Issues in Information Systems
Volume 20, Issue 3, pp. 11-21, 2019

ANTECEDENTS TO SUCCESSFUL SOFTWARE TRAINING: A RESEARCH FRAMEWORK & PRELIMINARY EVIDENCE

Douglas Havelka, Miami University, havelkdj@miamioh.edu
T.M. Rajkumar, Miami University, rajkumtm@miamioh.edu

ABSTRACT

Training has been identified as a critical success factor for the implementation of Information Technology (IT) in organizations. A qualitative study using a group technique to gather data was performed to identify antecedents to software training in order to improve the likelihood of successful training outcomes. We find that there is a broad set of antecedents to software training that may impact whether a specific training event would be considered successful. Our results include a preliminary model of software training success that includes constructs that can be found in prior research and several that have not received much, if any, attention from researchers.

Keywords: Training, Software, System, Qualitative Research

INTRODUCTION

When organizations invest in new systems or software the training of end-users is a critical factor in the successful implementation of the system (Sharma & Yetton, 2007). The appropriate mode and type of training can be contingent on the system being implemented and the business processes (or tasks being performed) by the user. Additional contingencies that impact the effect of training on software acceptance and use have been suggested (Yi & Davis, 2003). The positive impact of training on end-user computer use, effectiveness, system acceptance, software implementation success and other dependent variables is an accepted premise for in the IT literature. The value of specific activities or conditions of the training have received less attention. The term 'training' used in this study refers to formal efforts to transfer required IS knowledge and skills to individuals using computer-based IT to perform tasks or make decisions. The basic goal of 'software training' is to teach an individual how to use a particular software application (or system). The topics of training may include information systems (IS) concepts, technical skills, organizational skills, and other knowledge about specific IS products, but the primary focus is on how software can be used to perform specific tasks. In contrast, 'education' involves learning to understand fundamental concepts or abstract theory. While both terms are relevant to this research, training activities is the primary focus.

Much of the early research in this area focused on individual differences in ability, learning styles, or other characteristics and the method used for training. For example, Nelson and Cheney tested the relationship between computer-related training and ability; and the relationship between ability and acceptance of IS products and technologies; their results support both of these relationships (Nelson & Cheney, 1987). Another line of research performed a series of studies that examine the relationship between learning style and learning performance (Bostrom, Olfman, & Sein, 1990, 1993). However, the conclusions suggested in Bostrom et al. were called into question due to the instruments used (Ruble & Stout, 1993). Another early study used an experimental investigation based on Assimilation Theory of two approaches used to train end users, in-house training programs and "user friendly" computer systems, this distinction between 'live' training versus 'software' remains a consideration today (S. Davis, Bostrom, R., 1993). A later study explores the use of web-based virtual learning environments (VLEs) for basic IT skills training. Using a longitudinal experimental design to examine one VLE design variable, learner control, they found no differences in performance by the users; however, they found that computer self-efficacy was higher and that satisfaction with the learning process was lower for VLEs (Piccoli, 2001).

Other research has tested the impact of different types of 'antecedents' or 'dimensions' of the training. One study provides some evidence that the quality of the instructor impacts the effectiveness of the training. Ahrens and Sankar explored the impact of different tutors and the teaching effectiveness of facilitators for training end users

database design principles (Ahrens & Sankar, 1993). Another examines the content of training materials, Olfman and Mandviwalla compare concept-based to procedural-based content of training materials. Concept-based training focuses on teaching the objects on an interface and the relationships between these objects. Procedural approached focus on teaching the steps to perform a specific task. The results indicate that neither is significantly related to the amount learned, but that both approaches should be taught (L. Olfman & Mandviwalla, 1995; L. Olfman, Mandviwalla, M., 1994). And another study tested a model where training was contingent upon technical complexity and task interdependence (Sharma & Yetton, 2007). These studies suggest a wide variety of factors that may impact the success of any specific training endeavor. Given these results and the lack of any research to identify the larger set of potential factors the object study was undertaken.

RESEARCH METHODOLOGY

To identify a large set of potentially important antecedents to software training success, a theory building approach based on grounded theory was used. The research approach is based on procedures and philosophical underpinnings for theory development from theory building literature, specifically grounded theory (GT) literature. The study was performed using an interpretive theory development approach using a structured group process as the data gathering technique (S. G. Sutton, Arnold, & Havelka, 1998). The coding and data analysis technique follow the prescriptions from theory building and grounded theory development (Allan, 2003; K. Eisenhardt & Graebner, 2007; K. M. Eisenhardt, 1989; Fernandez, 2004; Glaser & Strauss, 1967; Rowlands, 2005; Strauss & Corbin, 1998).

The participant-oriented perspective of grounded theory development is embodied within the purpose and outcomes of the nominal group technique for data collection (Delbecq, Van de Ven, & Gustafson, 1975; Andrew H. Van de Ven & Delbecq, 1974; A.H. Van de Ven & Delbecq, 1971), especially when used for interpretive "outside research" (G. Walsham, 1995; Geoff Walsham, 2006). Outside research is defined by Walsham as research conducted with no direct involvement in action in the field or in providing significant feedback to field participants. This method has been used successfully in several domains including systems development and auditing (Duggan, 2003; Duggan & Thachenkary, 2004; Havelka, 2002, 2003; Havelka, Sutton, & Arnold, 1998, 2001; S.G. Sutton, 1993; S. G. Sutton et al., 1998; Steve G. Sutton, Hampton, Khazanchi, & Arnold, 2008; S.G. Sutton & Lampe, 1989, 1991). The data gathering technique is described in more detail in Table 1.

Table 1. Nominal Group Technique Used

Step 1:	The facilitator made general introductions of the participants, explained the purpose of the study and the meeting, gave some definitions to be used by the subjects, explained the nominal group process to be performed, and introduced the question to be answered by the subjects. A definition of IT training was given and the question presented to the groups was stated as: "**What factors do you believe influence the efficiency, effectiveness, and quality of IT training?**"
Step 2:	Each subject was then asked to silently generate as many of these factors as possible.
Step 3:	After 15 minutes the facilitator began to write the indicators on a white board or flip charts for all the participants to view. The factors were elicited from the participants in a round-robin fashion until all the participants' factors had been listed. Only questions related to clarifying the indicators being listed were allowed at this point and no discussion of the merits or importance of the indicators was allowed. The participants were encouraged to add to their lists as this step progressed.
Step 4:	After all of the factors were listed, discussion of the factors for clarification of the items and distinction from one another was allowed. Again, discussion of the relative merits or importance of the indicators was discouraged.
Step 5:	Each participant was then asked to identify (on a worksheet) those factors generated by the group that they considered "critical" to IT training.
Step 6:	The participants are then asked to rank the factors that they identified in step 5 from most to least important.
Step 7:	The participants were also asked to fill out a questionnaire for demographic data and when possible the participants are invited to an informal venue for further debriefing and discussion.

Other business researchers have used the Delphi group technique (De Haes & Van Grembergen, 2008; Kasi, Keil, Mathieson, & Pedersen, 2008; Schmidt, Lyytinen, Keil, & Cule, 2001) to perform exploratory studies or to gather preliminary data for further testing. Nominal group techniques have been found to be superior to personal interviews and surveys when the desired goal is the generation of a maximum number of ideas or alternatives (Delbecq, Van De Ven, & Gustafson, 1982; Andrew H. Van de Ven & Delbecq, 1974; A.H. Van de Ven & Delbecq, 1971). Recently, researchers in other fields have recognized the benefits in using group oriented data collection to develop theory (Lindsay & Hubley, 2006). Following this promising approach to data collection for theory development, the nominal group technique (Delbecq et al., 1975) was used in the current study for data collection in conjunction with the coding and "constant comparative" data analysis method used in grounded theory development (Allan, 2003; Fernandez, 2004) to develop a theory of the information technology audit process.

Table 2. Participant Demographics

Demographic	**Group A**	**Group B**
Number of participants	22	26
Age range	27-53	29-52
Gender (#F/#M)	4/18	7/19
Years of work experience (range)	6-30	5-32
Number of software packages or programs used (range)	3-100+	2-30+
Number of formal software training programs (range)	0-25	0-25

Data was gathered from two groups. The groups were composed of working professionals from various functional areas and organizations, and with varying degrees of work experience and software training (all the participants were students in a part-time, professional MBA program). Summary demographics for each group are given in Table 2.

Table 3. Analysis Approach (based on Grounded Theory)

Step 1: Theoretical sampling. The structured group process (the data gathering step) is the "theoretical sampling" step and is performed as the initial step. The output of the group session is a set of items that the group identified as having influence on IT audit process quality. Each of these items is an "incident" or the atomic unit of analysis for the coding process.
Step 2: Open coding. Specific incidents or "key points" are identified. The items generated by the group were coded one-by-one as separate incidents.
Step 3: Theoretical coding. Recognition of concepts or "concept emergence" occurs. Incidents are indicators of concepts (Fernandez, 2004). Once the incidents from the group are codified they are then analyzed and compared to one another and those that relate to a common theme or that address the same semantic notion are grouped together to form a higher level abstraction, i.e. a concept. These concepts are then compared and contrasted with one another, i.e. constant comparative method, to form a higher level abstraction referred to as a category. Simultaneously, tentative relationships (potential propositions or hypotheses) between these categories are developed.
Step 4: Selective coding. Paring down or delimiting the theory being developed by identifying core categories to guide the on-going analysis. As the analysis progresses some concepts, categories, and relationships are strengthened. Other codes or concepts that are not supported by the analysis may be dropped or de-emphasized.
Step 5: Achieving saturation. The final step in the iterative grounded theory process is to substantiate the theory. To achieve this step, the researcher can look to the literature as a source of additional data (and to support the theory being proposed). As categories and relationships emerge, the extant literature can be used to inform and bolster the theoretical propositions being suggested.

The data analysis approach used is a modified version of the Glaserian Grounded Theory (GT) approach as described by Fernandez and Allen applied to data gathered using the nominal group technique (Allan, 2003; Fernandez, 2004; Glaser, 1978). The approach used is an iterative process that is presented in Table 3. It should be noted that this manuscript only reports on the initial data gathering event and the results are the beginning "bricks" or foundation of the model being developed.

RESULTS

The outcome of the data gathering sessions yielded a total of 71 indicators from both groups (33 and 38 respectively) Using the constant comparison analysis technique to first determine concepts and then categories of antecedents for software training success, a total of 30 concepts within 9 categories were identified. For formatting and discussion purposes, the results are presented in 9 separate tables below (one for each category of antecedents).

Table 4. The Time Category

Concept	Id	Indicator
Adequate time	1	Time available for training was adequate.
	40	Time allocation.
	61	Timing, adequate (demands on trainees)
Timing of Training	17	The timeliness of the training was good, it did not occur too late or too early.

Simple timing issues (Table 4) can affect the success of the training itself. Adequate time must be allocated for the training to be completed by the trainee. This is critical, because job constraints and demands might restrict the time available. The elimination or compression of training activities at the end of an IT project is a common problem when projects run over schedule. The importance of taking enough time to allow users to learn a new system has been shown to be crucial to implementation success. In addition, the training itself should be 'just in time,' i.e. the training should not occur too early (usage might be forgotten) or too late.

Table 5. The Trainer Category

Concept	Id	Indicator
Trainer Quality	10	The trainer was knowledgeable.
	20	The trainer was attractive.
	58	Quality of trainer
Type of Trainer	31	The training was conducted by a professional trainer.
	32	The training was conducted by a "super user."

Participants identified two aspects of the person conducting the training as important antecedents to training success (Table 5). The first focuses on the knowledge, professionalism, ability, preparation, or competence of the trainer, i.e. how good a trainer they are and how well they perform their duties during the training. We label this concept *Trainer Quality.* A second aspect of the trainer was identified that is based on 'who' the trainer is or what role they might have in the organization. The participants suggested that there might be a difference between a professional trainer and a 'super-user.' This implies that having a 'super-user' as the trainer may lend more credulity to the training, as a 'super-user' would be a well-respected and productive employee of the organization. This has less to do with the actual training activities and more to do with the 'change management' and 'psychological' aspects of accepting the new software. In fact, it would be expected that a professional trainer would deliver a higher quality service than a 'super-user' that has little experience as a trainer. One implication of this observation is that it would be valuable to have both a professional trainer and a 'super-user' involved with the training. This would clearly require a cost/benefit analysis to determine the marginal value. Which leads to the next category.

Participants identified many items that suggest that the success of the training depended upon 'what' they were being trained to do (Table 6). The relevance of the training to help them do their job, i.e whether it was obvious how the training and the new software would make them more productive, has a direct impact on the value of the training to the individuals. The level of detail of the training to perform specific job-related task, demonstrable productivity improvements to the job via the training, and the use of recognizable, 'real' data increases this value. Separately, the existence of incentives such as bonuses and overtime pay and 'decentives' such as penalties if training activities were not done were mentioned as motivators. The presence of influential leaders was also mentioned as a positive motivator. These items imply the importance of making the training 'useful' and communications about the value of the training and new software to the organization. Cost is included as a concept in this category to recognize that all

of the training activities should be viewed in light of the return they will have by successful implementation of the new software.

Table 6. The Value (Usefulness) Category

Concept	Id	Indicator
Relevance	45	Relevance and goal of the training
	3	The reason for the training was made clear ahead of time.
Valuable	6	I thought the training was valuable.
Productivity	11	The training made my job easier.
	12	The training was tailored to specific, relevant workflows
	26	The training provided specific examples for performing specific tasks
	51	How quickly applied
Appropriate	9	The training was at a level I could understand
	71	Level of detail appropriate
	68	Real data
Motivation	46	Incentives (to learn, to do)
	56	Mandatory
	33	"Influencers" were involved with the training.
Cost	65	Cost

If software is used to provide the training, then easy to use interfaces, functionality that supports interactions with fast responses that are fun and exciting, and use of graphics or visual aids that are annotated with text or voice annotated using screen casts, all help with providing ease of use leading to software training success. These concepts all relate to the well-established construct *Ease of Use* from the Technology Acceptance Model proposed by Davis (F. D. Davis, 1989) and are shown in Table 7 below.

Table 7. Ease of Use Category

Concept	Id	Indicator
Interface	13	The training software included an intuitive user interface
	15	The training software was easy to use
	57	Ease of use of training software
Interaction	27	The training software provided adequate response times
	35	Fun, interactive, exciting
Graphics	16	The training included appropriate visual aids
	69	Annotated screen shots
	49	Screen casting
Communication	30	The training used understandable language, not computer jargon
	42	No technical jargon (in the trainee's language)
	59	Easy to understand, explanations

Participants identified items that relate to how well the training process and activities are designed. We show these in Table 8 below. Training may be conducted via live sessions led by an instructor or via computer-based software. Based on the participants suggestions, the *Delivery Mode* (live vs software), should depend on the software that the training is for. The level of complexity and detail of the tasks the software is used to support, as well as how new or different the new software is compared to what was being used would impact whether computer-based instruction

would be sufficient or if live, human-led sessions were needed. In either case, communication and the language used play a role in how well the trainee would absorb the information being presented. If the explanations are communicated using jargon free language, then it enables the trainee to better understand. Additionally, regardless of *Delivery Mode* the training needs to be designed 'well.' Based on the participants responses this would include how the content or material is organized (is it chunked up into small digestible pieces), whether the appropriate amount of repetition is embedded (enough but not too much), and whether there were frequent assessments of progress (to test comprehension and memory). Also, several participants indicated the value of having video support for the training that allows the trainee to skip, rewind, and repeat as needed. While this is most likely a component of a software-based training experience, it was included in this category because this could be included with 'live' sessions as well.

Table 8. The Training Design Category

Concept	**Id**	**Indicator**
Delivery Mode	2	The training was "hands-on"
	7	The training was performed in one session.
	19	The training was professional, it did not include any inappropriate jokes or goofy characters
Video / Personalization	18	The training was available via video, allowing me to pause or rewind
	44	Video available, to pause and rewind
	22	The training allowed me to skip topics, content that I already knew
	38	Skippable
Organization	47	Sequential, incremental (builds upon prior training)
	53	Order of content
	70	Small bursts
Repetition	36	Frequency, repetition
	4	The training did not repeat the same material again and again.
Assessment	66	Knowledge checks
	5	There were frequent tests, quizzes, or assessments to gauge how well I was doing.

These training design concepts help convey the material in a manner that is useful to the trainee. Hence, the organization of content that sequentially builds, and chunks the information into small bursts is necessary for training success. As content may be repeated if needed, it is important to allow the trainee to view videos at their own pace, skip material, and personalize the interaction. Assessment of the knowledge by the trainee using some method of checks such as quizzes help provide feedback to the trainee as to how they are doing. Such feedback must be incorporated within the training to improve software training success.

Table 9. The Support Category

Concept	**Id**	**Indicator**
End user support	55	Mentors/super user available
	54	Human support available
Tech support	34	IT support during training
	14	The training included adequate access to competent help desk support.
Online Help	24	The training included adequate "help" information.
	50	Embedded, searchable user manual

Participants identified several items that are related to the 'support' provided during the training activities. These support items were distinct from the 'trainer' and the 'design' concepts and so were placed in a separate category. The *End User Support* is similar to the 'super-user' concept mentioned earlier, but was described as a 'on-demand' resource that could be used when and if needed, rather than as the trainer. The inclusion of adequate *Tech Support* as a factor may imply that some trainees may require basic or foundational support (e.g. how to use a mouse) whrn learning new software. Or this could simply refer to the availability of tech support if there are issues with hardware or network connections, etc. Lastly, the inclusion of *Online Help* that is adequate and easy to use to search for information also contributes to the support category.

Table 10. The Environment Category

Concept	Id	Indicator
Interruptions	21	Interruptions affected the training.
Physical Space	25	The environment for the training was good (adequate lighting, temperature, seating, etc.)
	52	Location (on site, online, distractions)
	48	Simulation space
Food/Refreshments	8	There were adequate breaks for food, etc. during the training.
	37	Hunger

Participants also identified items related to the general environment where the training takes place (Table 10). This included basic hygiene factors such as food/refreshment breaks and physical space such as the availability of adequate lighting, temperature, locations all contribute to the environment category. This implies that the 'conditions' under which the training takes place are important to allow the trainees to focus on the training activities without being distracted by work related interruptions or physical comfort. The use of off-site versus *in-situ* type of training may be relevant depending again on the type of material being covered and the importance of the training being successful. Concepts in both the support and environment category help provide the trainee with a level of comfort in the training process and removes any unnecessary distractions or dissatisfaction with the training process itself.

Table 11. The Trainee Category

Concept	Id	Indicator
Skill level	23	I had the appropriate skill level to benefit from the training.
	39	Previous training/computer experience of trainee
	60	Trainability
	62	"abled"
Variation of skill	41	Skill level (variation in the level of trainees).
	43	Diversity of the trainees (skill level, function)

From a trainee perspective, it is clear that the skill sets of the trainee, and the diversity of the skill levels of the training participants are critical (Table 11). The skill levels and computer experience that the trainee has increases their ability to be trained. Diverse skill levels of multiple participants may both hinder the level of training the trainee receives, as different individuals would require a different level of detail or instruction and may absorb information at different rates. This might slow down the pace of 'live' training to a point that some individuals are lost while others are bored. Alternatively, with increased diversity, it may be possible for the trainee to see different viewpoints and functionality of the software; thereby increasing the acceptance of the software by understanding others' needs. The use of "paired" training might be useful, i.e. where two individuals work together to learn a new skill. At times, with a variety of skill levels, the training session may fail to show advanced features of the software that the trainee might be interested in. These observations emphasize the customization of the training for the given trainee population.

With software training, both the target software (one the user is being trained to use) and the training software (the one the user is training with), contribute to the software category (Table 9) and the success of the software training. Complexity of the target software, differences with the current software being used by the organization affects the training success. Similarly, how analogous the training software is to the target software, and whether it is up to date and current helps with the training success.

Table 9. The Software Category

Concept	Id	Indicator
Currency of Training Software	29	The training software is current and up-to-date (matches the current version of the software being learned).
	67	Current
Target Software	28	The new software is significantly different than the old.
	63	Complexity of system/software being taught
Similarity	64	Analogous systems

DISCUSSION

Our aim in this study is to identify a broader set of antecedents to software development training that could be used by researchers and practitioners to improve the outcomes (success) of training events. Our approach produces categories and concepts that have been identified by working professionals as factors that would impact the efficiency, effectiveness, and quality of software training. These concepts are not necessarily independent of one another and may show complex interactions between them. Based on the results of our analysis, we have developed a preliminary model of software training success that shows potential inter-relationships of the categories identified (see Figure 1).

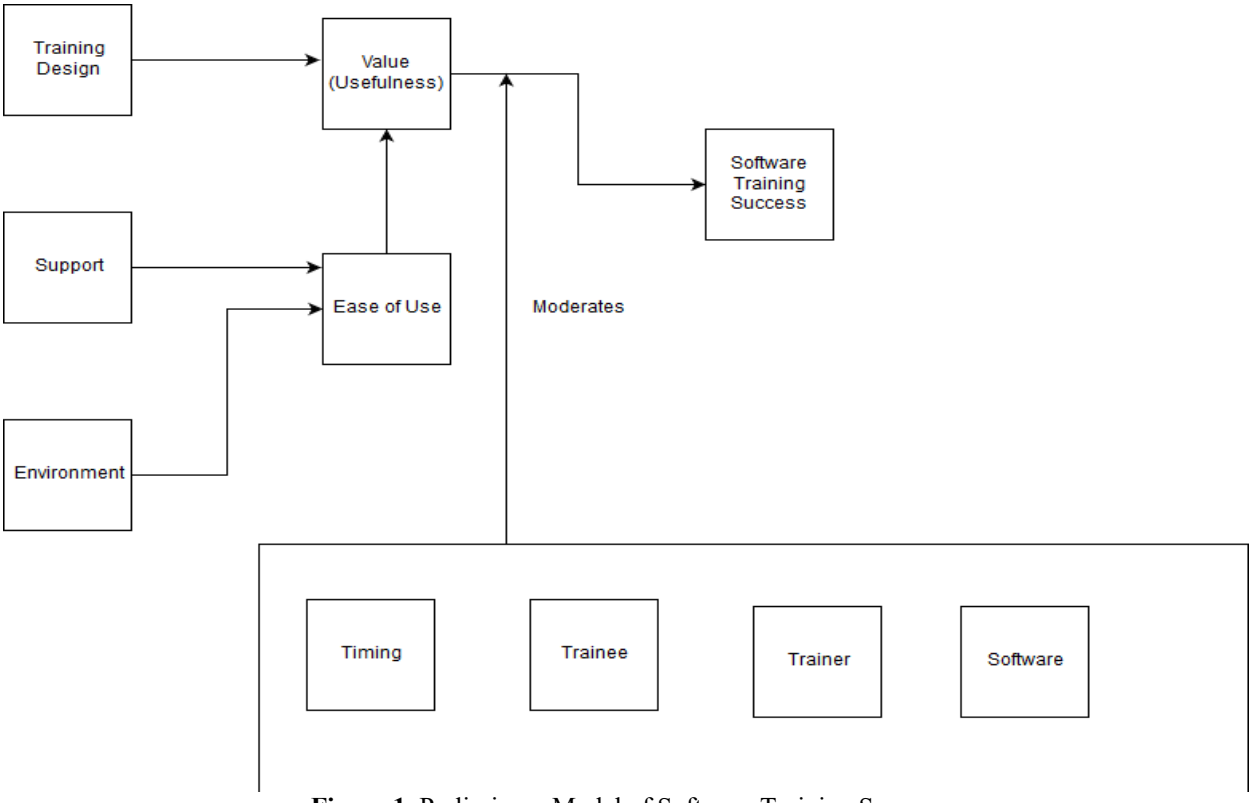

Figure 1. Preliminary Model of Software Training Success

We identify nine categories that are important for software training success. A recent study on technology mediated learning by Söllner et al. identifies about sixteen concepts (Sollner, Leimeister, Bitzer, & Janson, 2018). Some of their identified concepts: Trainer quality, learning materials, IT systems quality, interactivity, learning environment, IT support, organizational support, and intrinsic value have significant overlaps with the concepts found in our study. However, we note that there are several different concepts from our results that were not included in any prior research. We also note that the Technology Acceptance Model (TAM) has shown that both ease of use (EOU) and usefulness (USE) of the software leads to acceptance and usage of the software (F. D. Davis, 1989). Recognizing this well-established framework, we have incorporated the EOU and USE constructs in our model (Tables 3 & 4). We originally conceptualized these categories as value or content (usefulness) and training design (ease of use).

Identification of the antecedents to successful software training is just the first step in the larger process of creating best practices for software training activities. Additional research is needed to determine which of the antecedents identified would be most effective when considering cost and other constraints.

REFERENCES

Ahrens, J. D., & Sankar, C. S. (1993). Tailoring Database Training for End Users. *MIS Quarterly, 17*(4), 419-439.

Allan, G. (2003). A critique of using grounded theory as a research method. *Electronic Journal of Business Research Methods, 2*(1), 1-10.

Bostrom, R. P., Olfman, L., & Sein, M. K. (1990). The Importance of Learning Style in End-User Training. *MIS Quarterly, 14*(1 (March)), 101-119.

Bostrom, R. P., Olfman, L., & Sein, M. K. (1993). Learning Styles and End-User Training: A First Step. *MIS Quarterly, 17*(1 (March)), 118-120.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. *MIS Quarterly, 13*(3 (September)), 319-340.

Davis, S., Bostrom, R. (1993). Training End Users: An Experimental Investigation of the Roles of the Computer Interface and Training Methods. *MIS Quarterly, 17*(1), 61-86.

De Haes, S., & Van Grembergen, W. (2008). An exploratory study into the design of an IT governance minimum baseline through Delphi research. *Communications of the Association for Information Systems, 22*(24), 443-458.

Delbecq, A. L., Van de Ven, A. H., & Gustafson, D. H. (1975). *Group Techniques for Program Planning: A guide to nominal group and delphi processes.* Glenview, IL: Scott-Foresman.

Delbecq, A. L., Van De Ven, A. H., & Gustafson, D. H. (1982). Guidelines for conducting NGT meetings. In D. R. Hampton, C. E. Summer, & R. A. Webber (Eds.), *Organizational behavior and the practice of management.* (Fourth ed., pp. 279-298). Glenview, IL: Scott, Foresman, and Company.

Duggan, E. W. (2003). Generating systems requirements with facilitated group techniques. *Human-Computer Interaction, 18*, 373-394.

Duggan, E. W., & Thachenkary, C. S. (2004). Integrating nominal group technique and joint application development for improved systems requirements determination. *Information & Management, 41*, 399-411.

Eisenhardt, K., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal, 50*(1), 25-32.

Eisenhardt, K. M. (1989). Building Theories From Case Study Research. *Academy of Management Review, 14*(4), 532-550.

Fernandez, W. D. (2004). Using the Glaserian approach in grounded studies of emerging business practices. *Electronic Journal of Business Research Methods, 2*(2), 83-94.

Glaser, B. (1978). *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory*. Mill Valley: The Sociology Press.

Glaser, B., & Strauss, A. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago: Aldine Publishing Company.

Havelka, D. (2002). Requirements determination: An information systems specialist perspective of process quality. *Requirements Engineering, 6*(4), 220-236.

Havelka, D. (2003). A user-oriented model of factors that affect information requirements determination process quality. *Information Resources Management Journal, 16*(4), 15-32.

Havelka, D., Sutton, S. G., & Arnold, V. (1998). A methodology for developing measurement criteria for assurance services: An application in information systems assurance. *Auditing: A Journal of Practice & Theory, 17*(Supplement), 73-92.

Havelka, D., Sutton, S. G., & Arnold, V. (2001). Information systems quality assurance: The effect of users' experiences on quality factor perceptions. *Review of Business Information Systems, 5*(2), 49-62.

Kasi, V., Keil, M., Mathieson, L., & Pedersen, K. (2008). The post mortem paradox: a Delphi study of IT specialists perceptions. *European Journal of Information Systems, 17*, 62-78.

Lindsay, A. C., & Hubley, A. M. (2006). Conceptual reconstruction through a modified focus group methodology. *Social Indicators Research, 79*, 437-454.

Nelson, R. R., & Cheney, P. H. (1987). Training End Users: An Exploratory Study. *MIS Ouarterly*, 547-559.

Olfman, L., & Mandviwalla, M. (1995). An Experimental-Analysis of End-User Software Training Manuals. *Information Systems Journal, 5*(1), 19-36.

Olfman, L., Mandviwalla, M. (1994). Conceptual Versus Procedural Software Training for Graphical User Interfaces: A Longitudinal Field Experiment. *MIS Quarterly, 18*(4), 405-426.

Piccoli, G., Ahmad, R., Ives, B. (2001). Web-Base Virtual Learning Environments: A Research Framework and a Preliminary Assessment of Effectiveness in Basic IT Skills Training. *MIS Quarterly, 25*(4), 401-426.

Rowlands, B. H. (2005). Grounded in practice: Using interpretive research to build theory. *The Electronic Journal of Business Research Methodology, 3*(1), 81-92.

Ruble, T. L., & Stout, D. E. (1993). Learning Styles and End-User Training: An Unwarranted Leap of Faith. *MIS Quarterly, 17*(1 (March)), 115-118.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems, 17*(4), 5-36.

Sharma, R., & Yetton, P. (2007). The contingent effects of training, technical complexity, and task interdependence on successful information systems implementation. *MIS Ouarterly, 31*(2), 219-238.

Sollner, M., Leimeister, J. M., Bitzer, P., & Janson, A. (2018). Process is king: Evaluating the performance of technology-mediated learning in vocational software training. *Journal of Infomation Technology, 33*(3), 233-253.

Strauss, A., & Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (Second ed.). Thousand Oaks: Sage Publications, Inc.

Sutton, S. G. (1993). Toward an understanding of the factors affecting audit quality. *Decision Sciences, 24*(1), 88-105.

Sutton, S. G., Arnold, V., & Havelka, D. (1998). A methodology for developing measurement criteria for assurance services: An application in information systems assurance - Reply. *Auditing: A Journal of Practice & Theory, 17*, 103-107.

Sutton, S. G., Hampton, C., Khazanchi, D., & Arnold, V. (2008). Risk analysis in extended enterprise environments: Identification of critical risk factors in B2B e-commerce relationships. *Journal of the Association for Information Systems, 9*(3/4), 151-174.

Sutton, S. G., & Lampe, J. C. (1989). Monitoring Quality in a Knowledge Work Environment: Integrating Structured Group Processes With Decision Support System Design. *Productivity Management Frontiers, 2,* 365-372.

Sutton, S. G., & Lampe, J. C. (1991). A Framework for Evaluating Process Quality for Audit Engagements. *Accounting and Business Research,* 275-288.

Van de Ven, A. H., & Delbecq, A. L. (1974). The effectiveness of nominal, delphi, and interacting group decision making processes. *Academy of Management Journal, 17*(4), 605-621.

Van de Ven, A. H., & Delbecq, A. R. (1971). Nominal versus interacting group processes for committee decision making effectiveness. *Academy of Management Journal, 14*(2), 203-212.

Walsham, G. (1995). The Emergence of Interpretivism in IS Research. *Information Systems Research, 6*(4), 376-394.

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems, 15*(3), 320-330.

Yi, M. Y., & Davis, F. D. (2003). Developing and Validating an Observational Learning Model of Computer Software Training and Skills Acquisition. *Information Systems Research, 14*(2), 146-169.