews and blogs, can provide a wide range of in-depth information on issues a user may face with a product or service [47], [48]. For our DT for Instagram, a variety of sources were used, such as websites (Instagram help and troubleshooting), reviews and comments from different app stores (Apple app store and Google play, and forums (Reddit). Each provides inspiration to create the item bank.
We employ the term “item” in the same way as Edwards and Bagozzi [49, p. 156], i.e., “observable quantifiable scores obtained through self-report, interview, observation, or other empirical means.” A “concept” is a mental image or general notion of something. It summarizes observations and ideas about all the characteristics of that image [50, p. 49]. For instance, one online review stated, “I tried to set up an account on Instagram and an automatic pop-up told me someone was already using my email on Instagram.” In this context, several concepts exist, including account creation, pop-up, and email in use. New items can be developed based on these concepts such as “I know how to create an account using my email” and “I know what to do if I cannot sign up for an account because an account with my email address or phone number already exists.” As another example, a comment from 2018 from the Apple app store stated, “You can post whatever you want to add anything to your story and add filters, stickers, and gifs to your photos and you can explore different accounts and bookmark, like and comment on everything you find.” Here new concepts such as “adding to a story” or “exploring bookmarks” can be identified. Consequently, items such as I know how to add a story or I know what to do if I cannot sign up for an account because an account with my email address or phone number already exists.” As another example, a comment from 2018 from the Apple app store stated, “You can post whatever you want to add anything to your story and add filters, stickers, and gifs to your photos and you can explore different accounts and bookmark, like and comment on everything you find.” Here new concepts such as “adding to a story” or “exploring bookmarks” can be identified. Consequently, items such as I know how to add a story or I know how to use a bookmark can be constructed. As a result of step 1(a), we collected approximately 260 items for our self-efficacy Instagram survey.

For step 1(b), we test items for the following issues.

i) **Similarities across items:** If two items essentially capture the same idea, one is dropped. We dropped approximately 40 duplicates in our Instagram self-efficacy survey, and 220 items remained.

ii) **Scope testing:** We test for “theoretical saturation” [51]. This is done by ascertaining whether new items were created from step 1(a). We determine a threshold of a number of times to check for new items before we give up. In our Instagram self-efficacy instrument, this number was 10. This number is determined by the complexity of the context of the study, availability, and sample size of the knowledge sources. If new items were created in step 1(a), we reset the number of times back to 0, as there is potential for identifying more items and loop back to step 1(a). Otherwise, we increment the number of times by 1 (several sources have already been assessed). If the number of times is less than the threshold, we loop back to step 1(a). Otherwise, we consider that there is no further information from the knowledge sources which we can build our items upon that can be obtained and stop.

iii) **Content validity:** We test items for relevance to the context of the study. This is done by employing two independent and blind experts who go through the items. If both experts mark any item as irrelevant to the context of the study, that item is dropped. If only one expert marks an item as irrelevant, then that item is either dropped or edited, as this flags an error. For example, in our Instagram self-efficacy DT, several items referred to using Instagram for business. However, in our context, we only focused on perceived Instagram skills for daily use. These items were marked as not relevant by both experts and dropped. In our DT, nearly 20 items were dropped, leaving 200 items.

**Step 2: Identify top-level constructs for the DT.** Top-level constructs formatively define the domain of the DT. This step determines what these constructs are, so all items can be mapped to these constructs. There are two ways to identify top-level constructs. One way is to perform a traditional q-sort. There are two varieties of traditional q-sort possible, both of which require two independent and blind experts. In the first variety, boxes are prelabeled, and experts sort items into the boxes. In the second variety, experts determine the number of boxes and sort into those boxes. In both varieties, inter-rater reliability is assessed using the 0.6 Cohen’s Kappa threshold used for exploratory research [52]. As an example, in our Instagram self-efficacy DT, we employed two blind and independent experts to go through the items and put them into as many categories as they saw fit. Each expert grouped the items in five categories independently, where the researcher then labeled each category based on the items inside. The five identified categories were: 1) linking Instagram to other social media accounts; 2) social information management; 3) content creation; 4) interactions; 5) account management. We then assessed inter-rater reliability. Kappa was 0.601, above the recommended threshold for exploratory studies [52]. If Cohen’s Kappa threshold is met, there are still other issues that need to be resolved. For instance, some items may not map to the top-level constructs. When this occurs, a decision has to be made as to whether additional top-level constructs should be established. In this case, a new box is labeled and added to the other boxes. The q-sorting is then repeated, and inter-rater reliability is assessed.

Another way is to review the literature to establish the framework and scope of the top-level constructs of the DT. For instance, in the context of computer self-efficacy (CSE) which is a person’s judgment of his or her ability to use a computer system, the top-level constructs can be developed using the theoretical framework of a study such as Scott and Walzak [53] who unpack CSE to cognitive engagement, prior experience, computer anxiety, and organizational support.

**Step 3: Incorporate duplicate and distractor items in the bank:**

This step is principally employed to partial out the error of the DT instrument from the error associated with the expert. The typical DT test bank can contain hundreds of items. The duplicate and distractor items are used to assess rater attentiveness during the q-sorting methodology. Duplicate and distractor items are traditional mechanisms for assessing rater attentiveness [54]. It is important to ensure that distractor items are clearly independent of items being assessed by the expert. When experts miss the fact that items are duplicated, or categorize distractor items along with legitimate ones, it signals a lack of rater attention. For example, on a DT about Instagram, one distractor item could be “The education level is sufficient” as this is clearly independent of the Instagram context.

It could be argued that distractor items are unnecessary because poor inter-rater reliability would serve as an effective proxy. However, inter-rater reliability is also indicative of poor
item phrasing. It is necessary to be able to partial out the effect of experts and items separately, as fatigue is a real concern because of the number of items. The number of distractors is subject to the complexity and novelty of the context and the number of items an expert would need to sort. For instance, in our Instagram self-efficacy DT, the item bank consisted of 200 survey items, to which ten duplicate items and distractors were added. Hence, a total of 210 items were given to the experts.

**Step 4: Select one top-level construct at a time.** As each DT can contain hundreds of items to map all the items at once can be exhausting and too heavy of a cognitive load for experts to perform in a single session. Instead, experts are only given the items from one top-level construct to q-sort before moving on to other top-level constructs and their items. Also, they are given a period of time (e.g., a week) to finish q-sorting that branch. This is ok because the validity of the top-level constructs is already addressed in step 2. In this step, the branch to validate is selected. In our example, there were five top-level constructs. Experts began with the items in the top-level construct, “linking Instagram to other social media accounts.” Once this was completed, they moved to the items in “social information management” and so on.

**Step 5: Sort and map items into a tree.** Two blind and independent experts who have sufficient knowledge and skills need to be recruited. In our study, we employed experts who had experience and knowledge of Instagram skills. In addition, prior to study commencement, experts are trained to explicitly draw the tree. It is necessary prior to asking for a q-sort that experts be given examples of tree diagrams from other domains. Without such illustrations, experts tend to perform traditional qsorts. After training, experts are given a set of items and are told to map items in a tree. Each item is assigned a number from 1 to N, where N is the total number of items. The “root” item, which is a dummy item developed for statistical purposes, is given the number 0.

There are two sets of mapping rules. The second set is more comprehensive and designed to identify certain types of errors with the items. The first set of rules is disclosed to the experts who map the tree according to those rules. The first set of rules are the following.

1) Items that share a common theme are grouped together.
2) Each group should be labeled.
3) An item within each group which conceptually matches each label should be identified. For instance, if a label “security of account” is given, then items such as resetting the password, two-factor authentication, and security code may be mapped to that label. If such an item does not exist, then the expert should tag that group, such as circling the items or marking the group with a tick.
4) Within each group, subgroups are then created, and the labeling process continues.
5) Items concerning lower level concepts (children) are mapped to higher level concepts (parents). The root (dummy with value 0) has no parent.

Once the mapping is done, the second set of rules is applied by the researchers to the mapping to identify further problematic questions. The additional rules in the second set are the following.

1) An item cannot have just one child item. It must have two or more children.
2) A single parent should not have more than seven child items mapped to it. From a cognitive perspective, the number of objects an average human can hold in short-term memory is $7 \pm 2$ [55]. Based on this, other studies have suggested options and response categories should be constrained to this number [56], [57]. Therefore, we constrain branching to seven branches because when too many options are offered simultaneously, the burden placed on memory is increased [58], [59]. This in turn causes people to be unable to choose [55].
3) Except for items identified as distractors or duplicates, the tree must be fully connected.

By withholding the second set of rules from experts, we are able to identify certain classes of problematic items. If only one child item is mapped to a parent item, this indicates that the parent item is either underdeveloped or we have not identified sufficient child items representing subconcepts of that item. Similarly, if an item has more than seven child items, this suggests the concept is too complex, and perhaps should be refined into a smaller subset of child concepts. Finally, if an item other than the distractor is not mapped to any item by one or more experts, one of two possibilities exists: 1) the item is badly worded or vague, or 2) experts have made mistake(s) because of (for example) fatigue. Feedback from this step is employed to refine the items in the DT.

As an example of the process of mapping items into a tree for the first set of rules, consider Fig. 6, which presents the result of two experts developing trees around the concept of the overarching top-level construct “account management for
Step 6: Revising the DT. In this step, the experts are given the second set of rules and the items refined from the previous step and asked to remap the tree. We continue to assess the face validity [60] of the items by employing two blind and independent experts. Any item marked by either expert as confusing is re-edited. As an example of this process of mapping items, consider Fig. 7, which presents the result of the two experts developing trees around the concept of “account management for Instagram.” The trees that experts developed are clearly discrepant. In the following steps, we show how we identify and reconcile discrepancies.

Step 7: Calculate, evaluate and interpret inter-rater scores. To determine whether trees are “similar enough” to be valid, we perform an iterative contingency table analysis to evaluate the overall fitness and diagnose the problems. This step consists of two sub-steps, (a) transformation of the DT into a table and (b) the assessment of the inter-rater scores.

Step 7(a). To calculate the similarity of each tree, certain measures of association are used. To perform these calculations, the trees developed by the experts need to be transformed into a table with five columns, which are the list of items (1 column), the parent column from each expert (2 columns), and the level columns for each expert. In the analysis, we compare the parent columns of the two experts using the level columns to restrict the data for analysis. As an example, Table I demonstrates the transformation of the tree in the first round for the top-level construct “account management for Instagram.” Expert 2 has mapped three items (2, 9, 18) to item 1, while expert 1 has mapped six items (3, 9, 18, 26, 32, 40) to item 1. The results in Table I correspond to the diagrams in Fig. 7.

These tables are not analyzed in their entirety immediately. Instead, we begin the analysis with the top three levels of the tree and work our way downward one level at a time. This is because problems at higher levels can impact the lower levels. Thus, initially, only data from the top three levels of the tree are analyzed. Then, data from the top 4 levels, and so on.

As a result of the disagreement, it is possible for experts to have a different number of items at each level. Here, expert 1 has created a tree with four levels, while expert 2 has created one with five levels. Tests on contingency tables for two different sample sizes cannot be done. To address this, if an item does not exist for one expert, we replace the null value that represents the mapping in the table with a number that has not been previously assigned. This method of handling null values biases statistical results downward [61], which is desirable, as lower statistical values indicate poorer fit. As an example, consider the two trees created by expert 1 and expert 2 in Fig. 7. We first identify the top three levels, which are presented in Table II. Items such as 12 and 13 for both experts are both in level 3. However, in the example, experts disagree on the level of such items as 4, 5, 6, and 7. For these items, we insert dummy items (items 74–100) for expert 2, as presented in Table II. The dummies are italicized in the table.

Step 7(b). In the second substep, as presented in Fig. 8, we calculate the statistics of Goodman and Kruskal’s Lambda, Cohen’s Kappa, and Goodman and Kruskal’s Gamma [43]. These three statistics together provide useful information for evaluating the similarity of the two trees. Prior research recommends Lambda > 0.7, Kappa > 0.4, and Gamma > 0.3 as good thresholds for tree fit [37]. This corresponds to about 30% of the two trees being different. Thus, so long as the trees meet satisfactory thresholds for similarity, we add one
more level and perform the comparison again. If the entirety of both DTs built by the experts meets the thresholds, then the final tree can be built in step 9.

If the measures do not meet satisfactory levels, then the scores are interpreted to identify problematic items. These items will then be edited, and experts are asked to remap the items in step 8. This evaluation process is guided by our three statistics, which each behave differently depending on what is inconsistent between the two trees [37]. As presented in Table III, Gamma is particularly sensitive to differences in levels (hierarchy movements or swaps), while Kappa is sensitive to differences within a level of the tree (level movements or swaps). Lambda is sensitive to “movements” where a single node or branch differs between the two trees, and not sensitive to “swaps” where two nodes or branches are substituted.
In our Instagram self-efficacy case, Lambda ($\lambda$), Kappa ($\kappa$), and Gamma ($\gamma$) for the top three levels were calculated as per Table IV. Lambda ($\lambda$) is 0.532, Kappa ($\kappa$) is 0.109, and Gamma ($\gamma$) is 0.104. Here, Kappa and Gamma are the lowest scores, which identifies that the principal problem is disagreement with the mapping of items of the same level and hierarchy. Visually, as illustrated in Fig. 7, we can see this as expert 2’s tree has more levels than expert 1, while expert 1 has more branches. This affects items 3, 5, 11, 12, 17, 19, 20, 21, 22, 26, 27, 32, 35, and 40. In addition, the measures indicate many diagonal movements, which can be identified as items 11, 17, 26, and 32. The resolution of these discrepancies is addressed in step 8.

When these issues are resolved and experts remapped the items (i.e., came back from step 8 to step 7), the measures were recalculated as per Fig. 9. This time, Lambda ($\lambda$) was 0.711, Kappa ($\kappa$) was 0.556, and Gamma ($\gamma$) was 0.719, which met the thresholds and allowed us to proceed to the next level. We added the fourth level and fifth level and recalculated Lambda ($\lambda$), Kappa ($\kappa$), and Gamma ($\gamma$), and edited the items accordingly. The final results are in Table IV.

**Step 8: Diagnosing the trees.** In this step, as presented in Fig. 9, based on the results from step 7, we identify the problematic items and systematically resolve them. As a result of the analysis, we edited items 3, 5, 11, 12, 17, 19, 20, 21, 22, 26, 27, 32, 35, and 40. For instance, we edited item 40 from “I know how to secure my Instagram” to “I know how to secure my Instagram account” in which we added the term “account”—the term “my Instagram” is too general and could mean several things, such as one’s profile, one’s posts, or one’s login. We then gave the experts the modified set of items and asked them to redo the mapping.

**Step 9: Building the final DT and design.** When satisfactory levels of significance and suitable thresholds are achieved, items that experts disagree on need to be reconciled and the design of the tree finalized. This is done in three steps. First, the experts are given the problematic items and we explain why such discrepancies exist using the statistics Goodman and Kruskal’s Lambda, Cohen’s Kappa, and Goodman and Kruskal’s Gamma [43]. Next, we ask the experts to discuss the discrepancies. Finally, depending on the consensus decision by the experts, the items or the mapping of the items are revised. In cases where experts cannot agree, either the researchers intervene and assist with the final mapping, or alternatively a new set of experts are invited and asked to do the q-sort. In our case, this was not necessary as experts were able to arrive at a consensus for all discrepancies. As an example, consider the DT presented in Fig. 10. After the reconciliation, several items were remapped. For instance, in the third round, experts disagreed on which item should be the parent of item 40 “I know how to secure my Instagram account.” Expert 1 mapped item 40 to item 2 “I know how to create an Instagram account,” while expert 2 mapped item 40 to item 18 “I have the ability to control the visibility
and privacy of the Instagram account.” After discussion, the experts and researchers agreed to map item 40 to item 18 “I have the ability to control the visibility and privacy of the Instagram account.”

It should be observed that while we were concerned with the number of branches each item could have (i.e., the “breadth” of the tree), we implemented no controls on the “depth” of the tree. The rule of thumb when developing instruments requiring user feedback is to have at most ten questions. This normally takes users 5 min to fill. Any longer, and there is an increased risk of fatigue and dropouts [62], [63]. From the perspective of DTs, a DT with two branches per item nested ten levels deep would have 2^10 or 1024 questions, i.e., more questions than need to be asked. Thus, typically, the depth of the tree structure does not pose an issue in design.

**Step 10: Empirically validating the DT.** Our DT is designed to identify systematic reasons for users’ failure to engage with various Instagram functions. There are clearly many forms of validity, such as construct, internal, and external validity [64], and a full validation of any measurement instrument is time intensive. One way to validate a DT is to assess it against an external criterion. For Instagram self-efficacy for account management, online comments or reviews were used as an external criterion. We extracted comments from Reddit, in particular subreddits, such as, “r/Instagram” or “r/socialmedia,” using keywords such as account, profile, privacy, and security, because it concerns complaints about Instagram. We then retrieved all the data from the search. We then cleaned the data and dropped duplicates. We obtained a total of 193 comments. Next, two independent experts were given the comments and the categories and were asked to map each comment to the categories. In our research design, experts could elect to not map comments to categories, and this occurred. When unmapped comments were reviewed by researchers, it was determined they were either irrelevant to the context of the study or vague and hence were dropped. For instance, the comment “Instagram blocks my Live for using my own music” was dropped as it was marked as irrelevant, as it was not related to the context of Instagram self-efficacy for account management. A total of 41 such comments were dropped. The inter-rater reliability was significant, and kappa was 0.603. Interestingly, of the 153 comments, expert 1 and expert 2 could only map 52 and 62 items, respectively, to levels below the first two levels. What this means is that the DT allowed for a more in-depth diagnosis for why Instagram users had account management problems than the actual Reddit complaints. This, therefore, provides some evidence our DT is superior to at least other traditional sources of diagnostic information (i.e., online forums).

**VII. CONCLUSION**

In this article, we introduced a new q-sort methodology for validating DTs. Existing approaches are limited in their ability to assess these types of trees for a number of reasons. Consider the example of the DT in Fig. 1 for Instagram self-efficacy. Existing expert review and comparisons-based techniques are insufficient because they focus only on evaluating the tree’s overall fitness rather than considering the fitness of the individual branches—they do not include methodological processes for validating tree branches. For example, while the overall tree might be alright, if the “content creation” branch performs poorly, we would continue to have a segment of unhappy users. With respect to quantitative techniques, if a quantitative technique demonstrates our DT has poor validity, it does not then articulate for us what the issue with validity is—we just simply know the validity score is poor. Thus, the key distinction between our and others’ techniques is in our focus on evaluating not only the whole tree but parts of the tree. This is done by having our experts assess only part of the DT at one time, performing quantitative statistical tests on successive elements of the tree, working from the top of the tree down to the end nodes, and using statistical measures that what part of two trees are dissimilar.

This article makes several contributions. First, we introduce a methodology that not only validates the overall DT but also validates the components of the tree. In performing validation, we not only identify that something is wrong, but what is wrong, i.e., whether the problem is at the top or bottom of the tree, and whether the problem is a hierarchical, or level issue. Our methodology is furthermore systematic and follows a step-by-step procedure that others can follow. Our methodology, thus, provides a rigorous way by which existing DTs (e.g., in expert systems and follow-up customer satisfaction surveys) can be validated prior to launch. In addition, as part of evaluating our validation process, we developed a new DT for Instagram self-efficacy. This DT will prove helpful not only for identifying the key gaps in Instagram’s user experience, but also more generally expands our understanding of user self-efficacy as a concept [13]. Our work allows for a more systematic evaluation of the elements of user self-efficacy and why users perceive themselves as not particularly self-efficacious.

The development of our new q-sort methodology opens up several opportunities for IS research. First, most empirical IS research focuses on explaining causal relationships. For instance, methods such as structural equation modeling are used to examine causal relationships and to test the hypotheses between the observed and latent items in a research model [65]. However, in many situations, one may desire to know why such relationships exist or fail to hold. For instance, Bagozzi [66] highlights that research on the technology acceptance model has demonstrated the relationship between perceived usefulness, ease of use, and intention to use, but cannot articulate why this relationship holds. As a practical matter, we often want to determine why (for example) users find a technology to not be useful, or not easy to use. As an example, researchers have investigated why users find the Web not particularly easy to use; they identified several reasons, such as slow data access, difficulty searching for specific information, information clutter, time delays due to images, the unreliability of sites, and incomplete category searches [67]. In this case, a diagnostic theory would be useful where each end node represents a possible reason why the Web is not easy to use. Our development of a systematic methodology facilitates the development of DTs to answer these kinds of questions.
Finally, as ongoing research on validating our methodology, we used it to create a DT survey of café customer satisfaction and used it in a number of cafes [68]. We compared the results against two other techniques, specifically 1) a traditional customer satisfaction survey and 2) a content analysis of online customer reviews. Preliminary results indicate that our DT has several advantages over the other tools. It has a higher response rate, requires fewer items for respondents to answer, which reduces fatigue effect, has better item discrimination in that respondents tend to provide more extreme scores, and has a higher agreement among respondents for each item than a traditional survey. It also provides a more precise diagnostic than customer reviews, which tend to be short and nondescriptive. Our development and evaluation of this diagnostic tool are ongoing.

As with all validation methodologies, our q-sort methodology has limitations. One, similar to expert reviewers, it is fallible and biased. They often focus on specific kinds of problems and may miss important issues [17]. Two, the results of an expert review depend on experts’ qualifications [7], [69]. Many DTs are complex [70], requiring knowledge from multiple domains [3]. Thus, there may not be experts able to review an entire DT. Finally, most DTs are complex, and experts often experience fatigue during the review, which can compromise review quality [7]. We provide procedures that control the cognitive exhaustion of experts. However, our methodology remains exhausting.

Second, our methodology is best suited for diagnostic theories where one is interested in unpacking the reasons why. Therefore, it is not suitable for establishing relationships among variables, such as in process or variance theories. As an example, in one study, activity theory was used to identify contradictions and congruencies between flow techniques and software development practices, and the dialectical interaction between contradictions and congruencies which can lead to a stage of change [71]. Similarly, in many cases, we may be interested in identifying which variable has the biggest impact on another variable. For instance, the aim of one study was to identify factors that increased the use of technology in the workplace [72] or another who identified factors that influenced knowledge sharing behavior via weblogs [73]. In such a case, our methodology has limited applicability.

Finally, our methodology does not evaluate the nomological properties of the DT. Nomological validity refers to the degree to which the constructs fit within the logical network of theory [74], [75]. In other words, it is a measure of the theoretical correspondence between theory and the measures used in the DT. However, most methodologies do not have statistical tests of nomological validity [1], [75].

REFERENCES

[1] E. Shortliffe, Computer-Based Medical Consultations: MYCIN, vol. 2, New York, NY, USA: Elsevier, 2012.
[2] C. Angeli, “Online expert systems for fault diagnosis in technical processes,” Expert Syst., vol. 25, no. 2, pp. 115–132, 2008.
[3] S. C. Liu, and S. Y. Liu, “An efficient expert system for machine fault diagnosis,” Int. J. Adv. Manuf. Technol., vol. 21, no. 9, pp. 691–698, 2003.
[4] W. J. Hopp, S. M. R. Iravani, and B. Shou, “A diagnostic tree for improving production line performance,” Prod. Oper. Manag., vol. 16, no. 1, pp. 77–92, 2007.
[5] J. Luo, S. Ba, and H. Zhang, “The effectiveness of online shopping characteristics and well-designed websites on satisfaction,” MIS Quart., vol. 36, no. 4, pp. 1131–1144, 2012.
[6] C. L. Sia, K. H. Lim, K. Leung, M. K. O. Lee, W. W. Huang, and I. Beibasat, “Web strategies to promote internet shopping: Is cultural-orientation needed?” MIS Quart., vol. 33, no. 3, pp. 491–512, 2009.
[7] R. M. O’Keefe, O. Balci, and E. P. Smith, “Validating expert system performance,” IEEE Expert, vol. 2, no. 4, pp. 81–90, Jan. 1987.
[8] K.-C. Tai, “The tree-to-tree correction problem,” J. Assoc. Comput. Mach., vol. 26, no. 3, pp. 422–433, 1979.
[9] G. Torkzadeh and T. P. Van Dyke, “Effects of training on internet self-efficacy and computer user attitudes,” Comput. Hum. Behav., vol. 18, no. 5, pp. 479–494, 2002.
[10] D. R. Compeau and C. A. Higgins, “Computer self-efficacy: Development of a measure and initial test,” MIS Quart., vol. 19, no. 2, pp. 189–211, 1995.
[11] Y. Zheng, J. Wang, W. Doli, X. Deng, and M. Williams, “The impact of organisational support, technical support, and self-efficacy on faculty perceived benefits of using learning management system,” Behav. Inf. Technol., vol. 37, no. 4, pp. 311–319, 2018.
[12] A. Bandura, “Self-efficacy: Toward a unifying theory of behavioral change,” Psychol. Rev., vol. 84, no. 2, pp. 191–215, 1977.
[13] M. J. Keith, J. S. Babb, P. B. Lowry, C. P. Furner, and A. Abdullat, “The role of mobile-computing self-efficacy in consumer information disclosure,” Int. J. Hum. Comput. Interact., vol. 25, no. 6, pp. 637–657, 2015.
[14] J. Caverlee, L. Liu, and S. Webb, “The socialtrust framework for trusted social information management: Architecture and algorithms,” Inf. Sci., vol. 180, no. 1, pp. 95–112, 2010.
[15] V. Mirzaee, M. N. Razavi, and L. Iverson, “Improving personal and social information management with advanced tagging,” in Proc. 36th Annu. Conf. Inf. Assoc. Inf. Sci., Jun. 5-7, 2008, pp. 1–8.
[16] T. R. Chang, E. Kaasinen, and K. Kaipainen, “Persuasive design in mobile applications for mental well-being: Multidisciplinary expert review,” in Proc. Int. Conf. Wireless Mobile Commun. Healthcare, 2013, vol. 61, pp. 154–162.
[17] M. Tory and T. Möller, “Evaluating visualizations: Do expert reviews work?” IEEE Comput. Graph. Appl., vol. 25, no. 5, pp. 8–11, Oct. 2005.
[18] H. Blockeel and J. Struyf, “Efficient algorithms for decision tree cross-validation,” J. Mach. Learn. Res., vol. 3, no. 4/5, pp. 621–650, 2001.
[19] G. Lamperti and M. Zanella, “Flexible diagnosis of discrete-event systems by similarity-based reasoning techniques,” Artif. Intell., vol. 170, no. 3, pp. 252–297, 2006.
[20] R. Yuan, Z. Ma, Y. Zhao, and G. Kokogiannakis, “A decision tree based data-driven diagnostic strategy for air handling units,” Energy Build., vol. 113, pp. 37–45, 2016.
[21] K. H. Green and B. P. Ricca, “Graph theoretic methods for the analysis of data in developing systems,” Qual. Quant., vol. 49, no. 5, pp. 2037–2060, 2014.
[22] G. Kinkmüller and I. Weber, “Analyzing control flow information to improve the effectiveness of process model matching techniques,” Decis. Support Syst., vol. 100, pp. 6–14, 2017.
[23] L. R. Novick and S. M. Hurley, “To matrix, network, or hierarchy: That is the question,” Cogn. Psychol., vol. 42, pp. 158–216, 2001.
[24] J. F. Hair, R. E. Anderson, R. L. Tatham, and W. C. Black, Multivariate Data Analysis, vol. 1. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.
[25] M. G. Capra, “Factor analysis of card sort data: An alternative to hierarchical cluster analysis,” in Proc. Hum. Factors Ergon. Soc. Ann. Meeting, 2005, pp. 691–695.
[26] A. Lake, and C. Cook, “Use of factor analysis to develop OOP software complexity metrics,” in Proc. 6th Annu. Oregon Work. Softw. Metrics, September, 1999, pp. 1–15.
[27] J. Sun, “Assessing goodness of fit in confirmatory factor analysis,” Meas. Eval. Couns. Dev., vol. 37, no. 4, pp. 240–256, 2005.
[28] J. Anderson and D. Gerbing, “Structural equation modeling in practice: A review and recommended two-step approach,” Psychol. Bull., vol. 103, no. 3, pp. 411–423, 1988.
[29] R. B. Kline, Principles and Practice of Structural Equation Modeling. Third, New York, NY, USA: Guilford Press, 1998.
[30] D. Jiang, C. Tang, and A. Zhang, “Cluster analysis for gene expression data: A survey,” IEEE Trans. Knowl. Data. Anal., vol. 16, no. 11, pp. 1370–1386, Nov. 2004.
[31] J. Zhan, H. T. Loh, and Y. Liu, “Gather customer concerns from online product reviews—A text summarization approach,” Expert Syst. Appl., vol. 36, pp. 2107–2115, 2009.
[32] C. Serrano-Cinca and B. Gutiérrez-Nieto, “Partial least square discriminant analysis for bankruptcy prediction,” Decis. Support Syst., vol. 54, no. 3, pp. 1245–1255, 2013.

[33] J. C. Gower and G. J. S. Ross, “Minimum spanning trees and single linkage cluster analysis.” Appl. Statist., vol. 18, no. 1, pp. 44–64, 1969.

[34] R. Sokal, “A statistical method for evaluating systematical relationships,” Univ. Kansas Sci. Bull., vol. 38, pp. 1409–1438, Feb. 1958.

[35] D. J. Ketchen and C. L. Shook, “The application of cluster analysis in strategic management research: An analysis and critique,” Strateg. Manag. J., vol. 17, pp. 441–458, 1996.

[36] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, Cluster Analysis, 5th ed., Hoboken, NJ, USA: Wiley, 2011.

[37] S. Sabbaghan, C. E. H. Chua, and A. E. Gardner, “Statistical measurement of trees’ similarity,” Qual. Quant., vol. 54, pp. 781–806, 2020.

[38] J. Block, “An introduction to the Q-sort method of personality description,” in The Q-Sort Method in Personality Assessment and Psychiatric Research. Springfield, IL, USA: Charles C Thomas, 1961, pp. 3–26.

[39] B. G. Glaser and A. L. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research. London, U.K.: Aldine, 1967.

[40] J. Fleiss, B. Levin, and M. Cho Paik, Statistical Methods for Rates and Proportions. Hoboken, NJ, USA: Wiley, 2013.

[41] K. L. Gwet, “Computing inter-rater reliability and its variance in the presence of high agreement,” Brit. J. Math. Statist. Psychol., vol. 61, no. 1, pp. 29–48, 2008.

[42] J. Vargo, J. C. Nesbit, K. Belfer, and A. Archambault, “Learning object category labels,” in Proceedings of the 20th Annual Conference of the International Group for the Psychology of Mathematics Education, vol. 2, pp. 293–300, 2006.

[43] W. H. Goodman and L. A. Kruskal, “Measures of association for cross classifications,” J. Am. Statist. Assoc., vol. 49, pp. 732–764, 1954.

[44] P. Sheldon and K. Bryant, “Interaction: Motives for its use and relationship to narcissism and contextual age,” Comput. Hum. Behav., vol. 58, pp. 89–97, 2016.

[45] K. P. Hocevar, A. J. Flanagin, and M. J. Metzger, “Social media self-efficacy and information evaluation online,” Comput. Hum. Behav., vol. 39, pp. 254–262, 2014.

[46] G. Walsh, M. Schaarschmidt, and H. von Kortzfleisch, “Minimum spanning trees and single linkage clustering,” J. Interact. Mark. Vol., no. 3, pp. 269–282, 2000.

[47] Y. Yoon, T. Guimaraes, and Q. O’Neal, “Exploring the factors associated with expert systems success,” MIS Quart., vol. 19, no. 1, pp. 83–106, 1995.

[48] S. Dalal and R. S. Chhillar, “Empirical study of root cause analysis of software failure,” ACM SIGSOFT Softw. Eng. Notes, vol. 38, no. 4, pp. 1–7, 2013.

[49] D. Deane and K. Conboy, “Going with the flow: An activity theory analysis of flow techniques in software development,” J. Syst. Softw., vol. 133, pp. 160–173, 2017.

[50] V. Viswanath and B. Hilloll, “Technology acceptance model 3 and a research agenda on interventions,” Decis. Sci., vol. 39, no. 2, pp. 273–315, 2008.

[51] T.-T. Yu, L.-C. Lu, and T.-F. Liu, “Exploring factors that influence knowledge sharing behavior via weblogs,” Comput. Hum. Behav., vol. 26, no. 1, pp. 32–41, Jan. 2010.

[52] B. J. Calder, L. W. Phillips, and A. M. Tybout, “Beyond external validity beyond external validity,” J. Consum. Res., vol. 10, no. 1, pp. 112–114, 1983.

[53] S. B. Mackenzie, P. M. Podsakoff, and N. P. Podsakoff, “Construct measurement and validation procedures in mis and behavioral research: Integrating new and existing techniques,” MIS Quart., vol. 35, no. 2, pp. 293–334, 2011.

Sahar Sabbaghan received the Ph.D. degree in information systems from the University of Auckland, New Zealand, in 2019. She is a Lecturer with London South Bank University, London, U.K. Her main research interest is in creating new research methodologies for information systems. She has several publications in top conference proceedings and journals.

Cecil Eng Huang Chua received the master’s degree in business by research from Nanyang Technological University, Singapore, in 2000, and the Ph.D. degree in computer information systems from Georgia State University, Atlanta, GA, USA, in 2004. He is currently an Associate Professor with the Missouri University of Science and Technology, Rolla, MO, USA. He has several publications in journals such as Information Systems Research, Journal of the AIS, MIS Quarterly, and the VLDB Journal. He is an editor for multiple journals including Information Systems Journal and Information and Management.

Lesley A. Gardner received the Ph.D. degree in information systems from The London School of Economics and Political Science (LSE), London, U.K., in 1991. She is currently an Associate Professor and the Deputy Head of Information Systems and Operations Management with the University of Auckland, Auckland, New Zealand. She has more than 120 publications in top conference proceedings and Journals. Her research areas focus on hypermedia, digital documents, and digital influences on society.