International Review of Research in Open and Distributed Learning

IRRODL | THE INTERNATIONAL REVIEW OF RESEARCH IN OPEN AND DISTRIBUTED LEARNING

Blended Training on Scientific Software: A Study on How Scientific Data are Generated

Efrosyni-Maria Skordaki et Susan Bainbridge

Citer cet article

Skordaki, E.-M. & Bainbridge, S. (2018). Blended Training on Scientific Software: A Study on How Scientific Data are Generated. *International Review of Research in Open and Distributed Learning*, *19*(2). https://doi.org/10.19173/irrodl.v19i2.3353

Résumé de l'article

This paper presents the results of a research study on scientific software training in blended learning environments. The investigation focused on training approaches followed by scientific software users whose goal is the reliable application of such software. A key issue in current literature is the requirement for a theory-substantiated training framework that will support knowledge sharing among scientific software users. This study followed a grounded theory research design in a qualitative methodology. Snowball sampling as well as purposive sampling methods were employed. Input from respondents with diverse education and experience was collected and analyzed with constant comparative analysis. The scientific software training cycle that results from this research encapsulates specific aptitudes and strategies that affect the users' in-depth understanding and professional growth regarding scientific software applications. The findings of this study indicate the importance of three key themes in designing training methods for successful application of scientific software: (a) responsibility in comprehension; (b) discipline; and (c) ability to adapt.

Blended Training on Scientific Software: A Study on How Scientific Data are Generated

Efrosyni-Maria Skordaki[1] and **Susan Bainbridge** [2]
[1]*Royal Military College of Canada, Ontario, Canada,* [2]*Athabasca University, Alberta, Canada*

Abstract

This paper presents the results of a research study on scientific software training in blended learning environments. The investigation focused on training approaches followed by scientific software users whose goal is the reliable application of such software. A key issue in current literature is the requirement for a theory-substantiated training framework that will support knowledge sharing among scientific software users. This study followed a *grounded theory* research design in a qualitative methodology. Snowball sampling as well as purposive sampling methods were employed. Input from respondents with diverse education and experience was collected and analyzed with constant comparative analysis. The *scientific software training cycle* that results from this research encapsulates specific aptitudes and strategies that affect the users' in-depth understanding and professional growth regarding scientific software applications. The findings of this study indicate the importance of three key themes in designing training methods for successful application of scientific software: (a) *responsibility in comprehension*; (b) *discipline;* and (c) *ability to adapt*.

Keywords: blended learning, grounded theory, scientific software, training, distance learning, snowball sampling, purposive sampling

Introduction

Scientific Software Training

Scientific software is becoming increasingly important to the realms of science and engineering. It is a tool that is used to process data and solve models expressed mathematically in an augmented, timelier manner. Scientific software is employed in research areas that can directly affect public safety, such as nuclear power generation computer systems, groundwater quality monitoring and engineering designs. Academic researchers and industry professionals depend on such software in order to answer their scientific inquiries. Scientific software also provides infinite opportunities to share and collaborate. Howison and Herbsleb (2011) argue that the creation of new scientific knowledge requires the combination of evolving scientific methods, validated instruments, and theory. However, the value of training scientists and engineers on this type of software is underestimated. Literature has already acknowledged a general lack of formal scientific software training among users, especially for large research projects with societal importance. An overwhelming majority of researchers in natural sciences and engineering wish for increased computational skills, as they need to have sufficient knowledge of what the software is doing and whether it is, in fact, doing what is expected (Hannay et al., 2009; Joppa et al., 2013; Skordaki, 2016). As society's important scientific decisions rely on accurate scientific software application, "the scientific community must ensure that the findings and recommendations put forth based on software models conform to the highest scientific expectation" (Joppa et al., 2013, p. 815). Thus, users who study global climate change or the migration of contaminants in the environment ought to be able to trust and confirm the software output in order to publish the data and inform the public responsibly. However, despite the abundance of training literature, there is limited research that looks at successful strategies to train professionals specifically on the use of scientific software.

Scientific software continues to advance as knowledge obtained through continued scientific endeavor progresses. As the cognitive density of the software increases, so does the risk of incorrect use of the software or insufficient validation of the software output by the user. Managing the risk of making errors in scientific software output interpretation and application is critical (Fischer, 2009; Hannay et al., 2009). Risk is here defined as the likelihood of unintended mistaken scientific and engineering decisions based on the incorrect use and/or misinterpretation of data output from the scientific software tool (Skordaki, 2016). With the development of new scientific software products, the issue that becomes central for the users is the effectual comprehension of the knowledge that is entrenched in the software—that is, its capabilities and limitations, and how these can affect the software output. Obtaining an in-depth understanding of the software product can enhance the accuracy and reliability of its application (Holton, 2004; Segal, 2005, 2007; Sloan, Macaulay, Forbes, Loynton, & Gregor, 2009; Fischer, 2009, 2011; Adams, Davies, Collins, & Rogers, 2010). Thus, the problem is that there is a growing need for the identification of a good framework for scientific software training that can help mitigate risk in its applications or interpretations, but there is insufficient literature regarding this topic.

As such, effective training on the use of scientific software is essential in order to ensure correct scientific decisions. Training here does not mean learning repeated, predetermined tasks, but rather it refers to establishing an effective collaborative learning environment that can ensure successful problem solving using the appropriate scientific software. This research adopted the definition of training by Dearden (1984)

as this definition provides a framework for investigating how dealing with things (in this case, dealing with scientific software), people (adult learners-scientific software users), and change (traditional and distance learning, effective collaborative learning) can influence the learning process within the community of scientific software users. Dearden (1984) gives a holistic definition of training and links it to learning: "But in every case what is aimed at is improved level of performance ... brought about by learning" (p. 58-59).

This paper presents the results of a study on scientific software training in *blended learning* environments. Blended learning is considered here as the thoughtful integration of classroom face-to-face learning experiences with online learning experiences, as in Garrison and Kanuka (2004). Blended learning environments can afford opportunities for multiple forms of communications that can stimulate open dialogue, critical debate, negotiation and agreement (Garrison & Kanuka, 2004; Graham, Henrie, & Gibbons, 2014; Skordaki & Bainbridge, 2015a). This can be particularly important for open communities of scientific software users who may rely on peer collaboration for obtaining feedback on scholarly work.

The investigation focused on training approaches followed by scientific software users whose goal is the reliable application of such software. The *scientific software training cycle* that stems from the findings of this research is presented in the "Results" section. This training cycle encapsulates specific aptitudes and strategies that affect the comprehension and professional growth of scientific software users.

Scope of Research

The research looked into the needs of scientific software users as learners in their setting of practice. It examined the interactions of users with their professional environment, in traditional and blended learning settings. The main research question was: What software training approaches in a blended learning environment are chosen by users whose goal is to accurately apply scientific software to questions of research?

Methodology

This study followed a *grounded theory* research design in a qualitative methodology, as in Skordaki (2016). Grounded theory was selected because scientific software training is a field with limited existing research (Hannay et al. 2009; Howison & Herbsleb, 2011; Skordaki & Bainbridge, 2015b; Skordaki, 2016). The strongest cases for using grounded theory are in studies of comparatively unexplored areas (Corbin & Strauss, 1990; Strauss & Corbin, 1998; Moghaddam, 2006). As well, grounded theory was employed with a view to investigating scientific software training as a phenomenon in its natural context, without preconceived notions. This qualitative investigation did not seek to test a particular hypothesis on scientific software training or the learning needs of users; it aimed to explore this field of interest (Hoepfl, 1997). As Harrison (2015) pointed out, grounded theory methods allow "the researcher to 'listen to' the experiences of the participants as expressed freely without constraining prompts." The data collection continued until theoretical saturation was reached (Glaser & Strauss, 1999). In this investigation, observations in real-life settings, acquisition of insider accounts, and collection of empirical data in their naturalistic setting were used from which themes were identified and conclusions extracted. The analysis of the empirical findings

led to the construct of the scientific software training framework that is presented in this paper. As such, the emergent scientific software training concepts were grounded in the amassed data.

An eclectic foundation was created with elements that were drawn from previous studies. The studies that were pertinent to this investigation included: (a) qualitative research in software engineering by Lutters and Seaman (2007), and Robinson, Segal, and Sharp (2007); (b) research on adult training on computer use by Lowe (2004) and Hurt (2007); (c) research on hybrid learning experiences in geological sciences by Adams et al. (2010); and (d) research on distance learning in engineering programs by Bissell and Endean (2007).

Research Data Collection and Analysis

The recruitment of participants and the data collection took place in universities and industry sectors in Canada. Twenty female and male scientific software users were interviewed; the ratio was determined only by the availability and willingness of each respondent to participate in the study. The age of the participants varied from 20 to over 55 years old. The criteria for identifying the research study participants are listed below, as they were listed in the research ethics approval obtained for this study. Primary and secondary data were utilised in order to attempt to answer the research goal. The primary data were derived from unstructured, open-ended interviews with study participants in a familiar setting where the participants would not feel restricted or uncomfortable to share information (Miles & Huberman, 1994; Cohen, Manion, & Morrison, 2007). The interviews were approximately one to two hours long. They involved narrations of software applications and pertinent issues introduced by the users. Demonstrations of scientific software usage, reviews, and critiques of laboratory manuals by the users were also included. The informal, conversational nature of the interview technique offered substantial control to the interviewees (Turner, 2010). Hand-written notes were kept during the conversations with the interviewees throughout the data collection. Digital voice recordings were also kept when the interviewees consented to them. The interviews yielded 23 to 52 pages each, depending on their duration. Transcribing the audio recording of each interview took, on average, eight hours.

The secondary data were obtained from observations of scientific software users during training sessions, as well as reviews of published documents and literature that were relevant to the scope of this study. The latter included: (i) software training manuals; (ii) laboratory notes kept by students; (iii) journals kept by researchers; and (iv) course materials.

Study participants were identified according to the following criteria: (i) experience with scientific software usage; (ii) experience with academic research involving the usage of scientific software in science and engineering; and (iii) experience with industry applications regarding scientific software usage in science and engineering. Specifically, the recruitment of respondents for this study followed a two-layered strategy. Firstly, snowball sampling was employed in order to establish an initial pool of potential research participants. This sampling technique is used for identifying research respondents where one respondent gives the researcher the name of another potential respondent, who in turn provides the name of a third, and so on (Strauss & Corbin, 1998). Snowball sampling was suitable for the purposes of this investigation, which had an exploratory, ethnographic nature (Berg, 2001; Atkinson & Flint, 2001). Snowball sampling is also used to overcome the problems associated with studying concealed or hard-to-reach populations; these

groups—are usually small relative to the general population, and no exhaustive list of population members is available for them (Berg, 2001). The community of scientific software users has been described in the literature as being protective of their research ideas, laboratory practices and publication domains due to their competitive work environment in terms of publication records and funding resources. Thus, it can be considered a hard-to-reach population (Hannay et al., 2009).

As the data collection progressed, new knowledge-rich respondents with substantial experience in scientific software application were sought; as such, purposive sampling was employed after the initial data analysis and formation of categories in order to expand on or inform the initial data gathering. Due to the limited number of available respondents for this phase of the study, purposive sampling was employed in lieu of theoretical sampling (which normally requires larger sets of data) during this data collection phase. Previous research studies have successfully combined purposive sampling with a grounded theory design in order to collect rich, yet sufficient data for theoretical adequacy (Frazier, 2006; Bainbridge, 2013). The primary concern regarding employing purposive sampling is to obtain rich information from those who are in a position to provide it (Miles & Huberman, 1994; Cohen et al., 2007).

Given that this was an exploratory qualitative study, the data collection ceased when it was decided that the richness of the information gathered could support the formation of core categories and themes. In order to sufficiently ground the results of this study in the research context and as suggested by Cohen et al. (2007), the researcher ensured theoretical adequacy and ability to check emerging themes with further data by maintaining access to the participants and their information throughout the research study, in case further information needed to be collected.

The data analysis commenced with basic listening of the recordings as well as carefully reading the interview notes and memos in order to delve into the data and see if patterns emerge. The data analysis commenced during the data collection phase. An open and axial coding system of the information amassed during the interviews was employed. Documentation of this coding and the writing of memos functioned as an audit trail back to the sources of the research results and conclusions (Frazier, 2006), enhancing the dependability of this coding system and giving the authors the ability to confirm the research design. In grounded theory, open coding allows for both the exploration of the data collected and the identification of units of analysis in order to code for specific issues; axial coding allows for categories and themes to be interconnected (Corbin & Strauss, 1990; Saldana, 2008). Emerging trends and/or patterns in the data were identified and further illustrated in diagrams and matrices. Microsoft Office Excel spreadsheets were used for the organization of the data. After the core categories emerged from the information provided by the participants, selective coding was used to identify overarching themes in scientific software training (Corbin & Strauss, 1990; Saldana, 2008).

Two examples of the data analysis processes of this study are included in Tables 1 and 2, with excerpts from the interview data. Table 1 refers to issues associated with onsite learning. During one of the interviews, the participant was invited to describe his experiences with respect to learning from his mentors or supervisors in university or industry. The interviewee, a graduate student, responded:

> Every lab I have seen is different. Where I worked before, the supervisor sat there with me and showed me the basics. It is easy to ask questions when somebody is near. Some people might be

shy, I just ask my lab mates, but if you cannot find the answer on your own, the best way is to find someone who knows.

Table 1

Data Analysis Example: Onsite Training

Text excerpts	Analysis: Elements of onsite training
"Every lab I have seen is different."	Emphasis on the environment in which the learner operates
"…it is easy to ask questions when somebody is near."	Learning from one another via instruction, imitation and modeling
"Where I worked before, the supervisor sat there with me and showed me the basics."	Relationship of the learner with the Mentor—positive learning experience
"Some people might be shy, I ask my lab mates…"	Motivation of learner—enhanced due to the social aspect of the learning process

Table 2 includes pieces of text that refer to the learning skills of the user and provides an example of the open coding process that was conducted during the initial stages of the data analysis. To the interviewer's question, "How do you see your role as a mentor in this lab?" the interviewee responded:

I feel it is mostly up to them—if they are motivated, they will learn. And if they want to do some research … they ask, they know that they are in the driver's seat, they want to know.

Table 2

Data Analysis Example: User Learning Skills

Text excerpts	Analysis: Open coding
"…if they are motivated, they will learn."	Goal oriented learner
"But if they want to do some research… they ask, they want to know."	Internally motivated and self-directed learner
"…they are in the driver's seat."	Self-directed learning
"If they are serious about it…"	Internally motivated and self-directed learner

Figure 1, below, shows an example of a coding scheme in this study. The first categories that were created resulted from the axial coding process and had to do with topics related to the background and academic preparation of the user (e.g., type of knowledge, ability of the user to comprehend the research problem at hand) and their training needs. The second categories were developed during the selective coding process.
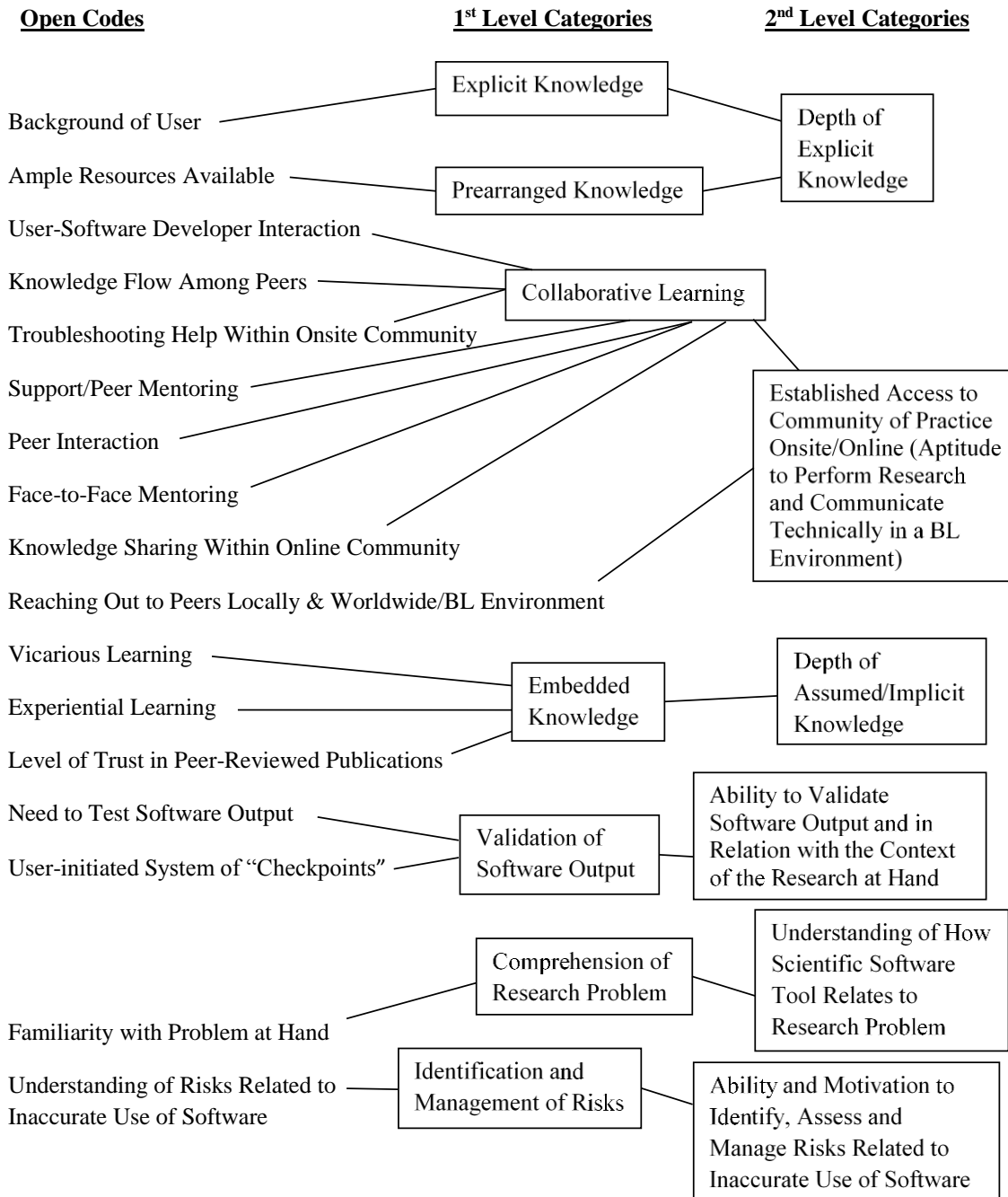

Figure 1. Example of coding scheme (Adapted from *Mobilizing knowledge in science and engineering: Blended training for scientific software users* (p. 66), by E. M. Skordaki, 2016, Athabasca, AB: Athabasca University. Copyright 2016 by Efrosyni-Maria Skordaki).

This study employed a key feature of grounded theory, the constant comparative method, in order to identify categories and relationships that exist among the data and generate concepts. The constant comparative method requires that the researcher compares each piece of data with data previously analyzed in all groups that have emerged (Glaser & Strauss, 1999; Cohen, et al., 2007; Frazier, 2006; Bainbridge, 2013).

Reliability and Validity Strategies During Data Collection

Ensuring rigor in a qualitative research study during its development is crucial (Morse, Barrett, Mayan, Olson, & Spiers, 2002). As such, the research design of this study incorporated strategies for reliability and validity checks during the course of the investigation. It is noted here that, due to very limited literature on scientific software training methods, the author obtained theoretical background knowledge during the research, which increased the credibility of the study (Miles & Huberman, 1994). Glaser and Strauss (1999) indicate that there is no need to review any literature of the studied area before entering the field, and this is in line with this research.

A first strategy used to achieve reliability and validity for this investigation was the ongoing analysis of the data collected, which subsequently directed the research path and specifically the application of the purposive sampling technique (selection of participants and interview themes/questions). Categorizing and comprehending (or "listening to") the data can influence the course of the investigation and enhance the quality of the research as well as its replication and confirmation (Glaser & Strauss, 1999). A second reliability and validity strategy was to interview the study's researcher prior to the commencement of the data collection; in this manner, the researcher confronted personal opinions and preconceptions about software training and could clearly compare these with the views of the actual study participants. A third strategy ensured the validity of the data collected by employing respondent validation; in this technique, a comparison was made between the accounts of different participants working in the same lab or work environment but in different roles. The emerging themes from the coding of the data were tested accordingly (Rajendran, 2001; Cohen et al., 2007). Additionally, reflective journaling ("memoing") was employed during the data analysis in order to accurately depict different realities and levels of understanding of the data collected. Reflective journaling was also used throughout the study as a tool to record ideas about the emerging themes and relationships between categories.

Further, the ongoing development of sensitivity and flexibility of the researcher with respect to the emerging themes from the data collected was also an important parameter in the study, as such development can enhance the verification process during an investigation (Berg, 2001).

Results

This section presents the core categories of the study as these emerged from the analysis of the amassed data in this investigation. These categories are:

1. Current training practices adopted by the community of scientific software users, and;

2. Main parameters that affect current scientific software training, as reported by study participants.

Current Training Practices

In total, the study respondents offered reviews of 13 scientific software products in terms of their training tools. Four of the software companies provided the scientific software manuals upon purchase of the product. Nine of them offered online video tutorials, demonstrations, and resources in addition to the product manual. Six of them offered in-classroom training to users and a suite of solved, generic examples in their online libraries. Two of them had research publications that used their software product available on their website. Each software product had specific strengths and limitations that could be revealed, according to the participants, only via the application of the software in particular research problems. The online resources and the manuals offered by the software companies were discussed during the interviews with the respondents with respect to their usefulness and effectiveness f updated information.

It is noted here that the respondents included users who had the ability to write code and build a numerical model as well as users who simply applied the software without interfering with its code. Various methods of training were mentioned by the participants. A major observation from the data is illustrated in the following statement made by a participant: "Understanding is more important than ease of use."

Regarding current training practices, the results illustrated that there were users who were self-taught by primarily using software company website documentation and other relevant online asynchronous resources without receiving significant online/onsite peer support and feedback. There were also users who had the opportunity to sit side-by-side with a more experienced colleague and learn the basics of particular software products in order to expand their abilities as users before they explored these products by themselves. In addition, there were users who were introduced to the basic principles of a software product by their work supervisor before they were expected to work with it on their own. These users reported that they often felt intimidated to ask questions or seek clarifications because of the experience gap between themselves and their supervisor/mentor. Lastly, there were users who reported that they learned their software in a collaborative learning environment with peers at similar levels of experience and felt comfortable asking questions about the software freely. Table 3, below, depicts the various training techniques employed by scientific software users as described by the research participants. In the table, an onsite mentor refers to someone with substantial experience on scientific software products and with whom the software trainee can consult in their physical work environment. An online mentor can be a scientific software developer who offers support and advice to a new user. An onsite peer can be a colleague or fellow graduate student with equal or slightly higher experience in scientific software who works in the same physical environment with the trainee and they frequently exchange feedback on the application of the software. An online peer can be a software user who contributes to an online forum about a particular scientific software product.

Table 3

Current Training Techniques Followed by Scientific Software Users

Training technique	Participation rate	Advantages	Disadvantages	Comments/Quotes from users
Onsite tutorials	4 of 20	Proximity	Expensive; "packaged" lecture (i.e., generic)	Topics presented are not shared with participants prior to the tutorial presentations; "participants cannot prepare beforehand."
Onsite documents	20 of 20	Availability	Outdated	These may involve reiteration of assumed knowledge; "documents are prepared by experienced users who may not see the gaps in knowledge flow" (i.e., tacit knowledge) from the junior users' perspective.
Online documents (including Wikis)	15 of 20	Current information; flexible access	Issues with trust and reliability of information sources	Online documents are occasionally "difficult to understand" if substantial time is not spent previously by the junior user troubleshooting alone or searching in manuals.
Online tutorials	17 of 20	Free; updated	Inflexible; generic material	Online tutorials are often "impersonal and not applicable" to users' specific questions; they give good demonstrations on basic use of software tool.
Onsite mentor	8 of 20	Proximity	Intimidation	Lack of mentors' formal preparation on giving constructive feedback techniques and creating a positive learning environment / "not intimidating."

Onsite peer	18 of 20	Immediacy; lower anxiety levels	Propagation of wrong information	This can encourage the development of teamwork skills but can also lead in "new users relying on more experienced peers" instead of expanding their own knowledge base.
Online mentor	6 of 20	Different perspective; flexibility in communica-tion	Trust issues; mentor feels removed from trainee (i.e., impersonal)	The protection of "ownership of ideas limits online knowledge sharing and critical review of work." Still, the user may benefit from being exposed to the viewpoint of an online mentor; this viewpoint may be different from that of an onsite peer or mentor.
Online peer	9 of 20	Flexible access	Reliability of exchanged information; trust issues	If the exchanged feedback is reliable and trust in the peer collaboration is established, then the "validity of the research results is enhanced by constructive input from peers with no invested interests."

As seen in Table 3 above, onsite tutorials usually offered by software companies have not been a preferred mode of training due to cost considerations; as well, the "packaged" training material is offered by software developers who may not be aware of the various specific applications of their software product. As Participant 02 stated: "I have not really experienced formal training. It is expensive. The software company has seminars, to promote their software, sometimes I find these seminars online, but it is not as if you had someone talking with you about your specific questions."

Six respondents mentioned that if the tutorial materials had become available to them prior to the onsite presentation, then perhaps they would have been able to review it and develop their questions in order to make better use of the time with the company trainer. As Participant 15 mentioned: "We were not prepared. They gave us a generic presentation on what the software does but there was not enough time for all of us to ask questions on particular problems with the software."

The online documentation that was discussed by the study participants included information and support tools available on software companies' websites that were available to users for troubleshooting purposes. The respondents also mentioned wikis that were developed by other users and were available on the

Internet. From the analysis of the references made by the respondents, it appears that the onsite peer support is crucial at the beginning stages of the training. As training progresses, the user may explore the online resources more independently and with a higher ability to critique the reliability of the information. While an onsite mentor can be useful to a new scientific software user in the sense that he/she can effectively direct the trainee, study participants commented on having high anxiety and intimidation levels because often the mentor was also their evaluator (e.g., work supervisor or university faculty). Participant 03 (a graduate student) mentioned: "If you are nervous, if you feel that you do not understand the problem enough to ask a question… if you work in a place where everyone is a senior software developer and they do not have time to answer questions… it is difficult." Another respondent, Participant 14 (a laboratory supervisor) said: "Some people look for someone to ask, some others spend a week before they build the courage to come to us… They should not need encouragement because we told them so (to come to ask).

Through onsite peer support, new users felt more at ease asking questions, showing their work in progress and learning in a relaxed environment. Participant 02 mentioned: "For approximately one week a friend showed me the basics." Participant 14 stated: "My group is pretty big…Very friendly people…If I talk to them, it will take me two minutes…If I keep looking for the answer myself it will take me two weeks."

The Main Parameters of Current Scientific Software Training

Three main parameters are involved in scientific software training: (a) the personal skills of the user, and the (b) onsite, and (c) online training environments.

Personal Skills of Users

The usefulness and successful outcome of the training environment is impacted by the profiles of the users. Specifically, the training is affected by:

1. The individual learning preferences of the users;

2. The degree of responsibility and knowledge-building goals of the users; and

3. The undergraduate preparation of the users.

Onsite/Online Training Environments

The work/training environment of the users is affected by:

1. Mentorship and instructional considerations;

2. Collaborative learning activities; and

3. Challenging assumed knowledge.

The results indicate that the issue of informal learning with peers online/onsite was an important ingredient in the training process on scientific software. Participant 02, a Master's student, stated: "The supervisor sat beside me for a week, gave me an introduction to their software and then I figured things out myself." Participant 04, a Master of Science student close to his graduation, also indicated that "if you stumble on something, go ask someone, it is much faster… There is also an online community that shares ideas, we help each other, it speeds up the process." Participant 09, a doctoral student, pointed out: "With my lab mates I feel a lot more comfortable asking questions than if you work with a senior software developer, because they may not have time to answer questions at your level." Participant 06, a master of science student with some experience in computing, also added: "I have not really experienced formal training like industry seminars. It is expensive…. After you learn the basics, there is an online community of users that you can go to."

It is important to note that the online environment is intertwined with onsite peer support throughout the training of all interviewees. The majority of the respondents used both modes of knowledge transfer during their typical work day. A prevailing observation from the interviews was that online documentation (from software company sites) can complement onsite laboratory resources (manuals, short courses) and support the needs of the users regarding expanding their knowledge. As Participant 17, a user with over 9 years of experience in scientific software applications, stated: "The students socialize online with a common issue; the exposure that they have to their professors is minimal to the one they get through online means. But they need to know how to filter the information."

However, 17 out of 20 participants suggested that posting questions in online discussion sites can wait until the users have confidence in their ability to critically review feedback from online peers. Also, the majority of new users that were interviewed for this study claimed that lack of proper terminology, at the beginning of their training, hindered their ability to use the online community as a resource as they would not be able to accurately articulate technical questions. Participant 20, a researcher working for the government, commented: "Pretty pics are posted now and you think you go somewhere but perhaps you may have gotten nowhere. The pace of research today moves too fast." "Understanding is more important than ease of use," mentioned Participant 15, a physicist with over 30 years of experience in scientific software.

The Blended Learning Environment

Figure 2 depicts the results of the data analysis regarding the efficacy of the blended learning environment in scientific software training. Parameters (1) to (9), listed in Figure 2 below, are linked with pieces of text from the interview data. All interviewees commented on the use of the Internet as a resource for their knowledge expansion in addition to onsite laboratory training sessions and onsite peer support, which was considered the starting point in scientific software training for most users—parameters (1) and (2) in Figure 2. There were 15 out of 20 participants who mentioned issues of trust, familiarity with technology, and not feeling sure about how to critically analyze the vast information on the Web; in Figure 2 these are referred to as issues that affected their independence as learners (4) in seeking specialized knowledge online. Academic preparation (5) regarding formal scientific software training was considered insufficient by all of the interviewees: "In my undergrad, we did not do much…We had one lab in third year…I learned most of what I needed after I graduated, I worked in an engineering firm for a while…then came back to do my master's." The interviewees who were the most confident in using online discussion forums to cross-check

and test their ideas with online peers were the ones who demonstrated a solid background on their subject—parameters (8) and (9) in Figure 2—saying, for example, that "the open community we work in, you can submit your data so that they can simulate it, you can see what the reviewers have said." Motivation in sharing data and receiving feedback from online peers indicates a drive to seek further knowledge (3) about a particular problem and, hence, improve as a scientific software user: "If they are motivated, they will learn…They are in the driver's seat." Adequate ability to articulate technical issues/questions (7) and communicate this to online peers (6) was an issue that was brought up by eight participants ("They need to read enough…to know to communicate in short what they need to ask"). Furthermore, the cost of resources and training was an issue for all interviewees; comments on the time and resources required to develop online tools were made, as well ("To do video tutorials and narrate what I am doing…The problem is that it takes several days to make a tutorial video, this is why they are usually not up-to-date").

The Internet is free (or low cost) and this can make it a popular tool. However, it is up to the users to develop the degree of sensitivity necessary to establish a solid level of understanding of the topic at hand; this can be accomplished by reading, investing time to develop their knowledge base, and critically analyzing pertinent information so that Internet resources can be used to their full capacity. Training on science- and engineering-specialized software appears, from the interviewees' perspectives, to be a flow diagram, as shown in Figure 3. The more the users understand, the more they can improve the interpretation of the software output and their research results. As well, the usefulness of the Internet as a resource appears to be connected to the degree of responsibility of the users, their degree of sensitivity regarding the topic at hand, as well as their ability to direct their own learning.
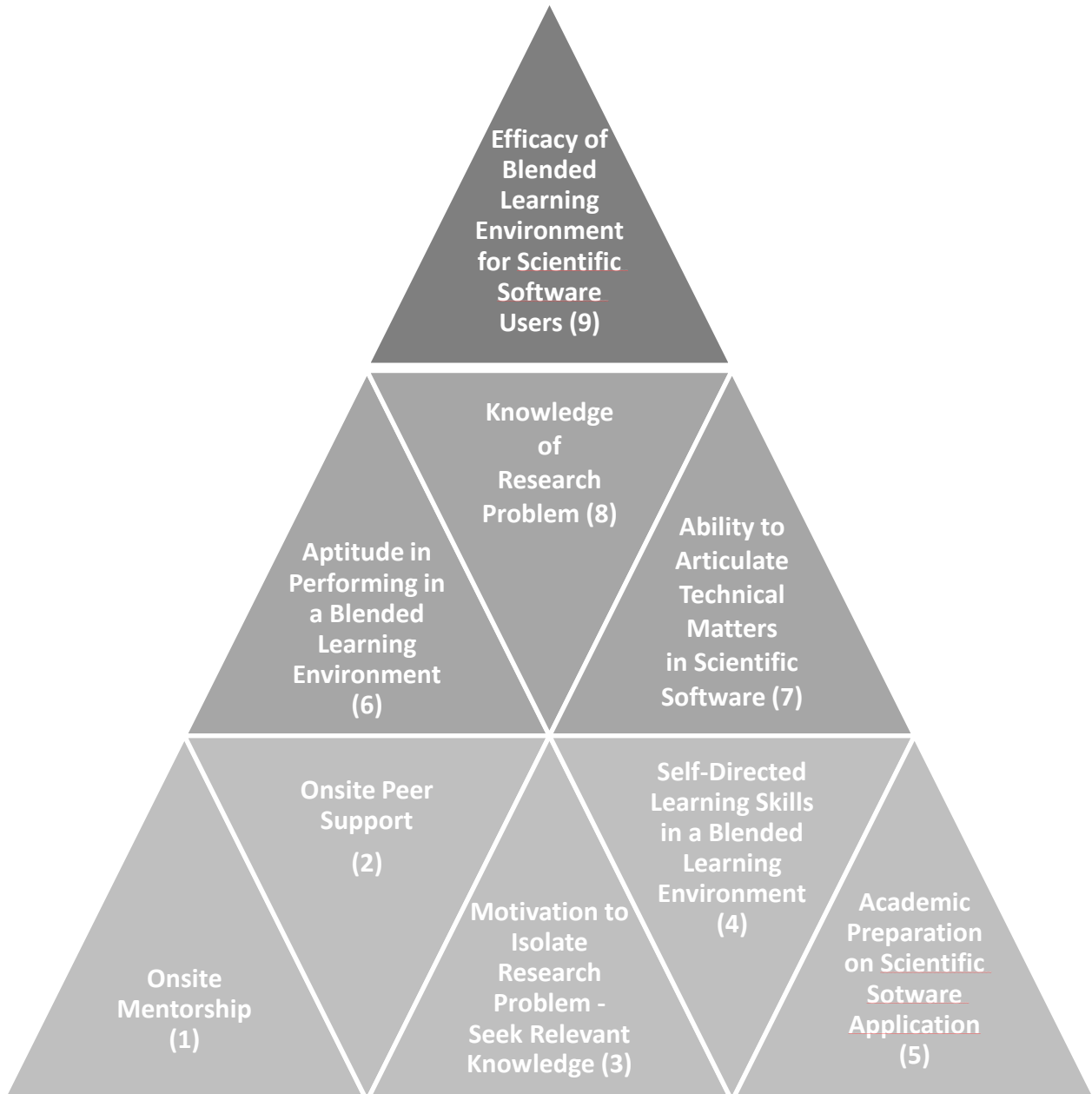
Figure 2. Parameters influencing the efficacy of blended learning in scientific software training.

As indicated by the data analysis of this study, once the learners become familiar with the blended environment, the mode of knowledge delivery does not affect them anymore. At this stage, they have become able to concentrate on the quality of the transferred knowledge, and not on the profile of the online peer or their level of familiarity of technology or the degree of social presence in their online interactions. This can significantly enhance their ability to learn by focusing and critically analyzing the core of the information rather than its mode of delivery.

A Systematic Training Cycle for Scientific Software Users

Figure 3, below, presents a systematic training cycle for scientific software users, as it emerged from this study. The data analysis in this investigation showed that the optimal training framework is an ongoing, methodical cycle that starts with: a dedicated user who makes a substantial investment of time on the software and topic at hand in order to expand his or her knowledge base, followed by the onsite peer support and guidance, which can progress to effective use of online resources (documentation and discussion forums) as the user becomes more and more independent. This leads back to the first step, with the user reflecting and dedicating time to absorb new knowledge, critique available information, and develop sensitivity about the research question at hand. As one of the interviewees noted: "It is good to explore your area."
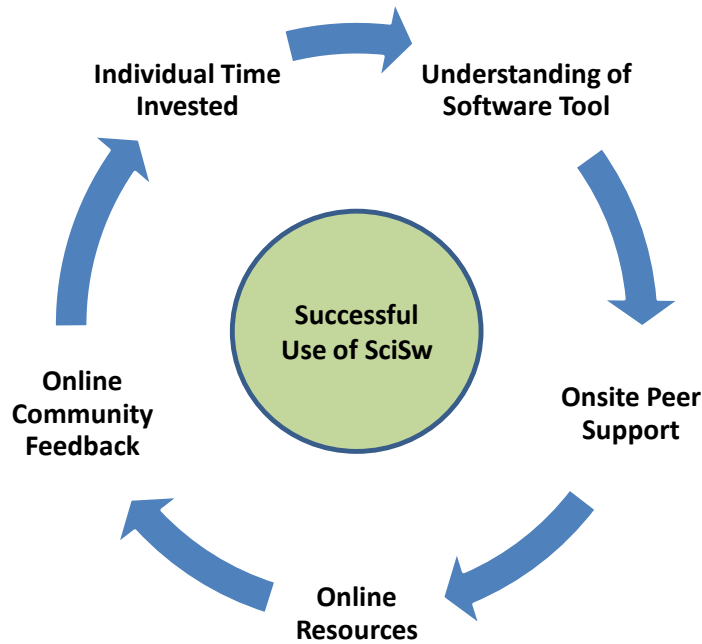
Systematic Training Cycle

Figure 3. A systematic training cycle for scientific software users.

The study participants pointed out that the limitations of the systematic training cycle for scientific software users, as presented above, include issues involving different levels of collaboration, potential intimidation, reliability of resources, as well as assumed knowledge in scientific software application environments where users receive training. For example, when asked to describe the collaborative learning design in their lab, participants responded that it really depended on the personality and background of the professor or mentor. As Participant 17 mentioned: "Well, this environment is totally built on the attitude of the

professors…" This improvised situation often exists in research laboratory settings and was described as a deeper issue by respondents who were young software users. The contributions of the respondents indicated that this ad hoc situation can impede their overall learning and confidence in their research results because of the following reasons: (a) the mentor (often supervisor) has uncertain previous training on constructive feedback techniques, teamwork, or teaching in a positive learning environment; (b) the student/trainee does not have adequate interaction time with mentors or colleagues; and (c) the student does not have previous formal training in working and learning within a team. Overcoming personality and interpersonal issues can often impede the development of the learner.

The reliability of online resources as well as the feedback offered by the online community were two elements that raised concerns among the respondents. This is in agreement with the findings of Pawlik, Segal, Sharp, and Petre (2012) who also raised the question of trustworthiness and reliability of the online sources from which the scientists learn about various aspects of software development. In their study, the scientist-participants did not have specific criteria with respect to assessing their online resources and whether these addressed their needs best or what type of online sources would be trustworthy and reliable for their work.

Discussion

The results of this study indicate that there are three key themes in the design of a training technique for successful application of scientific software. These themes are:

1. Responsibility in comprehension;

2. Discipline (and systematic validity procedures); and

3. Ability to adapt.

Figure 4 illustrates the relationships among these three key themes.
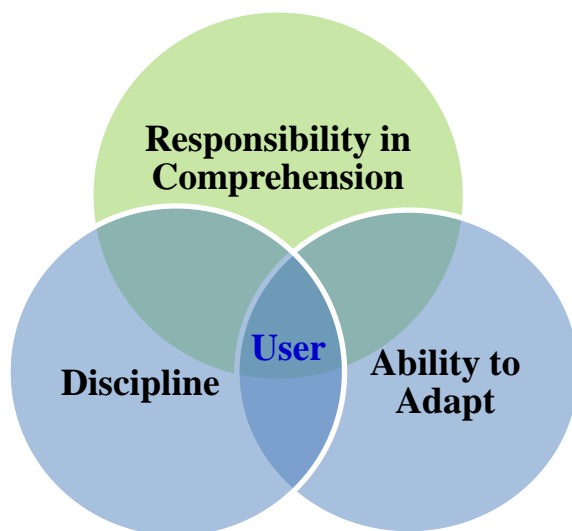
Figure 4. Key themes in scientific software user training.

The theme of responsibility in comprehension refers to the degree of ownership that the users demonstrate regarding the depth of their knowledge base in their area of study and the applicability of the scientific software tool. The latter would include the limitations or capabilities of the software and its usefulness in the problem-solving process of the research question at hand. Comprehension is defined here as having a solid background in the field of the problem that a user attempts to answer by using the scientific software product. Comprehension also refers to the degree of sensitivity that the users have developed regarding the context of the research question they are attempting to solve and how the software output relates to this context (Skordaki, 2016).

The theme of discipline refers to the ability of the users to build methodical strategies into their work and to diligently test their software programs. Also, it relates to the degree of responsibility of the users to systematically invest substantial time in updating their knowledge by reviewing current literature and software documentation that are pertinent to their research problem. Parnas (2010) pointed out that teaching students how to work in disciplined ways and to thoroughly test their software programs are critical elements in formal science and engineering education.

The theme of ability to adapt refers to a skill that is important to every scientific software user, as it empowers the individual with the aptitude to conduct his or her work regardless of the type of learning setting (traditional, online, or blended). Further, with distance technologies and software engineering continuously progressing, the users are expected to keep up with new information in their field of interest and collaborate in blended environments (Garrison, Anderson & Archer, 2001). As one of the research participants stated, "…our work is meaningful if the rest of the world can see it." Bereiter (1997) highlighted the importance of "learning how to function in a community of practice whose work is work with knowledge." McGreal (2009) indicated that reaching skilled professionals in today's global, dispersed work world remains a serious concern. Lingard (2010) pointed out that university students seldom receive any specific training on how to function collaboratively before group assignments are given, and little attention

is given to how teams are formed. Experienced users that were interviewed during this investigation reported that they had to "retrain" many times in order to stay current and understand how their field of interest continued to evolve in conjunction with their professional environment.

Through the careful analysis of all data that was available in this study, both from primary and secondary sources, it became evident that if any one of these three factors is missing, the user may not be in a position to employ the software reliably. The accurate application of scientific software requires that the users develop, in a responsible manner, a solid background both in the problem at hand, as well as the software tool of choice. If users are familiar with software validation procedures and have the ability to adapt to new learning environments but their knowledge of the topic at hand is insufficient (due to their undergraduate preparation or low degree of sensitivity in the subject), then the software application can be problematic because it may be used in the wrong context.

While this factor is necessary—i.e., knowing the dimensions of the problem at hand and the software—having the discipline to invest time to read pertinent documentation and follow software updates is equally crucial. By methodically keeping notes about software procedures and maintaining checkpoints to review the results during the software application, the users ensured that they were in control of the entire process, and that the software (with its capabilities and limitations) does not manipulate the research at any time. Current literature has indicated that the software product selection and usage are parameters that can influence the course of a research undertaking and, as such, there is a requirement for systematic reviews of the software output as well as careful analysis of the context within which the software product is employed (Lutters & Seaman, 2007; Joppa et al., 2013; Zacharia et al., 2015; Queiroz & Spitz, 2016).

Further, the results of this study show that the learning environment changes continually. This means that the users are expected to keep up with information that becomes available through various modes of delivery. As a result of this, the ability to adapt to new learning environments and modes of interaction becomes an indispensable parameter. Software evolves and knowledge is embedded in continually updated documentation that is shared through blended communications with various stakeholders; the skill to continually retrain becomes vital to the successful application of the software. Users of scientific software who wish to apply the software correctly need to constantly expand their ability to learn in Internet-supported environments and maintain their desire to enrich their knowledge base in their field of interest. As mentioned by research participants, self-paced learning and independent review of online resources and onsite documentation is often required from scientific software users working in industry and Higher Education. However, as Harrison (2015) indicated, this can be confusing if progress indicators and structure are not in place. Online resources such as video tutorials and e-learning documentation need to be carefully designed with accurate and updated information in order to support the learner and enhance the "quality of experience" (Ljubojevic, Vaskovic, Stankovic, & Vaskovic, 2014; Hsin & Cigas, 2013).

Conclusions

The purpose of this investigation was to explore current approaches regarding scientific software user training. The data analysis revealed that the optimal training framework is an ongoing cycle that

commences with the dedication by the user to explore the software tool and problem at hand, then seek onsite peer support and guidance, and progressively learn to use online resources effectively as the user becomes more and more independent. The cycle closes—that is, restarts—with the user reflecting and dedicating time to absorb new knowledge in order to develop sensitivity about their research question. The scientific software training framework that emerged through the amassed information of this study encapsulates various parameters that affect the development of the user knowledge base in a blended learning environment, including peer collaborative learning activities, methodical practices in checking software outputs with support from onsite/online communities of practice as well as user degree of sensitivity in the research problem and its associated context.

Acknowledgments

References

Adams, A., Davies, S., Collins, T., & Rogers, Y. (2010). Out there and in here: Design for blended scientific inquiry learning. In L. Creanor, D. Hawkridge, K. Ng, & F. Rennie (Eds.), *Proceedings of the 17th Association for Learning Technology Conference. "Into Something Rich and Strange: Making Sense of the Sea Change"* (pp. 149-157). Nottingham, England: Association for Learning Technology. Retrieved from http://oro.open.ac.uk/27397/3/out_there_and_in_here.pdf

Atkinson, R., & Flint, J. (2001). Accessing hidden and hard-to-reach populations: Snowball research strategies. *Social Research Update*, *33,* 1-4.

Bainbridge, S. (2013). Development and evaluation of a pedagogical model for an open university in Nepal based on geographical, regional, and linguistic factors (Doctoral Dissertation), Athabasca, AB: Athabasca University. Retrieved from https://dt.athabascau.ca/jspui/bitstream/10791/33/1/Dissertation.Bainbridge%20Final%20Copy.pdf

Bereiter, C. (1997). Situated cognition and how to overcome it. In D. Kirshner, & J. A. Whitson (Eds.), *Situated cognition: Social, semiotic, and psychological perspectives* (pp. 281-300). Hillsdale, NJ: Erlbaum.

Berg, B. L. (2001). *Qualitative research methods for the social sciences* (4th ed.). Boston, MA: Allyn & Bacon.

Bissell, C., & Endean, M. (2007). Meeting the growing demand for engineers and their educators: The potential for open and distance learning. In *Proceedings of the 2007 IEEE Meeting the Growing Demand for Engineers and Their Educators 2010–2020 International Summit* (pp. 1-18). Munich, Germany: Institute of Electrical and Electronics Engineers. Retrieved from https://www.infona.pl/resource/bwmeta1.element.ieee-art-000004760382

Cohen, L., Manion, L., & Morrison, K. (2007). *Research methods in education* (6th ed.). New York, NY: Routledge Taylor and Francis Group.

Corbin, J., & Strauss, A. (1990). Grounded theory research: Procedures, canons and evaluative criteria. *Zeitschrift fur Sociologie*, *19*(6), 418-427.

Dearden, R. (1984). Education and training. *Westminster Studies in Education*, *7*, 57-66.

Fischer, G. (2009). End-user development and meta-design: Foundations for cultures of -participation. In *Proceedings of the 2nd International Symposium on End-user Development* (pp. 3-14). Berlin, Heidelberg: Springer-Verlag. Retrieved from https://dl.acm.org/citation.cfm?id=1530504

Fischer, G. (2011). Understanding, fostering and supporting cultures of participation. *Interactions*, *18* (3), 42-53. Retrieved from http://l3d.cs.colorado.edu/~gerhard/papers/2011/interactions-coverstory.pdf

Frazier, L. J. (2006). *A qualitative study of technology-based training in organizations that hire agriculture and life sciences students.* (Master's thesis) College Station, Texas: Texas A & M University. Retrieved from http://oaktrust.library.tamu.edu/bitstream/handle/1969.1/5917/etd-tamu-2006A-AGED-Frazier.pdf?sequence=1

Garrison, R. D., Anderson, T., & Archer, W. (2001). Critical thinking, cognitive presence, and computer conferencing in distance education. *American Journal of Distance Education*, *15*(1), 7-23. Retrieved from https://www.tandfonline.com/doi/abs/10.1080/08923640109527071

Garrison, R. D., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education 7*(2), 95-105. doi:10.1016/j.iheduc.2004.02.001

Glaser, B. G., & Strauss, A. L. (1999). *The Discovery of grounded theory: Strategies for qualitative research.* New York: Aldine de Gruyter.

Graham, C. R., Henrie, C. R., & Gibbons, A. S. (2014). Developing models and theory for blended learning research. In A. G. Picciano, C. D. Dziuban, & C. R. Graham (Eds.), Blended learning: Research perspectives (v. 2; pp. 13-33). New York, NY: Routledge.

Hannay, J. E., Langtangen, H.P., MacLeod, C., Pfahl, D., Singer, J., & Wilson, G. (2009). How do scientists develop and use scientific software? In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Computational Science and Engineering,* (pp. 1-8). Washington DC: IEEE Computer Society Washington. Retrieved from https://dl.acm.org/citation.cfm?id=1556928

Harrison, D. J. (2015). Assessing experiences with online educational videos: Converting multiple constructed responses to quantifiable data. *International Review of Research in Open and Distributed Learning*, *16*(1), 168-192. Retrieved from www.irrodl.org/index.php/irrodl/article/view/1998/3205

Hoepfl, M. C. (1997). Choosing qualitative research: A primer for technology education researchers. *Journal of Technology Education*, *9*(1), 47-63. Retrieved from https://scholar.lib.vt.edu/ejournals/JTE/v9n1/hoepfl.html

Holton, G. A. (2004). Defining risk. *Financial Analysts Journal*, *60*(6), 19-25.

Howison, J., & Herbsleb, J. D. (2011). Scientific software production: Incentives and collaboration. In *Proceedings of the ACM 2011 Computer Supported Cooperative Work* (pp. 513-522). ACM. doi: 10.1145/1958824.1958904

Hsin, W. J., & Cigas, J. (2013). Short videos improve student learning in online education. *Journal of Computing Sciences in Colleges, 28*(5), 253-259. Retrieved from https://dl.acm.org/citation.cfm?id=2458622

Hurt, A. C. (2007). Exploring the process of adult computer software training using andragogy, situated cognition, and a minimalist approach. In *Proceedings of the International Research Conference in the Americas of the Academy of Human Resource Development* (pp. 6-14)*,* Indianapolis, IN, United States of America. Retrieved from http://files.eric.ed.gov/fulltext/ED504840.pdf

Joppa, N. L., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K. … Emmott, S. (2013). Troubling trends in scientific software use. *Science*, *340*(6134), 814-815. doi: 10.1126/science.1231535

Lingard, R. W. (2010). Teaching and assessing teamwork skills in engineering and computer science. *Systemics, Cybernetics and Informatics*, *8*(1), 34-37. Retrieved from http://www.iiisci.org/journal/CV\$/sci/pdfs/GQ816EX.pdf

Ljubojevic, M., Vaskovic, V., Stankovic, S., & Vaskovic, J. (2014). Learning and quality of experience using supplementary video in multimedia instruction as a teaching tool to increase efficiency of learning and quality of experience. *International Review of Research in Open and Distributed Learning*, *15*(3), 275-291. Retrieved from www.irrodl.org/index.php/irrodl/article/view/1825/2903

Lowe, J. S. (2004). *A theory of effective computer-based instruction for adults* (Doctoral dissertation). Baton Rouge, LA: Louisiana State University. Retrieved from http://etd.lsu.edu/docs/available/etd-04132004-172352/

Lutters, W. G., & Seaman, C. B. (2007). Revealing actual documentation usage in software maintenance through war stories. *Information and Software Technology, 49,* 576-587.

McGreal, R. (2009). A case study of an international e-learning training division: Meeting objectives. *International Review of Research in Open and Distance Learning*, *10*(6), 1-20. doi: 10. 10.19173/irrodl.v10i6.619

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Thousand Oaks, CA: Sage Publications.

Moghaddam, A. (2006). Coding issues in grounded theory. *Issues in Educational Research, 16,* 52-66. Retrieved from http://www.iier.org.au/iier16/2006conts.html

Morse, J. M., Barrett, M., Mayan, M., Olson, K., & Spiers, J. (2002). Verification strategies for establishing reliability and validity in qualitative research. *International Journal of Qualitative Methods, 1*(2), 1-19. Retrieved from https://sites.ualberta.ca/~iiqm/backissues/1_2Final/pdf/morseetal.pdf

Parnas, D.L. (2010). Inside risks: Risks of undisciplined development. *Communications of the ACM October 2010,* *53*(10), 25-27. doi: 10.1145/1831407.1831419

Pawlik, A. Segal, J. Sharp, H., & Petre, M. (2012). Developing scientific software: The role of the Internet. In Tokar, A., Beurskens, M., Keuneke, S., Mahrt, M., Peters, I. Puschmann, C. … & Weller, K. (Eds.), *Science and the Internet* (pp. 263-273). Düsseldorf: Düsseldorf University Press.

Queiroz, F., & Spitz, R. (2016). The lens of the lab: Design challenges in scientific software. *The International Journal of Design Management and Professional Practice, 10*(3), 17-45. Retrieved from http://ijgmpp.cgpublisher.com/product/pub.239/prod.94/m.2

Rajendran, N.S. (2001). Dealing with biases in qualitative research: A balancing act for researchers. Paper presented at *Qualitative Research Convention 2001*: *Navigating Challenges*. Kuala Lumpur: University of Malaya. Retrieved from http://nsrajendran.tripod.com/Papers/Qualconfe2001.pdf

Robinson, H., Segal, J., & Sharp, H. (2007). Ethnographically-informed empirical studies of software practice. *Information and Software Technology*, *49*, 540-551. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0950584907000110

Saldana, J. (2008). *Coding manual for qualitative researchers*. Los Angeles, CA: Sage Publications.

Segal, J. (2005). When software engineers met research scientists: A case study. *Empirical Software Engineering, 10*, 517-536. Retrieved from https://rd.springer.com/article/10.1007/s10664-005-3865-y

Segal, J. (2007). Some problems of professional end user developers. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing* (pp. 111-118). Coeur d'Alene, Idaho, United States of America. Doi**:** 10.1109/VLHCC.2007.17

Skordaki, E. M. (2016). Mobilizing knowledge in science and engineering: Blended training for scientific software users (Doctoral Dissertation), Athabasca, AB: Athabasca University. Retrieved from https://dt.athabascau.ca/jspui/bitstream/10791/213/1/SKORDAKI%20EfrosyniMaria_EdDThesis%20-December%208th%202016_Final.pdf

Skordaki, E. M., & Bainbridge, S. (2015a). Knowledge mobilization in scientific communities. *Education Innovation & Research Symposium 2015*. Retrieved from https://www.researchgate.net/publication/297460865_Knowledge_Mobilization_in_Scientific_Communities_Blended_Training_for_Accurate_Application_of_Scientific_Software

Skordaki, E. M., & Bainbridge, S. (2015b). Blended training and collaborative learning for software users in the geosciences. *GeoQuebec Conference 2015: Challenges from North to South*, September 2015, Quebec City, Canada. Retrieved from http://members.cgs.ca/documents/conference2015/GeoQuebec/papers/752.pdf

Sloan D.**,** Macaulay C., Forbes P., Loynton S., & Gregor P. (2009). User research in a scientific software development project. In *Proceedings of the 2009 British Computer Society Conference on Human-Computer interaction (Cambridge, United Kingdom, September 01 - 05, 2009)* (pp. 423-429). Swinton, UK: British Computer Society. Retrieved from http://www.bcs.org/upload/pdf/ewic_hci09_paper55.pdf

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Newbury Park, CA: Sage Publications, Inc.

Turner, D.W., III. (2010). Qualitative interview design: A practical guide for novice investigators. *The Qualitative Report, 15*(3) 754-760. Retrieved from http://www.nova.edu/ssss/QR/QR15-3/qid.pdf

Zacharia, Z. C., Manolis, C., Xenofontos, N., de Jong, T., Pedaste, M., van Riesen, S. A. N. … Tsourlidaki, E. (2015). Identifying potential types of guidance for supporting student inquiry when using virtual and remote labs in science: A literature review. *Educational Technology Research and Development*, *63*(2), 257-302. Retrieved from https://www.learntechlib.org/p/159959/