2013

Developing Software Requirements for a Knowledge Management System that Coordinates Training Programs with Business Processes and Policies in Large Organizations

James Richard Kiper
Nova Southeastern University, kiper@nova.edu

Developing Software Requirements for a Knowledge Management System that Coordinates Training Programs with Business Processes and Policies in Large Organizations

by
J. Richard Kiper

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Computing Technology in Education

Graduate School of Computer and Information Sciences
Nova Southeastern University
2013

An Abstract of a Dissertation Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Developing Software Requirements for a Knowledge Management System that Coordinates Training Programs with Business Processes and Policies in Large Organizations

by
J. Richard Kiper
April 2013

For large organizations, updating instructional programs presents a challenge to keep abreast of constantly changing business processes and policies. Each time a process or policy changes, significant resources are required to locate and modify the training materials that convey the new content. Moreover, without the ability to track learning objects to processes and policies, training managers cannot conduct an effective training gap analysis in these areas. As a result, the corporate training picture is unclear and instructional needs cannot be accurately determined.

The research addressed these problems by recognizing the need for linkages between an organization's business processes, its policies, and the learning objects that package the corresponding training content and deliver it to the workforce. The overall investigation was completed in three parts. In the first study, a thorough examination of the literature was conducted to determine the extent of the research problem and to provide a theoretical foundation for a solution. In the second study an expert panel was used to elicit user needs for a knowledge management system that addresses training management shortcomings in a large law enforcement agency. Another expert panel from that agency validated and prioritized the user needs during the third study. Through a combination of research-based elicitation and validation techniques, an accurate list of natural language software requirements emerged to represent the collective needs of the law enforcement training experts. The software requirements may now serve to analyze the capabilities of existing information technology systems or to form the basis for a request for proposal (RFP) to build the envisioned knowledge management system.

Acknowledgements

This work represents a cornucopia of encouragement and expertise I have received from people who were kind enough to invest in my life, for I have nothing that I did not first receive from others (1 Corinthians 4:7).

I would like to thank my dissertation committee, Dr. Timothy Ellis, Dr. Trudy Abramson, and Dr. Marti Snyder, whose sharp eyes kept my research focused and my writing style intelligible. I credit Dr. Ellis in particular, for his rigorous adherence to quality and for his patience as we forged new ground with a three paper dissertation format. Because of their efforts I know that my work is the best it can be.

I want to thank my wife, Judy, who endured many days and nights alone while I plunked away on a laptop always claiming that I was "almost done." She deserves at least half of this doctoral degree as compensation for lost time and attention. I thank my children, Ted, Cindy, Valerie, and Ingrid, for their encouragement during these years and appreciating that when it came to writing assignments – I felt their pain.

I am grateful to my late father, who always told me I could do anything I put my mind to.

I thank my colleagues and the participants in this research who believed we can truly solve problems if we combine our brainpower and work together.

Finally, I am grateful to my employing agency, which funded my graduate education and gave me the best job in the world – the privilege of serving our great nation.

Table of Contents

List of Tables

Tables

Chapter Three

Chapter Four

List of Figures

Figures

Chapter Two

Chapter Three

Chapter Four

Chapter One

Research Agenda

Introduction

Large organizations that manage numerous business processes have special training needs. Many of these organizations have begun implementing the principles of business process management (BPM), which seeks to make an organization more efficient by modifying workflow and eliminating wasteful tasks (Trkman, 2009). However, if these changes affect the entire organization, then many thousands of employees may need to be trained on hundreds of policies and tasks in the new system (Reijers, Mans, & van der Toorn, 2009). Unfortunately, organizational policies and business processes are rarely mapped to the existing training products and policies that support them (Hawryszkiewycz, 2005). This oversight makes it difficult to locate and modify training content to reflect organizational changes.

Problem Statement

The business processes and organizational policies of large organizations are not coordinated with their training strategies, resulting in irrelevant and ineffective training programs. The research focused on the training challenges of large organizations that manage several thousand employees, with dozens of job families and offices located all over the world. For the sake of readability this type of large organization will be referred to as a CALO, defined by Kiper (2008) as "a Company, Agency, or other Large Organization" (p. 14). Examples of CALOs exist in the public sector (e.g., U.S. State Department, Federal Bureau of Investigation) as well as in the private sector (e.g., Microsoft, Coca-Cola).

Definition of Terms

A business process (BP) is a segment of defined activity in an organization. It is a set of tasks that are logically related to fulfill an organization's objective (Trkman, 2009). A BP may be modeled in workflow diagrams using Business Process Modeling Notation (BPMN), and executed in software applications using Business Process Modeling Language (BPML) (Fischer, 2002). Through the use of symbols and arrows, BP models define how work is done and who does it. Due to its size and diversity, a CALO's workforce participates in many hundreds of BPs every day (Reijers, Mans, & van der Toorn, 2009).

An organizational policy (OP) refers to a CALO's documented guidance to the organization. OPs articulate outcomes and strategic goals that are in practice fulfilled by BPs (Trkman, 2009). OPs may be produced internally by the CALO or adopted from a higher authority, such as the CALO's parent organization. Statutes, regulatory guidelines, employee handbooks, and standard operating procedures are examples of OPs. An OP may take the form of an electronic or physical document, and convey the specific rules, protocols, or directives approved by CALO executive managers.

Finally, a learning object (LO) is a construct used to chunk educational material into smaller units for content management in online learning environments (Beck, 2005). However, Hawryszkiewycz (2005) questions whether the popular concept of LOs has been implemented appropriately in large organizations. In fact, Feldstein (2006) argues that the electronic packages in learning management systems are more similar to traditional instructional materials than true LOs. Therefore, for the purpose of this research an LO is defined broadly as a container of learning materials that manages instructional content as well as supporting resources maintained by an organization. It is a collection of BPs, OPs, and the instructional products (e.g., PowerPoint slides, job aids, demonstrations) needed to communicate the BPs and OPs to the learner.

LOs, BPs, and OPs are the primary concepts that drive the research described herein. Other terms used throughout the research include the following acronyms, which for convenience are repeated in Appendix A.

- **Analytic Hierarchy Process (AHP)**, a method of prioritizing software requirements involving pair-wise comparisons (Karlsson, Wohlin, & Regnell, 1998).
- **Business Process (BP)**, a segment of defined activity, logically related to fulfill an organization's objective (Trkman, 2009).
- **Business Process Management (BPM)**, a corporate effort that seeks to make an organization more efficient by modifying workflow and eliminating wasteful tasks (Trkman, 2009).
- **Business Process Modeling Language (BPML)**, a computer language used to automate business processes (Fischer, 2002).
- **Business Process Modeling Notation (BPMN)**, the standardized set of diagram elements used to depict business processes (Fischer, 2002).
- **Company, Agency, or other Large Organization (CALO)**, a shorthand term that refers to large organizations in general (Kiper, 2008).
- **Computer Managed Instruction (CMI)**, an IEEE standard for building learning objects (see http://www.ieeeltsc.org/working-groups/wg11CMI/CMIdescription).
- **Course Management System (CMS)**, an application used by Zhang and Su (2007) to build computing courses.

- **Commercial Off The Shelf (COTS)**, a common term referring to commercially available solutions.
- **Electronic Performance Support System (EPSS)**, an application proposed by Raybould (1995) to manage corporate knowledge in an organization.
- **Enterprise Resource Planning (ERP)**, an industry standard for the integration of business processes and transactional data (Liang & Xue, 2005).
- **Federal Law Enforcement Training Accreditation (FLETA)**, the organization that accredits training programs for federal law enforcement agencies (see www.fleta.gov).
- **Hierarchical Cumulative Voting (HCV)**, a token-based method for prioritizing software requirements (Berander & Svahnberg, 2009).
- **High-level Requirement (HLR)**, a software requirement written at a general level and is supported by lower-level requirements (LLRs) (Berander & Svahnberg, 2009).
- **Institute of Electrical and Electronics Engineers (IEEE)**, a body that governs the standards for hardware and software development (see www.ieee.org).
- **Institutional Review Board (IRB)**, the common term for a committee that ensures compliance with ethical research guidelines.
- **Joint Application Design (JAD)**, a consensus-building technique for requirements elicitation that involves a facilitator and face-to-face meetings (Coughlan & Macredie, 2002).
- **Job Task (JT)**, a common term referring to an individual's contribution to a business process.
- **Knowledge Management (KM)**, a business field that involves the creation, codification, and sharing of corporate information and experiences (Turban & Aronson, 2001).

- **Low-level Requirement (LLR)**, a software requirement that describes a specific task expected to be accomplished in software. LLRs are organized under high-level requirements (HLRs), which they support (Berander & Svahnberg, 2009).
- **Learning Object (LO)**, a container of learning materials that manages instructional content as well as supporting resources (Feldstein, 2006).
- **Lesson Plan (LP)**, the common term for a description of the content and strategies used in a particular unit of instruction.
- **Nominal Group Technique (NGT)**, a consensus-building technique that uses the individual, written contribution of members to build a common understanding of a group (Landeta, Barrutia, & Lertxundi, 2011).
- **Online Learning Environment (OLE)**, a computer-based system that delivers training electronically (O'Brien & Hall, 2004).
- **Organizational Policy (OP)**, a general term referring to the written guidance to an organization.
- **Quality Assurance (QA)**, a requirements elicitation strategy for ensuring the value of requirement descriptions (Denger & Olsson, 2005).
- **Requirements Elicitation (RE)**, the process of gathering software requirements from the customers or users of a system (Davey and Cope, 2008; Laporti, Borges, & Braganholo, 2009).
- **Request for Proposal (RFP)**, the common term for a document an organization publishes to elicit bids from potential vendors to address a particular business need.

- **Sharable Content Object Reference Model (SCORM)**, a standard that defines the metadata and components of reusable learning objects (see http://www.adlnet.gov/Technologies/scorm/default.aspx).
- **Soft Systems Methodology (SSM)**, a requirements elicitation technique that represents the problem space in pictorial form (Checkland, 1998; Coughlan & Macredie, 2000).
- **Software Requirements Specification (SRS)**, the written representation of user needs for a software application (IEEE, 1998).
- **Technology-enhanced Learning (TEL)**, a computing-based strategy for delivering corporate training proposed by Capuano, Gaeta, Ritrovato, and Salerno (2008).
- **Training Needs Analysis (TNA)**, a process by which an organization may define its instructional requirements (O'Brien and Hall, 2004).

Research Goal

The goal of the research was to develop software requirements for a knowledge management system by which training content may be mapped to a CALO's business processes and organizational policies. In essence, the research explored how CALOs decompose corporate activity and policy into business processes and job tasks, whose success is dependent on effective training strategies. Considerable attention was given to existing learning products such as lesson plans, and how they may be tagged with metadata to facilitate their mapping to OPs and BPs. The OPs and BPs themselves were also examined for mapping opportunities.

Large scale training programs are difficult to create and manage – and their failure could be very costly. The results of the research will assist CALOs in developing strategies to analyze their OPs, define their BPs, and then map them to meaningful instructional content used to teach those concepts to their employees. It was designed to contribute to the existing knowledge and practices of business process modeling, knowledge management, and instructional systems design. Although the research focus was centered on a sizable law enforcement organization, it is anticipated that the results may be generalized to any CALO, including other government agencies and private corporations.

Barriers and Issues

Conducting research in the context of a large law enforcement organization posed special challenges. For example, no law enforcement sensitive or classified content could be used in the collection of data or the reporting of findings. Instead, the focus of the methodology was on the mechanism(s) that managed the content, and on representative content that would not compromise law enforcement or national security guidelines.

As requirements were developed to link BPs and OPs to LOs, study participants frequently identified additional problems (such as political or ownership issues), which were unrelated to KM shortcomings and out of the scope of the research. However, the participants were reminded about the goals of the research, and they cooperated completely as their raw input was guided into relevant statements of user needs.

Finally, the concepts of LOs and BPs are foreign concepts to those who are not immersed in the areas of learning management systems and business process management. To mitigate participant confusion, therefore, the term *lesson plan* was substituted for LO and the concept of *job task* replaced BP. In this way, participants were not required to learn a new vocabulary to contribute to the development of software requirements.

Relevance and Significance

A major concern of those who manage KM projects is that the definition of requirements is expensive, and occupies a large portion of the KM project development schedule – as much as 20-25% of the total project time (Japenga, 2011). However, one of the advantages of the current research project is that the time required to develop robust requirements had no impact on the cost or schedule of the KM project itself. This benefit is due to the fact that such a project had not yet been initiated – or even envisioned – by the stakeholders. Rather, the requirements developed by this research will help justify a plan to create such a KM system, with the requirements forming the basis of a formal Request for Proposal (RFP).

In order to arrive at a rich, accurate set of software requirements for the future KM system, three separate studies were completed. The first study explored the relevance, significance, and theoretical foundation for the problem under investigation. A thorough review of the literature uncovered similar problems identified in other CALOs, and the current state of success for remedies that were attempted in those cases. This first paper provided the justification for pursuing the next phase of investigation, the development of software requirements for a KM system.

The goal of the second study was to develop a set of rich, natural language software requirements for a KM system that will map LOs to BPs and OPs. Knowledgeable, experienced members of a CALO workforce were recruited to participate in a requirements elicitation (RE) process. The employed RE techniques were based on a combination of best practice methods found in the literature. Qualitative data from the participants were used to construct requirements that represent their collective goals for the future KM system.

The third study aimed to validate the software requirements generated in the second study. Research-based methods were used to validate the content as well as the priority ranking of the requirements. The output from this third study was a validated set of prioritized requirements that may be used as the basis for a possible RFP to solicit a contractor to build the KM system.

The most common reason for the failure of information system projects is poor software requirements resulting from poor RE processes (Davey and Cope, 2008; Laporti, Borges, &

Braganholo, 2009). The research documented by these studies will help CALO managers develop a KM system that will avoid this fate.

The following three chapters describe each of the studies that addressed the research problem. Chapter Two represents a paper that established the theoretical foundation for the problem. Chapter Three documents the study that elicited preliminary software requirements from an expert panel. Chapter Four describes the study whereby participants validated and prioritized those requirements, resulting in a final list representing user needs for a KM system. Each chapter that describes a study includes the research questions, definitions, assumptions, background, methods, analysis, results, and conclusions that pertain to that particular study. By considering all three studies as a whole, Chapter Five documents the conclusions, implications, and recommendations for future considerations. Finally, Appendix A is provided as a quick reader's guide to the acronyms most frequently used in this research.

Chapter Two

Theoretical Study

Abstract

Training programs suffer when they become outdated and irrelevant, which happens when the training content does not match what is supposed to be taught. It is the duty of curriculum managers to update instructional materials to reflect the latest organizational policies— what the workforce is allowed to do— and business processes— job tasks that are actually carried out. However, organizations with large workforces have difficulty modifying their individual training products to keep up with the thousands of policies and processes that change frequently. By exploring the issue through a review of the literature this paper sets forth the theoretical foundation for creating a knowledge management (KM) system that will address the problem. Readers who are employees of large organizations will benefit from learning about the nature and impact of the problem–which is rarely understood–and the challenges of creating a KM system that will solve the problem.

Keywords

Business process management, knowledge management, corporate learning, systems development, instructional systems design

Introduction

Large organizations that manage numerous business processes have special training needs. Many of these organizations have begun implementing the principles of business process management (BPM), which seeks to make an organization more efficient by modifying workflow and eliminating wasteful tasks (Trkman, 2009). However, if these changes affect the entire organization, then many thousands of employees may need to be trained on hundreds of policies and tasks in the new system (Reijers, Mans, & van der Toorn, 2009). Unfortunately, organizational policies and business processes rarely are mapped to the existing training products and policies that support them (Hawryszkiewycz, 2005). This oversight makes it difficult to locate and modify training content to reflect organizational changes.

This paper is an exploratory study that uncovers the presence and impact of the above problem and provides the basis for creating a knowledge management (KM) system that can map an organization's educational products to its policies and business processes. By clarifying the impact and significance of such a gap between the KM and training efforts of large organizations, this paper seeks to raise awareness of the problem so that organizational leaders are equipped to evaluate solutions. The principle question for this study is: How does existing literature address the problem of coordinating learning programs with business processes and organizational policies?

The remainder of this paper is organized as follows: A set of definitions is provided to clarify the scope and context of the research. Next, the challenges of corporate training, the problems with business process modeling, and the lack of commercial off-the-shelf (COTS) solutions are explored to provide a deeper understanding of the research problem. This is followed by a discussion of the impact of effective (and ineffective) corporate training and the specific benefits of getting it right. Next, a thorough review of the literature will uncover similar problems identified in other organizations and the current state of success for any remedies that were attempted in those cases. Finally, a summary of the research findings is presented along with considerations for future work. Appendix A is provided to assist the reader in recalling pertinent acronyms and definitions.

Definitions

The business processes and policies of large organizations are not coordinated with their training strategies, resulting in irrelevant and ineffective training programs. This paper will focus on the training challenges of large organizations that manage several thousand employees with dozens of job families and offices located all over the world. For the sake of readability, this type of large organization will be referred to as a CALO, defined by Kiper (2008) as "a Company, Agency, or other Large Organization" (p. 14). Examples of CALOs exist in the public sector (e.g., U.S. State Department and Federal Bureau of Investigation) as well as in the private sector (e.g., Microsoft and Coca-Cola).

A business process (BP) is a segment of defined activity in an organization. It is a set of tasks that logically are related to fulfill an organization's objective (Trkman, 2009). A BP may be modeled in workflow diagrams using Business Process Modeling Notation (BPMN), and executed in software applications using Business Process Modeling Language (BPML) (Fischer, 2002). Through the use of symbols and arrows, BP models define how work is done and who does it. Due to its size and diversity, a CALO's workforce participates in hundreds of BPs every day (Reijers, Mans, & van der Toorn, 2009).

An organizational policy (OP) refers to a CALO's documented guidance to the organization. OPs articulate outcomes and strategic goals that are, in practice, fulfilled by BPs (Trkman, 2009). OPs may be produced internally by the CALO or adopted from a higher authority such as the CALO's parent organization. Statutes, regulatory guidelines, employee handbooks, and standard operating procedures are examples of OPs. An OP may take the form of an electronic or physical document and convey the specific rules, protocols, or directives approved by CALO executive managers.

Finally, a learning object (LO) is a construct used to chunk educational material into smaller units for content management in online learning environments (Beck, 2005). However, Hawryszkiewycz (2005) questions whether the popular concept of LOs has been implemented appropriately in large organizations. In fact, Feldstein (2006) argues that the electronic packages in learning management systems are more similar to traditional instructional materials than true LOs. Therefore, for the purpose of this research, an LO is defined broadly as a container of learning materials that manages instructional content and supports resources maintained by an organization. It is a collection of BPs, OPs, and the instructional products—PowerPoint slides, job aids, and demonstrations—needed to communicate the BPs and OPs to the learner.

The Challenge of Corporate Training

Several large organizations, government and private, currently are undertaking challenging efforts in business process management (BPM). For example, government agencies, such as the Federal Bureau of Investigation (FBI), are attempting to modify existing business processes as they move from 100-year-old paper-based workflow systems to electronic systems of record (Miller, 2005). However, radical modifications to corporate business processes require updated policies, new technologies, and effective training methods to communicate these changes to the entire enterprise. As an additional challenge large organizations (CALOs), such as the FBI, maintain a diverse workforce that may include dozens of job families, various levels of experience, and offices that are located all over the world.

Capuano, Gaeta, Ritrovato, and Salerno (2008) identified the significant problem of coordinating corporate learning programs with business processes, which is why they focus on technology-enhanced learning (TEL) solutions. They recognize that “a better integration of TEL with business process management is, in fact, one of the greatest challenges for today’s knowledge management" (p.56). In a CALO, hundreds of OPs are used to generate thousands of BPs. But how do workers learn about relevant BPs and the OPs that undergird them?

It is obvious that effective corporate training is needed to communicate OPs and BPs. However, what happens when BPs and OPs change? How do training managers know which instructional materials need to be modified in order to bring the workforce into compliance with the new OPs and BPs? In other words, which parts of existing web-based training modules, new employee training curricula, and PowerPoint presentations contain the content that covers the OPs and BPs in question? According to Raybould (1995), the coordination between organizational performance and learning products "often relies on haphazard and inefficient manual processes. Since these processes often are slow the cycle time is long, and as a result the organization is not responsive to changes in the business environment" (p.12).

CALOs cannot afford to be unresponsive to changing OPs and BPs. Figure 1 illustrates the relationship between OPs, BPs, and traditional training products. Training products are depicted as items on an assembly line, where OPs and BPs are inserted into each product. After merchandise leaves an assembly line there is always a chance that the product components will need to be updated, requiring the manufacturer to announce a recall so that the products may be replaced or updated with new components. In the world of consumer products a failure to recall defective items could result in public safety issues.

In the same way, when OPs and BPs change and the training products do not the training becomes either 1) irrelevant, because it teaches concepts that are no longer part of the updated OPs and BPs; or 2) ineffective, because the training no longer covers current requirements. The problem is obvious: After training products leave the "assembly line," they lose their association with the supply source of OPs and BPs.

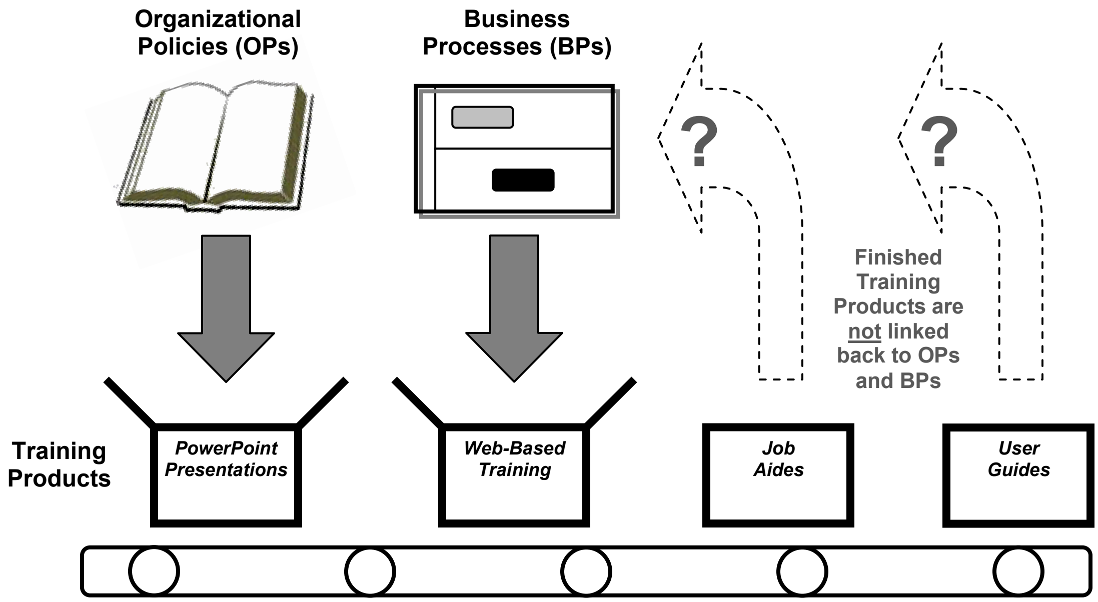


Figure 1: The relationship between organizational policies (OPs), business processes (BPs), and training products.

Without a link back to the changed OPs and BPs it is difficult for a CALO to "recall" the training products and retrofit them with current OPs and BPs. However, training workers in *current* OPs and BPs is crucial for the success of BPM, according to Trkman (2009) who complains that the lack of a good grounded theory restricts BPM to "the fad phase" (p.1) of development.

The Difficulty of Business Process Modeling

To model a business process is to depict a portion of an organization's enterprise knowledge with flowcharts and diagrams. This activity can be very useful in helping a CALO understand the details of what is taking place in the organization, which people (actors) are making things happen, and where streamlining processes may be necessary to improve efficiency. Unfortunately, however, the task of defining a business process is complex and time-consuming, requiring multiple iterations to document the relationships properly (Virine and

Rapley, 2003). Moreover, the difficulty of developing BP models is compounded by the thousands of BPs managed by a CALO together with its large and diverse workforce.

If a CALO's knowledge is structured, static, and explicit the formal documentation, such as BP modeling, is an effective way to record that knowledge (Song, Nerur, & Teng, 2007). However, the truth is that BPs are constantly changing–either through internal pressures or by external mandates. CALO managers understand that "changes in business strategies or new business collaborations lead to modifications… of their underlying business process models." (Koschmider & Oberweis, 2007, p. 1263). If OPs affect changes in BPs, then BP modeling strategies need to be very agile.

While BP models are difficult to develop for meeting OP demands, they are even more difficult to incorporate into training. The challenge lies in the fluidity of the BPs (Koschmider & Oberweis, 2007), which must be incorporated into training products that may not be as modifiable. As a CALO's business process modeling committee debates the specifics of a particular business process, instructional designers–who must incorporate the process content into a variety of instructional products–are forced to wait for the process to be defined or risk extensive rework on learning aids that have been produced in the meantime. In either case, the strong dependency between defining a business process and its corresponding training content can cause a CALO to spend significant time and money in an attempt to keep the two activities synchronized.

The Trouble with Commercial Off-The-Shelf (COTS) Solutions

Simply placing curriculum materials in an electronic format does not guarantee successful BPM training. Zhang and Su (2007) argued that the variety and scope of business processes require careful attention to matching online learning tools to training needs. Newton and Doonga (2007) reviewed several e-learning initiatives and discovered that "corporate e-training did not address strategic business objectives." They concluded their study by stating that the problem "continues to be a major issue" (p. 127).

Unfortunately, it appears that commercial vendors are not coming to the rescue. Much of today's corporate learning concentrates on tasks that are common across the workforce rather than on more granular, job-specific responsibilities (Hawryszkiewycz, 2005). Even software packages designed for enterprise resource planning (ERP)–which is supposed to integrate business processes and transactional data– have not provided the flexibility needed to train a large, diverse workforce (Liang & Xue, 2005). According to Kay (2003), the e-learning market is flooded with canned, "one-size-fits-all" solutions that do not meet the needs of different audiences within organizations. Kay also argues that large audiences require a blended approach to training whereby e-learning is integrated with traditional modes of instructional delivery, such as instructor-led training.

However, multiple modes of training create enormous difficulties for updating curriculum materials, especially when the content of the training materials is derived from OPs and BPs that are changing. In CALOs, new business strategies and collaborations always will lead to changes in the underlying BP models (Koschmider & Oberweis, 2007). The challenge is to effectively communicate the new BPs to the workforce. Regrettably, the literature indicates that current online learning strategies often miss the mark. Martin, Leyking, and Wolpers (2008) articulate the problem succinctly: "Existing eLearning approaches miss consistent alignment with business operations and objectives" (p. 1). The challenges in updating organizational policies and codifying tacit business knowledge—business process modeling—together with the difficulties of creating and maintaining instructional products call for an intervention that is robust and responsive enough to bridge the gap between a CALO's organizational policies, its business processes, and its training strategies.

The Impact of Ineffective Corporate Training

An assessment of the research problem begins with a look at how CALOs train their workforce and what happens when the training falls short. CALOs have many business processes–sometimes thousands of them–that need to be effectively communicated to their employees (Reijers, Mans, & van der Toorn, 2009). This section briefly explores the "pain and suffering" consequences of ineffective corporate training as well as the benefits that could be realized if a solution is found to bridge a CALO's learning programs with its policies and processes.

Reduced Efficiency

In 2004 organizations in the United States with 100 or more employees spent $51.4 billion on formal training programs with close to $20 billion dedicated to end-user training programs alone (Gupta & Bostrom, 2006). With such a large investment, it is easy to see why CALO executive managers are concerned about the effectiveness of their training programs. However, it is not just the cost of the training itself that should concern them. Ineffective training produces ineffective employees, and lost productivity can cut deeply into the corporate bottom line.

Barnum (2002) evaluated how "a small inefficiency can lead to heavy costs" (p. 23). She presented a way to calculate the cost of time wasted on an inefficient procedure. This straight-forward calculation may also be adapted to account for the cost of time wasted due to ineffective training.

For example, if a CALO pays its employees an average of $15 per hour then it is effectively paying 25 cents for every minute of the worker's time ($15 divided by 60). If the worker wastes 10 minutes unsuccessfully trying to complete a business process for which he was not adequately trained, then the organization loses $2.50 (25 cents times 10). For a CALO with

10,000 employees who need to perform the same business process, $25,000 in productivity may be lost on a *single* business process that was attempted *once* during a *single* day ($2.50 times 10,000). Multiplying this cost by 250 workdays per year, it is clear that the main training problem – that is, the disconnect between the business process and its training content – would now be affecting the CALO bottom line at a rate of $6.25 million annually. Not surprisingly, researchers such as Newton and Doonga (2007) agreed that "…there is a need for better metrics that demonstrate the link between training and bottom-line business results (p. 117)."

Secondary Effects

Lost productivity is not the only casualty of ineffective training. Sorenson (2002) examined the factors that influence the design and implementation of training programs and found that reduced efficiency is only one consequence of ineffective training. He noted that organizations also must spend money on system repair and maintenance due to the actions of untrained employees. Untrained employees are more likely to break equipment, lose information, or crash computer systems by not following the correct business processes. Naturally, CALOs would assume higher risks of business process mistakes.

Sorenson (2002) suggested that other significant costs are associated with accidents that could be prevented with the appropriate training. Such costs include worker's compensation claims, lost time on the job, and liability lawsuits. Also, when a serious mistake is committed by someone employed by a CALO, the CALO's training program is inspected carefully to determine whether the organization provided adequate training in policies and procedures. A finding of insufficient training can harm the CALO's reputation and bring civil penalties. Finally, an untrained employee could cost a CALO thousands of dollars in fines or sanctions for noncompliance with regulatory agency requirements. It is plain to see how the costs of an inadequately trained workforce can have a serious impact on a CALO's effectiveness, morale, reputation, and bottom line.

Benefits for Policy and Practice

It is difficult to overestimate the benefits of correcting the gap between the development of a CALO's myriad LOs and the management of its OPs and BPs. For purposes of discussion, consider a CALO such as the FBI. The effort to improve training management directly would track with its strategic management objectives and facilitate its compliance with Federal Law Enforcement Training Accreditation (FLETA) standards by mapping OPs and BPs to the instructional objectives in its LOs. Enabling the effective creation and management of LOs also would enhance the FBI's adjunct faculty program, which requires a centralized repository of validated training content.

For FBI curriculum managers, the ability to organize LOs electronically by OPs and BPs means they would be able to locate training products that have been outdated by changing policies, such as the Domestic Investigation Operations Guide (FBI, 2011). For executive managers, the policy-to-training mapping of a KM system would enable them to respond to data calls regarding how training programs specifically are addressing policy-driven enterprise initiatives, such as those requiring that intelligence be integrated into training received by new agents and new analysts. In general, finding a solution to the current KM problem would promote more accountability of training across the enterprise, especially as the Training Division seeks to assume all training responsibilities for the FBI.

Contribution to the Body of Knowledge

The Business Process Modeling Language (BPML) has been standardized for business process modeling application development (Fischer, 2002). This authoring language enables the creation of software that electronically manages BPs. However, although the modeling language provides a powerful structure for defining activities that need to be accomplished within a CALO (i.e., their business processes), it is not mapped to any educational paradigm in the literature.

For the past several years, reusable learning objects increasingly have been used to chunk educational material into smaller units for content management in online learning environments (Beck, 2005). This standardization allows instructional designers to create learning modules that can be edited, reused, and shared among compatible systems. However, neither the computer managed instruction (CMI) standard developed by IEEE (see http://www.ieeeltsc.org/working-groups/wg11CMI/CMIdescription) nor the Sharable Content Object Reference Model (SCORM) specification developed by the Advanced Distributed Learning Initiative (see http://www.adlnet.gov/Technologies/scorm/default.aspx) has addressed the issue of mapping lesson content to policies or business practices.

To summarize, the current BPM authoring standards do not have a means to link to learning objects, and the current learning object authoring standards do not offer a way to link to business processes. The research in this paper seeks to establish the theoretical basis for ensuring that business processes and polices are adequately tracked to learning objects. By linking multiple areas of the KM domain this research makes a significant contribution to the field of instructional technology.

Literature Review

As described in the previous section, the difficulties in communicating a CALO's business processes or synchronizing its business processes to training products may add up to ineffective training programs. What makes this exercise so challenging? Through a review of existing research, this section briefly explores current learning theory, knowledge management challenges, research that relates training programs to business processes, and recent attempts to transform business processes into learning products.

Learning Theory

Learning theory offers an explanation of how business processes are defined and communicated to an organization. In effect, the exercise of codifying a CALO's business processes is an attempt to document a particular knowledge schema. According to cognitive information processing theory (Driscoll, 2005), a schema is a mental depiction of a person's memory, which includes learned concepts and how they relate to each other. BP models are similar to concept maps described by Chang and Chang (2008), whereby modelers "build a solid schema and visual representation of a set of abstract concepts" (p.18). Those who model the business processes of a CALO are depicting their own understanding of the organization's business processes as interpreted from relevant organizational policies.

Unfortunately, different employees who work in the same job will have differing schemas as to how all of the job's events, activities, and tasks are related to one another. This diversity in perspectives creates difficulties for training new business processes because instructional designers use learners' existing schema to relate the new knowledge to previously learned knowledge (Driscoll, 2005). When the corporate understanding of existing processes is not certain it is difficult to define instructional goals and objectives for training the new business processes. A CALO's business process (or job task) training program will suffer as a result.

One way to help a CALO understand what its workforce needs to learn is to conduct a training needs analysis (TNA). O'Brien and Hall (2004) noted that online learning environment (OLE) solutions created by third-party vendors are seldom adequate to meet the unique needs of individual organizations that wish to communicate their current business processes to their employees. They surveyed and visited organizations in several countries and examined user manuals, training materials, and organizational data to inform their development of a general TNA approach. As an outcome of their development study, they proposed a TNA design methodology that helps CALO managers identify training needs and instructional levels in their own organizations.

The recommended design structure includes up to four training levels depending on the type and size of the organization. Level 1 identifies general training that is relevant to every employee in the company. Level 2 training pertains to the employee's level in the organizational hierarchy, such as an executive manager, supervisor, or general worker. Level 3 incorporates training needs that are relevant to each functional area or department of an organization. Finally, Level 4 identifies specific training that is required for each job role, such as a file clerk or salesperson. The authors suggested that their TNA templates not only will enhance an organization's TNA process, but also may improve its efficiency in training as their online learning requirements will "meet their individual requirements using cost-effective methods and just enough training techniques" (p.939).

Perhaps the biggest advantage of the TNA model presented by O'Brien and Hall is that it allows a CALO's own subject matter experts (i.e., its employees) to define the training needs. This strategy is contrasted with hiring an outside e-learning "expert" who is neither familiar with the company's business processes nor will be available to address the adaptation of any future training needs.

Instructional designers with large organizations may feel that conducting a TNA is an overwhelming task. Fortunately, O'Brien and Hall provided a way to "eat the elephant." Their approach is flexible enough to be applied to a variety of organizations and facilitates the development of reusable learning objects for learning management systems. This may be an approach worth pursuing in the attempt to map a CALO's business processes to its training program.

Knowledge Management Challenges

The difficulty of connecting a CALO's business processes and policies to its training strategy fundamentally is a problem of knowledge management (KM). Human beings carry around a vast amount of knowledge–personal experiences, best practices, academic ideas, and physical techniques–which have been collected and internalized over many years. This type of knowledge is referred to as tacit knowledge (Turban & Aronson, 2001) or implicit knowledge (May & Taylor, 2003), and it describes an understanding of reality that has not yet been processed for common use. Keeping knowledge in this form creates difficulties in transferring the knowledge to others; therefore, when an employee leaves an organization there is significant knowledge drainage (Wei, Hu, & Chen, 2002). It is important for implicit knowledge to be converted to explicit knowledge, which can be codified and documented for sharing (Turban & Aronson, 2001; Wickramasinghe, Fadlalla, & Sharma, 2004). This knowledge conversion is the very essence of business process management (BPM) efforts.

Regarding the two types of knowledge, May and Taylor (2003) highlighted an important technical challenge in KM—the labor-intensive process of 1) converting implicit or tacit knowledge to explicit knowledge, 2) transferring the knowledge to where it is needed, and 3) internalizing the knowledge (converting it back to implicit knowledge). For structuring knowledge and building a framework for knowledge sharing May and Taylor (2003) advocated the use of what they call "patterns." These information templates provide a standardized way to identify the context, problem, forces, solution, rationale, resulting context, and related patterns for each "knowledge fragment" (p. 97). Providing the level of detail required for each pattern requires a significant investment of time and resources. However, applying a simplified version of the pattern concept may help create highly contextualized knowledge needed to build components of a CALO's business process models.

After an organization embarks on a BPM or KM initiative it will face daunting technical challenges as it attempts to make knowledge available to everyone in the organization. Turban and Aronson (2001) stressed that individual and organizational knowledge must be captured, evaluated, processed, shared, and renewed, and that each of these exercises requires a mechanism to carry it out.

Hlupic, Pouloudi, and Rzevski (2002) took a hard look at organizational KM, and during their descriptive research they investigated different approaches to researching KM issues. Specifically, they examined how organizational "soft" issues interact with technical "hard" issues. Through a meta-analysis of the KM literature, they argued that "research in knowledge management should reflect this synergy of organizational and technical issues" (p. 97). While developing their framework for KM research, they recognized at least three categories of KM tools, those needed to generate knowledge, codify knowledge, and transfer knowledge between people in an organization.

However, the researchers did not address the role of KM tools in the specific context of corporate training programs. In that context generating knowledge refers to a CALO's developing new BPs and OPs. Codifying knowledge takes place as BPs are modeled and OPs are documented. Transferring knowledge refers to communicating BPs and OPs to the workforce through corporate training LOs. These activities define the features and functionalities of KM tools that could be used to address the corporate training problem. However, as stated previously, the complexity of coordinating a CALO's training programs with its KM efforts makes it unlikely that a "one size fits all" technical solution will be found (Kay, 2003). Each organization must find the KM tools and techniques that are compatible with its policies, capabilities, and budget.

Relating Business Processes to Training

Training products must accurately reflect the current business processes and policies of an organization or they cease to be relevant and effective. In their development study, Caetano, Pombinho, and Tribolet (2007) argued that organizations do not have the ability to map employee competencies to their business processes. As an organization changes, job skills may be lost as employees transition from one section of the organization to another. According to these authors, *competency-based management* is a paradigm that focuses on individual employees as organizational assets whose talents and skills may be codified and exploited for the benefit of the organization.

The authors noted that current methods of documenting an organization's activities, such as Business Process Management Notation (BPMN), may accurately represent how an organization functions, but they do not contain any information regarding the competencies needed to accomplish those activities. The researchers suggested that competencies serve to "classify human actors according to their ability of performing tasks in a specific environment. They are the manifestation of knowledge attained through the performance of an action" (Caetano, Pombinho, & Tribolet, 2007, p. 1258).

A competency modeling approach is presented in their article. It treats competencies as reusable collections of nodes and links (representing nouns and verbs, respectively) that describe what needs to be done to accomplish a certain business process. An individual competency, which is a subset of the larger collection, is then mapped for each employee. The goal of this exercise is to develop a competency model for each employee so that s/he can be recruited within the organization to be assigned to an appropriate activity or project. The researchers developed a web-based system, which they report is being tested to validate their model.

One interesting aspect of their research is their attention to the problem of granularity. They observed that "on the one hand, a high-level, coarse-grained representation will not provide enough information. On the other, if it is too detailed, the entire representation process may become compromised, as it is effort and time-consuming" (p. 1258). This analysis seems to address the same concerns as those of business process modeling. In fact – although it was not suggested in the article – using a similar competency-based method may yield a way to map training programs to competencies, and thus to business processes.

Hawryszkiewycz (2005) recognized the fact that an organization's business processes are continually changing, and that people must be taught how to adapt to these changes. He suggested that training at work is the most effective way to communicate the new processes, but noted that most educational programs are aimed at training large numbers of organizations on a limited number of goals. A customized approach is needed to enable employees to learn what they need to know – and only what they need to know – within the context of their work activities. In other words, employees need to be trained on the specific business processes for which they are responsible.

Hawryszkiewycz's framework aimed to create a learning space that is more effective than a traditional learning object, which he defined as "an integrated set of subject material together with its supporting services" (p. 23). His *learning space* approach integrates the subject material (i.e., the business process) with support tools while providing access to subject experts. After the learner's knowledge gap is identified, s/he is presented with an online learning space whereby various learning activities may be chosen to address those knowledge gaps.

The proposed workspace for learning activities is designed to help organize the training resources necessary to address an organization's changing business processes. However, the paper does not address how the components of those training resources are modified in order to coincide with the business processes. A tracking mechanism is still needed to ensure that the online learning resources represent the most current business processes of the organization.

The performance of a CALO's business processes often requires specialized training, even certifications. David, David, and David (2011) conducted a content analysis of corporate job descriptions, business student resumes, business course syllabi, and business textbooks to determine the extent to which business schools are "offering students what it takes to get a job" (p. 51). The researchers first created an inventory of 140 skill sets based on 200 job descriptions for various business tracks. Then they examined 200 resumes of business students nearing graduation and found that 95% of the resumes did not mention any of the skills or certifications identified in their inventory. To determine why there was such a disparity between skills needed and skills obtained they looked at 100 business course syllabi and 20 textbooks representing the five business major areas. Almost without exception, neither the syllabi nor the textbooks covered any of the 140 skills or certifications required in the corporate job descriptions.

The primary lesson learned from the business school study may be applied to any CALO: Those who train workers need to focus less on theory and more on specific skill sets required for the desired job. The authors offer their skill set inventory as a starting point to help address needed reforms in business schools. However, a more effective solution would be to provide the educators with a method to track the ever-changing corporate skill sets with the courses and instructional materials that are being delivered. In other words, they could benefit from a KM system that could track BPs to the LOs that are meant to teach them. In this way, educators in both business schools and CALOs would be able to proactively identify training gaps between what is practiced in the field and what is being taught.

Transforming Business Process Content into Learning Content

There are surprisingly few documented attempts in the literature to address the transformation of business process models to usable learning content. Virine and Rapley (2003) proposed the unification of business process visualization techniques to aid in understanding business process models, but they do not attempt to tie business processes directly to the training of those processes. Moreover, they admitted that their proposed methodology "cannot be applicable to comprehensive and large scale decision and risk analysis involved [in] a large number of steps and alternatives" (p. 1781). In other words, their methodology is not applicable to CALOs.

In their development research study, Reijers, Mans, and van der Toorn (2009) attacked the problem of unwieldy business process models by proposing a system of aggregating complicated BP models for an organization. By exploiting common elements of the models, they suggested that fewer BP models need to be maintained and will require updates less frequently. They developed a modeling language that allows for the on-demand extraction of specific BP models from aggregate BP models, thereby giving users the ability to view the relevant segment of the overall BP universe. While the researchers acknowledged that their design has not yet been empirically tested, their approach seems to provide a useful way to characterize events and functions within BPs. However, the researchers did not attempt to connect those elements to anything related to training. Indeed, the need to connect BPs to corporate training – specifically, to learning objects (LOs) – may have been overlooked because "the management of process models as a discipline is at its infancy in comparison with fields like product and software management" (p. 241).

Jay Cross (2003) of the Internet Time Group proposed a strategy called "Workflow-based eLearning" that reduces the "lag time" between the assignment of a business process task and the training of a worker to complete the task. Such a mechanism would be helpful in strengthening the relationship between a CALO's business processes and its training products. In fact, Cross claimed that his "workflow optimization products" actually were designed to reduce the need for training.

According to Cross, these tools make computer-based business processes and the corresponding online training available in the same environment. But while his system may be useful for computer-based business processes, it does not seem applicable to physical business processes or to existing legacy training products. Furthermore, no research has been found to validate any of the products that Cross describes, and no peer-reviewed articles have been found to be authored by Cross. The ideas he presents have merit, but need to be validated with further research.

Zhang and Su (2007) call for harnessing the read/write power of Web 2.0 technologies to create "living educational materials" (p. 153). To encourage real-world learning in computer science students they built BRIDGE, an online system comprised of three components. One component allowed the students to collaboratively annotate open-source software. A second component enabled the creation of student-authored educational modules that connect computing theories with industrial practices. The third component of BRIDGE was a course management system (CMS) that was used to build computing courses from the modules. Although the BRIDGE system was successfully launched in a computer science education environment, its implementation has not been formally evaluated. Further research is needed to determine if such a model can be successful.

Despite its relatively new appearance in the literature, the challenge of tying together theory and practice with flexible learning materials is a goal worthy of pursuing. A model similar to the BRIDGE concept could be developed to bridge organizational policies to business practices through collaboratively-developed learning objects.

Raybould (1995) introduced an organizational learning model whereby an organization designs a knowledgebase, initiates performance based on that knowledge, achieves individual learning as a result of the performance, generates new knowledge based on the learning, and then captures that knowledge for the organization's knowledgebase. This process, which Raybould called the Organizational Performance/Learning Cycle, seems to be an effective framework that could be used for managing policy and business process knowledge for CALOs.

To implement this model, Raybould (1995) suggested the development of an electronic performance support system (EPSS) that "…captures, stores, and distributes individual and corporate knowledge assets throughout an organization, to enable individuals to achieve required levels of performance in the fastest possible time…" (p. 11). Although he advocated strongly for electronic knowledge management, Raybould also recognized that legacy training products include both electronic and non-electronic components. To maximize the use of the latter he suggested "…non-electronic components should be indexed, managed, and coordinated electronically wherever feasible, or we are not improving on the ad-hoc, unorganized manual systems that have contributed to the problems of information overload" (p.10). Managing information in legacy, manual training materials is one of the primary concerns of any training program overhaul project.

Future Considerations

The research documented in this paper established the relevance, significance, and literature basis for the problem of mapping LOs to BPs and OPs, and in doing so it raised important training management issues for CALOs. The knowledge gained here could serve as the theoretical foundation for a future study, the goal of which would be to define software requirements for a KM system that enables a CALO's training content to be mapped to its business process models and organizational policies.

A major concern of those who manage KM projects is that the definition of requirements is expensive and occupies a large portion of the KM project development schedule–as much as 20-25% of the total project time (Japenga, 2011). However, this issue disappears if the CALO does not already have a training-related KM project underway. Rather, the requirements developed by the future study would help justify a plan to create such a KM system, with the requirements forming the basis of a formal Request for Proposal (RFP).

On the other hand, many challenges could surface during a requirements elicitation study. As requirements are developed to link BPs and OPs to LOs it may be difficult to identify the existing training strategies and instructional materials that already are in use in the organization–sometimes in poorly documented curricula (Raybould, 1995). In addition, business processes and policies are part of a CALO's intellectual capital and are subject to hoarding (Hlupic, Pouloudi, & Rzevski, 2002). Owners of this knowledge may be reluctant to make it available to others in the organization and could exhibit a low level of tool readiness (Sun, 2011) that would prevent them from supporting any type of computer-based solution. Finally, existing instructional materials (e.g., long PowerPoint presentations and web-based training modules) may be related to several BPs and OPs, which would complicate how they are associated to LOs.

The selection of a CALO is a key decision for a future study. A large law enforcement organization such as the FBI may be considered, but it would pose special challenges for the researcher. For example, the researcher would not be able to use any law enforcement sensitive or classified content in the final report. Instead, the study must focus on the mechanism that manages the content and carefully select representative content that would not compromise law enforcement or national security guidelines. Also, famously unsuccessful efforts to automate business processes have haunted law enforcement CALOs like the FBI (Israel, 2012), and these setbacks may cause the workforce to lose confidence in the CALO's ability to successfully build a KM system.

Despite the possible challenges listed in this section, pursuing a KM system to manage training products, policies, and business processes would be a worthwhile goal for a CALO. The benefits of dynamically relating these learning constructs would outweigh any inconvenience or problems experienced by the researcher.

Summary

For CALOs undergoing transition, modifying the training products that correspond to business processes becomes an endless exercise in trying to catch up to the latest policy or technical implementation that affects those processes. Indeed, depending on the CALO's corporate training approach, every change in a policy or business process could require a corresponding change in a policy manual, job aide, user guide, online help section, PowerPoint presentation, computer-based training module, instructor-led training unit, or train-the-trainer component. Hundreds of business process changes easily can breed thousands of adjustments to training products, and the chore of making these adjustments is often attempted without a mechanism to find exactly where in the curriculum the changes need to be made. For large organizations the coupling of policies, processes, and training content is no trivial task.

The goal of this paper was to shed light on the significant problem caused by not coordinating a CALO's learning products with its policies and processes. Large-scale training programs are difficult to create and manage, and their failure could be very costly. The research presented in this study will assist CALOs in developing strategies to analyze their organizational policies, define their business processes, and map them to meaningful instructional content used to teach those concepts to their employees. It is meant to contribute to the existing knowledge and practices of business process modeling, knowledge management, and instructional systems design.

Chapter Three

Elicitation Study

Abstract

Large organizations have trouble keeping their training programs aligned with their business processes and organizational policies. Thousands of lesson plans are often stored as static documents, which are neither readily searchable nor linked to approved processes and policies, rendering corporate training programs inefficient and outdated. A well-designed knowledge management (KM) system would be able to meet this technological need, but generating software requirements for such a system is typically a time and resource intensive process.

A unique combination of e-mail-based surveys, qualitative data-coding techniques, and Soft Systems Methodology (SSM) was employed to elicit a robust set of user needs without the need for face-to-face meetings. Employees of a large government agency were asked to tell their stories about the current problems they experienced in training management as well as their ideas for future solutions to those problems. Common themes were identified in their responses and then categorized. The resulting goals and preferences for the KM system were documented in a set of conceptual diagrams that represent the participants' view of the current and future state of affairs. These models represent the collective user needs from all the participants, and will serve as artifacts to be used for future requirements definition, validation, and prioritization.

Keywords

Requirements elicitation, user needs, business process management, knowledge management, corporate learning, systems development, instructional systems design

Introduction

Large organizations that manage hundreds of business processes typically are governed by an even greater number of policies. To increase efficiency some of these organizations have implemented business process management (BPM) strategies, which seek to eliminate workflow bottlenecks and wasteful tasks (Trkman, 2009). Efforts to manage business processes, however, may be wasted if employees are not trained adequately on those processes and the policies that authorize them to perform those processes. In fact, connecting all three of these areas–training, processes, and policies–is a task often neglected by large organizations (Hawryszkiewycz, 2005), which results in ineffective and irrelevant training programs.

Coordinating corporate training programs with processes and policies is a problem that is well-established in the literature, and the proposed solutions predictably involve some sort of knowledge management (KM) system. For example, Zhang and Su (2007) argued that the variety and scope of business processes require careful attention to matching online learning tools to training needs. Capuano, Gaeta, Ritrovato, and Salerno (2008) proposed a technology-enhanced learning (TEL) system because they recognized that “a better integration of TEL with business process management is, in fact, one of the greatest challenges for today’s knowledge management” (p.56). Teaching a large number of processes and polices to a large workforce naturally has pushed organizations to seek a technical solution.

Indeed, organizations have spent billions of dollars to move their training to online learning environments (Gupta & Bostrom, 2006). However, many corporate learning initiatives concentrate on tasks that are common across the workforce rather than on more granular, job-specific responsibilities (Hawryszkiewycz, 2005). O’Brien and Hall (2004) found that online learning solutions created by third-party vendors seldom are adequate to meet the unique needs of individual organizations that wish to communicate their current business processes to their employees. Likewise, Newton and Doonga (2007) reviewed several corporate e-learning initiatives and concluded that corporate e-training did not address strategic business objectives. Martin, Leyking, and Wolpers (2008) summarized the corporate training problem succinctly: "Existing eLearning approaches miss consistent alignment with business operations and objectives" (p. 1).

Typical e-learning solutions fail to meet corporate training needs because they fail to connect training content to an organization's processes and policies. Some have attempted to map online learning to business processes (Cross, 2003; Capuano, Gaeta, Ritrovato, & Salerno 2008) or to map business processes to competencies (Caetano, Pombinho, & Tribolet, 2007). However, there is no published work that explicitly addresses the mapping of training products to both the business processes that are taught to the workforce and the policies that govern the execution of those processes. Clearly, a new KM system is required to meet this need.

The goal of the current study was to elicit user needs in preparation for defining software requirements for a KM system that manages the linkages between an organization's training products, its business processes, and its organizational policies. The subject chosen was the Federal Bureau of Investigation (FBI), a large organization that continues to struggle with the challenge of changing its business processes and policies in the face of an expanded national security mission and a new case management system (Miller, 2005). Since the terrorist attack on September 11, 2001 the FBI has responded to a great deal of public scrutiny of the agency's efforts to automate its business processes (Israel, 2012). In addition, post 9-11 information overload and high-pressure decision-making policies have contributed to what Krause (2012) calls "vigilance fatigue" (p.3), a condition that weakens security awareness. As a result, the FBI's capacity to manage the training of new processes and policies has received even less attention.

An expert panel of FBI training professionals was used to develop a natural language description of a KM system that could manage training content by linking to processes and policies, while continuing to support usability and facilitate compliance with Federal Law Enforcement Training Accreditation (FLETA) requirements. This user-based description is the essential first step in defining the system goals and preferences (Liaskos, McIlraith, Sohrabi, & Mylopoulos, 2011) from which software requirements are written. The remainder of this paper is organized as follows: A set of definitions is provided to clarify the scope and context of the research. Next, the purpose of the study and research questions is introduced, followed by a review of the literature and a discussion of methods used. Finally, a summary of the research findings is presented, along with a summary and considerations for future work. Appendix A is provided to assist the reader in recalling pertinent acronyms and definitions.

Definitions

Coordinating training programs with business processes and organizational policies is one of the most difficult challenges of a large organization. The current study focused on the training programs of the FBI, but its results are generalizable to all large organizations that manage several thousand employees and dozens of job families. For the sake of readability, this type of large organization will be referred to as a CALO, defined by Kiper (2008) as "a Company, Agency, or other Large Organization" (p. 14). Examples of CALOs exist in the public sector (e.g., U.S. State Department and the Drug Enforcement Agency) as well as in the private sector (e.g., Microsoft and Coca-Cola).

A business process (BP) is a segment of defined activity in an organization. It is a set of tasks that logically are related to fulfill an organization's objective (Trkman, 2009). Through the use of symbols and arrows, BP diagrams define how work is done and who does it. Due to its size and diversity, a CALO's workforce participates in hundreds of processes every day (Reijers, Mans, & van der Toorn, 2009). In the FBI Training Division each person's contribution to a business process is referred to as a job task (JT), which is the entity used to map to instructional materials used to train that person. To avoid confusing study participants with an unfamiliar term, JT was used in place of BP throughout the study.

An organizational policy (OP) refers to a CALO's documented guidance to the organization. OPs articulate outcomes and strategic goals that are, in practice, fulfilled by BPs (Trkman, 2009). OPs may be produced internally by the CALO or adopted from a higher authority such as the CALO's parent organization. Statutes, regulatory guidelines, employee handbooks, and standard operating procedures are examples of OPs. An OP may take the form of an electronic or physical document and convey the specific rules, protocols, or directives approved by CALO executive managers. The FBI manages several hundred OPs through its Policy and Guidance Library (FBI, 2011).

Finally, a learning object (LO) is a construct used to chunk educational material into smaller units for content management in online learning environments (Beck, 2005). An LO may be defined broadly as a container of learning materials that manages instructional content and resources maintained by an organization. However, like other CALOs, the FBI typically manages this type of information in the form of a lesson plan (LP). And while an LO is probably a more accurate term to describe this concept in the anticipated KM system, to avoid introducing a new term to the study participants, LP was used to refer to the anticipated collection point for JTs, OPs, and instructional products—PowerPoint slides, job aids, and demonstrations—needed to communicate the JTs and OPs to the learner.

Purpose

As noted previously, the FBI needs a robust KM system to coordinate its training content with its processes (JTs) and policies (OPs). To build such a system it must first collect the user-defined needs, and collect them in a way that may be transformed into software requirements.

Requirements engineering is a branch of software engineering that involves the elicitation, modeling, analyzing, communicating, agreeing, and evolving requirements for software (Nuseibeh & Easterbrook, 2000). The primary focus of this study was on the *elicitation* process, which employs the techniques used to capture software requirements from a selected group of stakeholders in a CALO.

The overall goal was to develop a rich set of user needs that describes a KM system by which FBI training content may be mapped to its job tasks and organizational policies. To attain this goal, the following pair of questions were addressed:

- What KM processes and technologies are required to associate LPs with JTs and OPs?
- How will user needs be defined for a KM system that will manage LPs, JTs, and OPs?

These research questions effectively establish the system boundaries (Nuseibeh & Easterbrook, 2000) that guide the process of defining the problem scope and requirements for such a system.

Background

Nuseibeh and Easterbrook (2000) suggested that "the primary measure of success of a software system is the degree to which it meets the purpose for which it was intended" (p. 37). Defining software requirements–statements about how the system will function–is the most effective way to ensure that goal is met. The process for defining those requirements is known as *requirements elicitation (RE)*, which Saiedian and Dale (2000) defined as "the specific processes of gathering, determining, extracting, or exposing software requirements" (p. 420).

Japenga (2011) distinguished between *system* requirements and *software* requirements. The former sets forth specifications about the entire information technology system, which would include hardware, software, and possibly networking components. The latter focuses exclusively on the features and functionalities associated with the software, because that is where the end-users interact and where most of the complexity is encountered. For the purposes and goals of this study, RE techniques will be used to generate user needs and software requirements, which will refer only to the natural language descriptions of the envisioned KM system and not to the more technical statements used to bind software developers to activities specified in contracts. Regardless of the technical granularity one chooses for software requirements, there is a considerable amount of literature-based guidance on how to best develop them.

The Institute of Electrical and Electronics Engineers (IEEE) offers many types of standards for hardware and software developers (see www.ieee.org). Their most recent set of recommendations regarding software requirements specification (SRS) is contained in the document known as *IEEE STD 830-1998* (IEEE, 1998). In this standard, the IEEE presents several benefits for developing software requirements, which include establishing common agreements between customer and developer, reducing development effort, facilitating estimating costs and schedules, and providing the basis for validation and compliance testing. IEEE also suggested several issues that should be covered in SRS development, such as functionality, attributes, and design constraints.

Notwithstanding all of its guidance regarding the characteristics of requirements, the IEEE standard offers very little in terms of *how* the requirements should be developed. It simply recommends that customers (users) and suppliers (developers) should work together on defining requirements, so “the functionality, interfaces, performance, and other attributes and constraints of the software are not predefined, but rather are jointly defined and subject to negotiation and change” (p. 14). Fortunately, there are many studies that have suggested best practices for accomplishing the RE task. In general, they have focused on two areas: the *people* involved and the *techniques* used .

The two primary groups of people involved in RE are the customers and the developers. Saiedian and Dale (2000) identified customer stakeholders as those who pay for the system (buyers), those who understand the organization's problem addressed by the new system (domain experts), those who will maintain the system on behalf of the organization (software maintainers), and, finally, those who will actually use the system when it is built (end users). On the developer side, there are those who oversee the project (project managers), those who identify and document the requirements (requirements engineers), those who provide design constraint expertise (software engineers), and those who evaluate the system (testers). For ease of reference, this paper refers to these groups as simply *customers* and *developers*.

The costs of not conducting effective RE are well documented. Laporti, Borges, and Braganholo (2009) report that ineffective RE processes account for 55 percent of a system's technical problems and 82 percent of the effort in correcting mistakes. Davey and Cope (2008) claim that 71 to 90 percent of failed software projects can be attributed to poor RE and to the mismanagement of requirements. Failed and abandoned projects are extremely costly to a CALO and – if it is a government organization – to the U.S. taxpayer.

One may wonder why RE is so difficult. One problem is that the two groups may speak different languages. Users are accustomed to expressing themselves in natural language, whereas the developer may prefer the more precise syntax of technical specifications (Laporti, et al., 2009). Another major problem lies in the fact that different stakeholders, even those who reside on the same side of the customer-developer aisle, have differing goals (Nuseibeh & Easterbrook, 2000). For example, a CALO's buyers and managers may be focused on budget and accountability, while its users are concerned only with functionality and features of the system. On the developer side, a project manager may want to influence requirements based on schedule constraints, while the requirements engineer places a higher priority on meeting the customer's needs. With such a diverse range of goals among stakeholders, it is difficult for any side to speak with one voice.

A close working relationship between customers and developers is essential to the success of the RE process, and, ultimately, to the success of the project. How they interact with each other is defined by the selected methodology. Chakraborty, Sarker, and Sarker (2010) reviewed more than a dozen studies that propose RE methodologies and noted that the majority of them do not attempt empirical validation of their findings. Rather, the studies focused on the elements of various RE methods as a way to generate requirements by following a set of prescribed steps. Moreover, the steps identified in these studies seemed to fall into a familiar trend – they began to mimic each other as well as development methodologies in unrelated fields. For example, the activities of *identification, conceptualization, formalization, implementation,* and *testing* presented by Byrd, Cossick, and Zmud (1992) seem to echo the phases of *analysis, design, development, implementation,* and *evaluation* that comprise the ADDIE model of Instructional Systems Design (Martínez-Ortiz, Sierra, & Fernández-Manjón, 2009).

The empirical studies reviewed by Chakraborty, et al. (2010) were qualitative in nature, seeking to uncover dimensions of RE based on observed patterns of interaction behavior among stakeholders. The problem with these studies, according to the reviewers, was that they represented stand-alone research rather than offering a unifying framework that could represent all elements of RE found in the various studies. Chakraborty, et al.'s solution was to analyze the RE process from the perspective of knowledge sharing, trust, and development of shared mental models using a grounded theory approach. By collecting (primarily interview-related) data from two very different organizations, the researchers uncovered RE characteristics that were common to both organizations and, presumably, generalizable to other organizations.

Chakraborty, et al. (2010) identified four “states” of the RE process: *scoping, sense-making, dissension,* and *termination*. In each of these states, representatives of the customers and developers participated in interactions that involved *objectives, knowledge transfer, trust,* and *mental models*. While one might expect that the RE process would progress from one state to the other in a linear fashion, the researchers actually found that the participant interactions would cause them to skip around and revisit states depending on “triggers” that were also identified during the study. At certain states, specific RE techniques were invoked to mitigate the negative effects of certain interactions or to facilitate a transition to a different state. Issues with objectives and trust factors, for example, might cause the RE process to move from a *sense-making* to a *dissension* state where those issues would be addressed with a group consensus technique. Eventually, the participants would want to arrive at the *termination* state, which is typified by a clear set of requirements, explicitly codified knowledge, a high level of trust, and a shared frame of reference. In this way, the process model explains user-developer interactions in the RE life cycle that are independent of specific techniques.

Techniques–or variations of techniques–must be employed to elicit requirements. The RE techniques described in the literature may be categorized into several major types. Nuseibeh and Easterbrook (2000) referred to *traditional techniques* as those that employ surveys; questionnaires; interviews; and the review of organizational policies, guidelines, manuals, and other existing documentation. *Group techniques* such as brainstorming, focus groups, and Joint Application Design (JAD) seek to build consensus among stakeholders (Coughlan & Macredie, 2002). Other techniques include *modeling* (requiring detailed pictures), *prototyping* (requiring a functioning mockup of the system), and *contextual* techniques, requiring the developer to observe user job tasks in their natural setting (Nuseibeh & Easterbrook, 2000).

While several studies in the literature have recommended certain sets of RE techniques, there are fewer that attempted to research the effectiveness of those techniques, complained Davey and Cope (2008). However, one such study (Coughlan & Macredie, 2002) involved a comparison of four common RE methods: MUST (a Danish acronym for initial design theories and methods), Joint Application Design (JAD), User-Led Requirements Construction (ULRC), and Soft Systems Methodology (SSM). The study found certain methods, such as MUST and SSM, involve close communications between customers and developers and employ a wide variety of techniques. On the other hand, JAD is "intensely group-focused and much responsibility is placed on the skill of the facilitator to direct the session" (p. 67). While using the ULRC method, most of the burden of modeling requirements lies with the users. With MUST being a relatively new methodology and SSM enjoying a 30 year track record in research, one is left with the impression that SSM may be the more prevalent and therefore more validated methodology to consider.

SSM was not developed exclusively as an RE technique, but rather as a problem-defining methodology (Checkland, 1998). However, according to Coughlan and Macredie (2002), the SSM approach is particularly well-suited to the RE process for complex systems because it calls for representing the problem in pictorial form. As the RE activities progress, SSM guidelines call for other conceptual models to be used for contrasting the desired system with the current system. These illustrations are sent to participants throughout RE to spark discussions and debates regarding the requirements. As the visual models mature, they are used to refine the requirements in the final stages of RE.

Although Coughlan and Macredie (2002) provided a thorough analysis of available RE methodologies, they recognized their study was theoretical in nature and called for more research in realistic settings. To achieve results in such a practical setting, Laporti, et al. (2009) attempted to develop, and then validate, a unique RE method. Their study was based on the premise that "On one side, users and clients prefer natural language to express their needs; on the other, analysts prefer a more formal, less ambiguous language…" (p. 367). The goal was to transform the natural language *stories* provided by the customers to the more rigid and technical format of *use cases*, a commonly used tool for capturing software requirements. They noted that RE techniques such as extreme programming and JAD focus too much on the role of the analyst (developer) rather than on that of the user (customer). The roles and structure defined by these two techniques actually inhibit participation and collaboration among stakeholders.

Rather than attempting to extract knowledge from customers via rigid surveys or questionnaires, Laporti, et al., enabled them to tell stories of their experiences–good and bad–regarding systems and processes. Each story was then deconstructed into fragments, which described specific activities in the story. The story fragments from various participants were consolidated and transformed into a standardized description format called *scenarios*, which were made available to the customer group for comments and corrections. When consensus was reached, the scenarios were transformed into the standard format of use cases. In this way, the RE process enabled the evolution of requirements from contextualized, free-form stories to structured data in scenarios and use cases, from which formal software requirements may be written. In addition, the researchers were able to track this evolution so that use cases and software requirements may be traced back to the scenarios and stories that inspired them.

Although the development of use cases and formal software requirements are outside the scope of this paper, it is worth noting that Laporti, et al., designed a simple experiment to test the effectiveness of their method. The researchers tasked two groups of six Masters degree level students to elicit the requirements for a system that would sell movie tickets in an online environment. One group was asked to strictly use a traditional interview technique for RE, while the other group was asked to use the new method. The second group developed fewer use cases, but their use cases were more detailed than those of the first group. The differences were attributed to the fact that the activities of the second group involved more discussion, and therefore conflict, which needed to be addressed on a regular basis.

However, the researchers noted that the second group did not strictly follow the story-to-scenario-to-use case conversion rules. To further evaluate the new method, a different scenario was given to a third group who strictly followed the prescribed approach (although their results could not be compared to the first group). Questionnaires given to the participants of the first and third groups indicated that the new model served to direct the discussions and resulted in more accurate and complete requirements. Reflecting on this experiment, the researchers concluded that the conversion process for their method was effective, but difficult and time-consuming. They noted that an asynchronous collaborative method of communication would have made better use of participants' time.

The above examination of the literature uncovered a wide variety of techniques used by developers to elicit requirements from customers. Regardless of the specific techniques used, however, the consensus among all researchers is that RE processes should concentrate on collaborative activities between those who represent the users, or the consumers of the product, and those who represent the developers, or the providers of the product. All RE techniques must focus on the relationship between these two groups (Saiedian & Dale, 2000) so that a common understanding of the envisioned system will lead to a strong and stable set of software requirements.

Methods

The consensus from the literature rejects the idea that a single RE method or technique is appropriate to every situation (Chakraborty, Sarker, & Sarker, 2010; Nuseibeh & Easterbrook, 2000; Saiedian & Dale, 2000). In fact, Coughlin and Macredie (2002) observed, "When methodologies are used, it is more the case that parts of them are used (or parts from different methodologies) rather than following all the steps required by a particular methodology" (p. 68). Consistent with this opinion, the current study adopted a variety of activities from the established RE methods reviewed in the previous section. Saiedian and Dale (2000) justified the use of RE techniques in the design of information systems:

> In order to gain an understanding of the user's work, we try to appeal to the very resources upon which the participants draw to achieve their own understanding of their work. Knowledge of many of these aspects can only be gained from experienced co-workers (p. 421).

The methods employed by the current study were implemented in three phases. First, existing documentation was reviewed to guide the parameters of the study and to serve as a reference for the identification of requirements. Next, FBI stakeholders were identified and study participants were selected based on experience, expertise, and ability to influence the implementation of the proposed system. Finally, data collection was conducted using a mix of techniques tailored to the goals of the study.

Review of existing documentation

While some researchers recommend the reviewing of existing documentation as an RE technique itself (Coughlan & Macredie, 2002; Nuseibeh & Easterbrook, 2000), it may be even more important to review documentation *before* employing an RE technique. Such documentation may inform the questions asked in customer interviews, such as: Which documents containing organizational policies and job tasks are most relevant to lesson plans?

A thorough review of the documentation helped to articulate the constraints of the system to the user, which is a challenging task (Saiedian & Dale, 2000). In this case, the proposed KM system will be constrained by internal security policy, investigative guidelines, U.S. code,instructional systems design procedures, and guidelines published by FLETA (see www.fleta.gov).

Identification of stakeholders

Of course there are advantages to selecting a randomized sample of users to participate. However, there would be no guarantee that those users would have the time, expertise, or willingness to participate. Fortunately, in CALOs it is common to have "user representatives who are domain experts" (Chakraborty, Sarker, & Sarker, 2010). In other words, larger organizations have employees who not only have experience as end users but also are familiar with the system analysis process.

The stakeholder participants were recruited personally from the customer job categories described previously as buyers, domain experts, software maintainers, end users, project managers, requirements engineers, software engineers, and testers (Saiedian & Dale, 2000). The seven FBI employees who agreed to participate included two frontline instructors, who regularly design and deliver training; one supervisory instructional systems specialist, an expert in instructional systems design (ISD); two unit chiefs who supervise instructors and training programs; one unit chief who regularly answers division-level data calls regarding FBI training; and a unit chief with significant technical expertise in learning management systems. These participants not only were knowledgeable of the needs of the FBI Training Division, but also had sufficient status in the organization to influence the implementation of the project if it was approved (Coughlan & Macredie, 2002).

Because the elicitation study described in this paper does not support any current KM system project, there is no developer who has been named or assigned. As a result, representatives from an actual developer could not be included in the RE process. Instead, the researcher acted in the role of the developer and served to collect, analyze, and present the data as an advocate for a future KM project plan. This decision carried a risk of researcher bias, as the researcher has intimate knowledge of both the problems and participants involved in this validation study. However, this risk was mitigated by the application of sound, objective, and literature-based methodologies that governed the collection and analysis of data. The participation of the researcher, who is an onboard employee and future user of the proposed system, ensured a strong contextual understanding throughout the RE process (Nuseibeh & Easterbrook, 2000) and eliminated the "us vs. them" mentality (Saiedian & Dale, 2000) that typifies customer-developer relationships in these types of projects.

Data collection

Saiedian and Dale (2000) observed, "As with any process, elicitation has to be adapted to match the scope of the task, the initiative maturity of the using agency, and the cost and schedule constraints" (p. 426). Informed by those techniques with documented success in the literature, the RE processes followed by this study included semi-structured interview questions, conducted asynchronously, informed by a review of existing documentation, augmented with SSM diagrams, and subjected to group consensus.

Taking into consideration the time constraints, the diversity of stakeholders, and the ability of those stakeholders to fit participation into their schedules, the collection of data involved a series of written interview questions, resembling open-ended surveys, which were delivered by e-mail. Collecting this type of qualitative data via e-mail is becoming increasingly more popular (Gay, Mills, & Airasian, 2009), and in this case proved effective in eliciting user needs from busy people without the need for time-draining face-to-face meetings.

Participants were asked to recount stories about their jobs as they relate to curriculum management, including the problems they encounter on a regular basis. This free-form story telling technique is similar to one used by Laporti, et al. (2009), although their strict procedure of converting the stories to scenarios and use cases was replaced with activities that supported the construction of SSM diagrams (Coughlan & Macredie, 2000) as described below. The participants were asked to substantiate their stories with specific examples of the "pain and suffering" experienced with inadequate curriculum management resources. When required, follow up questions and responses also were sent through e-mail.

The original elicitation and clarification of participant stories were conducted via one-on-one e-mail communication; none of the participants knew who else was participating in the study. The data collection method approximates a typical e-mail survey and was used to avoid undue influence of ideas by other participants. However, after the stories were deconstructed into coded activities (see Analysis below), the activity descriptions–both current and future–were consolidated from those submitted by all participants, and the list was provided to the entire group in the form of illustrative diagrams. This approach followed previously described SSM guidelines (Checkland, 1998; Coughlan & Macredie, 2000). At this point, the participants were asked to respond to the consolidated product with comments and corrections about the activities and models.

Effective communication is "notoriously difficult to achieve and is a recurring problem" during the RE process (Coughlan & Macredie, 2002, p. 48). Asynchronous communication, such as e-mail, provides advantages in terms of response time flexibility for participants and the ability for participants to "think through" their responses, which increases the quality of the responses. The one-on-one e-mail communication method used in the study sparked uninhibited discussions about user needs that led to a common understanding among all participants. As expected, the participants expressed their preference for this method rather than a long series of group meetings. In addition, the e-mail exchanges created a written record of each response, without expending additional resources in recording and transcription. However, these communications were planned very carefully, making the most out of each exchange so that the responses closely tracked with the study's RE goals. Firm deadlines were set on responses, and tactful reminders were required to keep the information flowing.

As Scheinholtz and Wilmont (2011) noted, successful RE "strongly depends on the right questions being asked in such a way that the user stakeholder can provide the right details in his response" (p. 72). This philosophy was used for developing both original questions and follow-up (probing) questions to induce elaborative, explanatory, and resolving types of responses from the participants. The first e-mail message sent to the participants contained the following questions:

1. Organizational policies (OPs) define what we're allowed to do in the FBI. If FBI training products could be linked to policy documents, which policy documents would be the most important?

2. When someone (e.g., FBIHQ divisions, the Director, Inspection Division, Congress, etc.) asks you questions about the curriculum you manage, what kinds of questions are you asked? In other words, on what kinds of training information have you been asked to report?

3. What are the specific challenges (i.e., "pain and suffering") that you've faced when creating/managing content for courses or curricula? What circumstances have prompted you to say, "There must be a better way of doing this"?

4. What are some ways you think an information technology system could help the FBI develop and manage curriculum?

The number of open-ended questions was kept to a minimum to respect the participants' time. However, in order to fully capture their requirements, the questions were crafted in a way that encouraged the participants to express their needs in both a negative (i.e., what is currently wrong) and a positive way (i.e., how things could be better). The questions also were meant to elicit procedural skills as well as explicit knowledge, as suggested by Scheinholtz and Wilmont (2011). The wording of the questions was based on personal conversations with the participants during their recruitment, thus ensuring the questions would be understood clearly. The analysis of their responses will be discussed in the following section.

Analysis

As with other qualitative studies, the free-form data collected was first subjected to a content analysis whereby patterns and common themes among written submissions were identified. Next, following the approach offered by Laporti, et al. (2009) the stories contained in the e-mail responses were deconstructed into fragments that describe specific activities. A total of 37 "Activity Themes" emerged from the story fragments to reflect the shortcomings of current processes as well as the desired features for a future KM system. Finally, from these Activity Themes six categories of KM features were identified: Access, Reporting, Accountability, Development, FLETA Requirements, and Versions. As described in the next section, the results of the coding and categorizing efforts were checked for accuracy by the participants.

Table 1 provides a small sample of story fragments, quoted directly from the participants' responses, along with the corresponding activity themes and feature categories. In addition, each story fragment was tagged with a number (indicated in parenthesis) that corresponds to a particular participant. This numerical assignment became an indicator of the variety of participation from individuals as well as the uncanny convergence of issues brought forth without collaboration among participants. It also provides a means to track support for each identified need back to the original raw expression of that need.

Story Fragment	Activity Theme	Category
"For me it is really about having all the documents to include archives in a central repository that can be easily searched and information pulled to address inquiries." (2)	• No centralized storage of training info – repeated digging to answer questions • Need secure, centralized storage of LPs • Need ability to search content across all LPs	ACCESS REPORTING
"Right now the "what" part of the training equation only goes as deep as the course title, description, sponsor, hours, topic and a few more minor data elements. We need to expand our knowledge of "what" was taught to include the detailed lesson plans, course objectives, and job tasks… If this was in a searchable database, reporting functions could easily be developed." (4)	• Need ability to search content across all LPs • Need learning objectives tracked to LPs • Need mapping and updating of job task list	ACCESS REPORTING DEVELOPMENT
"I am usually asked about Lesson Plans and Curriculum Maps. Who created the material, what SMEs were involved with creating the material or consulted." (7)	• Need mechanism for collaboration and review	ACCOUNTABILITY DEVELOPMENT FLETA REQUIREMENTS
"There needs to be links to the audience being taught, the number of times it is taught, the effective date of the material, why were changes made and a record of approvals." (2)	• Need additional metadata for training programs • Unclear ownership of training • Justification for changing LP, with effective date • Need clear approval process for LPs	REPORTING DEVELOPMENT VERSIONS ACCOUNTABILITY
"Specific challenges: Finding a job task list that is actually accurate and useful. Re-tooling lesson plans to match the latest LP format. Finding time to thoroughly go through curriculum and update it… one of my big challenges has been keeping instruction in line with the constantly shifting world of FBI policy." (3)	• Need mapping and updating of job task list • Need facilitated creation of LPs, including templates and reusable LPs • No method to map to policies and changes in policies • Need LPs linked to policies, policy subsections	DEVELOPMENT FLETA REQUIREMENTS
"I think just linking a course to a policy is 80 to 90% of the goal… it would be helpful to be able to map to the most granular requirement, whether that is at the course level or learning objective level, whether it is a policy, regulation, or statute (law)." (4)	• No method to map to policies and changes in policies • Need LPs linked to policies, policy subsections • Need learning objectives tracked to LPs	DEVELOPMENT FLETA REQUIREMENTS REPORTING
"Something else that would be useful is some kind of a lesson plan template tool with drop-down menus and fields for entering information that would then generate a perfectly formatted lesson plan. Lesson plans could be linked to PowerPoint presentations, worksheets, instructor materials, resources, etc." (6)	• Need facilitated creation of LPs, including templates and reusable LPs • Need LPs linked to instructional materials, evaluations, and other LPs	DEVELOPMENT
"…we seem to be arbitrarily creating forms just for FLETA compliance even though the information is already contained elsewhere… If there were a system that could allow the data to be automatically entered or drawn from supporting (existing) documents, it seems that would dramatically reduce the frustrations… I don't care which place it is housed, but asking us to do the same thing twice is redundant and unnecessary." (1)	• FLETA compliance data contained in multiple, static documents • Need FLETA compliance data pulled from existing documents	FLETA REQUIREMENTS
"[I am asked] Do you retain your curriculum for a period of time? Yes, as required by federal law… I think we have several approved lesson plans out floating around. Who knows who has the latest and greatest…" (5)	• Need archival of LPs for historical analysis	VERSIONS

Table 1: Sample data show how Story Fragments map to Activity Themes and Categories

From the time the participants were recruited, they were told that the primary purpose for the study was to address the problem of lesson plans (LPs) not being electronically associated with other training information, such as job tasks (JTs) and organizational policies (OPs). Although most of their responses did support this stated purpose, the participants also brought up several other issues – such as the *ease of development* of LPs, the *accountability* for LP collaboration and workflow, and the *versioning* of LPs – which superficially seem to be outside of the scope of ensuring LP linkages to JTs and OPs. Upon closer analysis, however, these additional issues are inextricably tied to JTs and OPs. For example, an LP developer benefits greatly from having quick access to lists of JTs and OPs that help to organize and justify the LP content. A workflow and accountability system ensures that training managers can record their official approval of the way JTs and OPs are associated with the LPs. Finally, LP versioning is required for historical lookups on which LP was taught on a certain date, and which JTs and OPs were in effect at the time. All relevant user needs were captured as part of the analysis and included in the SSM models that represent the results of the study.

Results

The study participants were selected from three different sections at the FBI Training Division and their job descriptions varied from special agent to professional support, from instructor to unit chief, and from instructional systems specialist to information technology specialist. Nevertheless, they all seemed to independently converge on a common set of themes that represent both the current state of affairs as well as envisioned solutions. Every participant, for example, recognized the need to link LPs to JTs and OPs. Some expressed the need as supporting the development of LPs, while others justified it as a way to comply with FLETA requirements.

One of the surprising phenomena observed during the study was the level of emotion expressed by the participants during their responses. One could sense their frustration as they recounted the time and effort wasted in dealing with rudimentary content management processes that result from unsearchable static documents and training data that regularly is unavailable for reporting – the type of reporting required to respond to FLETA-related inquiries. Consider the following comment from one of the participants:

> The first challenge that comes to mind is FLETA, and everything associated with it… The amount of time these tasks have taken… gives the appearance that we have lost our focus on why the [FBI] Academy exists and what its function truly is in the organization.

It is likely that the researcher's efforts to safeguard the identity of the participants enabled such candid and detailed responses. Indeed, by allowing the participants to tell their stories without fear of reprisal likely has resulted in a more accurate and robust set of user needs than what could have been obtained otherwise.

Analysis of the story fragments revealed that some activity themes corresponded to one particular category, while others could be associated with multiple categories (see Table 1). This many-to-many relationship between themes and categories lent itself to depicting the model as a webbed system rather than a hierarchical one. Therefore, two SSM diagrams were developed to illustrate these complex relationships–one depicting the current problems with training management and the other illustrating the goals of the proposed KM system. These diagrams were e-mailed to the participants for their review and feedback. After their feedback was incorporated, the final SSM models were developed. Figure 1 shows the current problems encountered with training management and Figure 2 shows the goals of the proposed KM system.

Figure 1 represents the relationships between activity themes, identified from participant story fragments, and the categories that represent *problematic conditions* to which they contribute. Solid arrows represent primary relationships, and dotted arrows represent secondary relationships based on the analysis of the data.

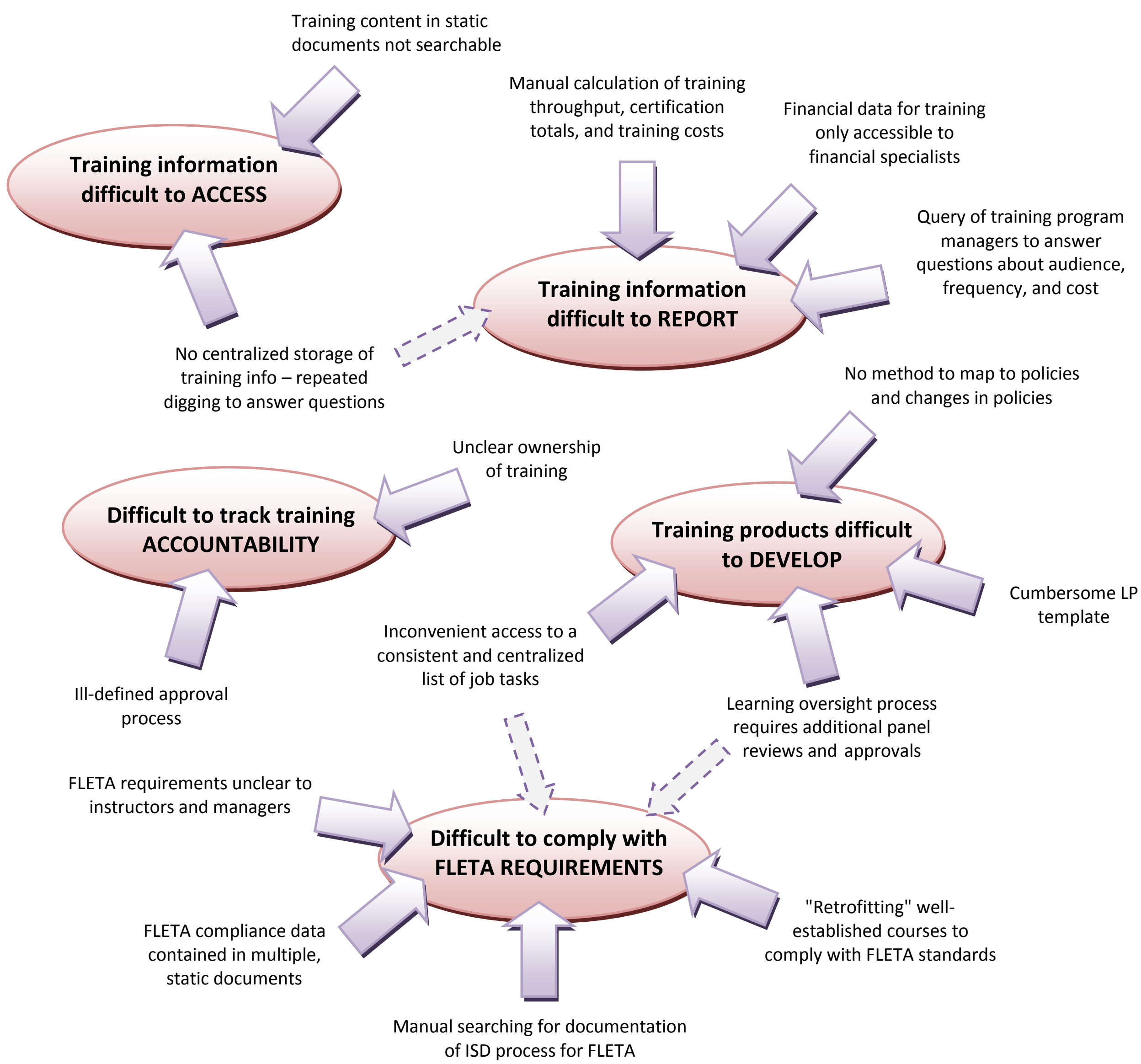


Figure 1: The current problems encountered with training management.

The SSM diagram in Figure 2 represents the relationships between the activity themes and the categories of *positive conditions* to which they contribute. Liaskos, et al. (2011) refer to these concepts as *goals* and *preferences*, which form the basis for defining software requirements.

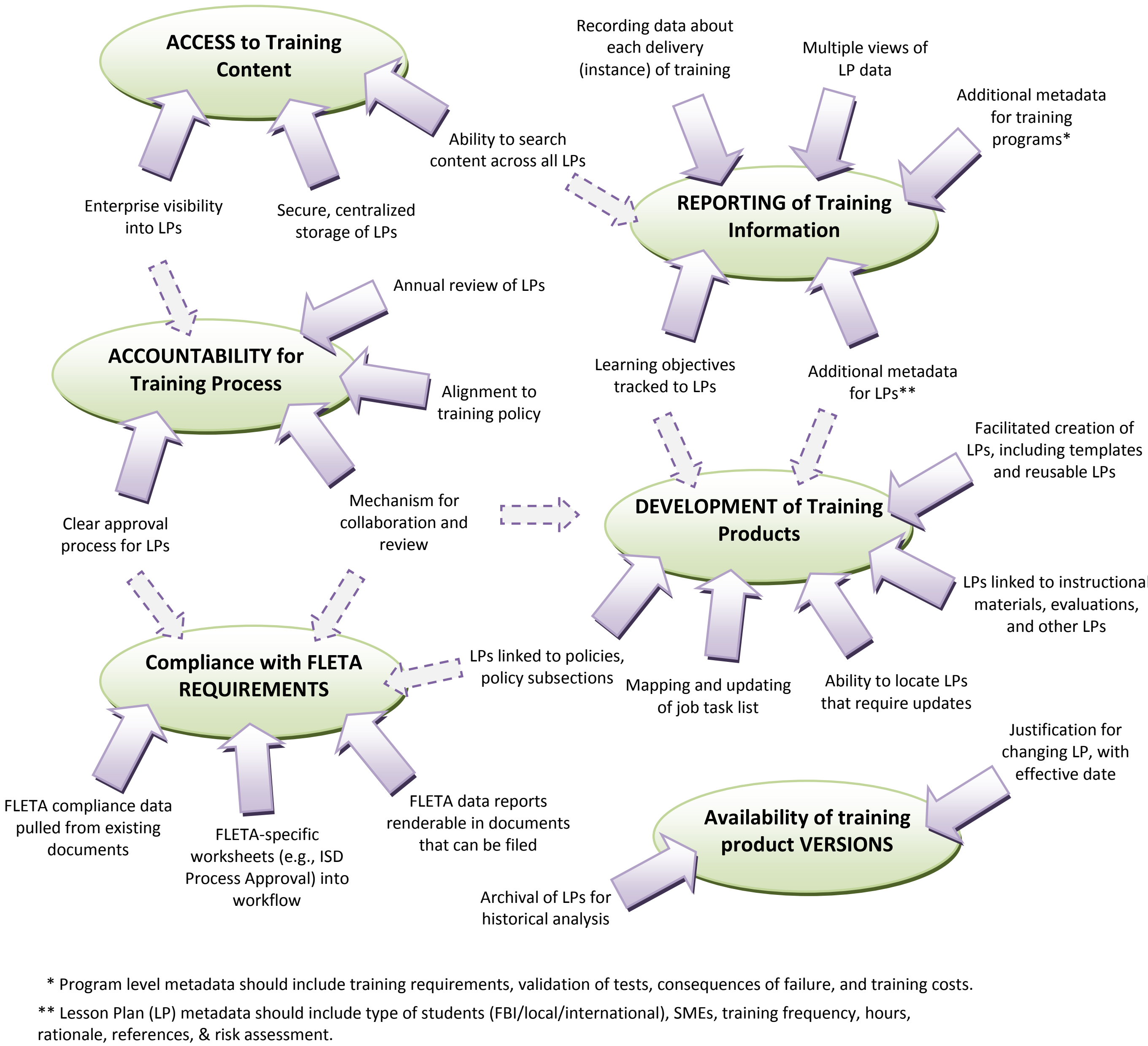


Figure 2: The goals of the proposed knowledge management (KM) system.

Summary and Future Considerations

The goal of this study was to discover the KM capabilities required to associate LPs to JTs and OPs and to develop a strategy to capture the user needs for such a KM system. User needs were elicited and documented using a combination of traditional and group RE techniques (Coughlan & Macredie, 2000, Laporti, et al., 2009, Scheinholtz & Wilmont, 2011), none of which included time-consuming face-to-face meetings. Indeed, it usually is difficult to find a representative group of stakeholders due to constraints on time, status level, and required expertise (Coughlan & Macredie, 2002). However, managing communications by e-mail enabled the participation of several knowledgeable and influential people, who are often the busiest in the Training Division. In addition, the safeguarding of identities resulted in rich responses that were candid and extensive.

Extracting key statements from participants' stories enabled the expressed needs to be coded and categorized, resulting in meaningful goals and preferences for the KM system. In this way, the strength of support for particular KM goals (and consequently, KM preferences) may be traced directly back to the raw participant input that provides context for the user needs. Finally, transforming the activity themes and categories into the goals and preferences displayed in SSM diagrams promoted a common understanding of user needs among the participants.

The participants represented a variety of FBI Training Division stakeholders. It is easy to understand their eagerness to help define the KM system that will improve the management of training content across the enterprise. For senior managers, the KM system tracks to the Training Division's strategic objectives and will enable them to respond quickly to data calls regarding training content and management. For instructors, the ability to map content to policies and processes will facilitate LP creation and modification by directly linking to JTs and OPs, which change frequently. For curriculum managers, having ready access to mapped LP content means they will be able to determine whether current training content still is relevant and whether JTs and OPs are covered adequately by training programs. Virtually every Training Division job function may be improved by the implementation of a KM system described herein.

Drawing a contrast with their own theoretical treatment of requirements elicitation (RE) methodology comparisons, Coughlan and Macredie (2002) suggested, "More research therefore is required on methods in use that conduct studies in real-life settings employing more naturalistic techniques so as to reveal the facets of communication in action and context" (p. 71). The findings presented by this paper answer the call for research in real-life settings. In fact, while most studies examine RE techniques after the fact, the FBI study examined techniques to develop user needs and requirements for a KM system that has not yet been formally proposed. Moreover, the researcher acting in the role of developer for a project that does not yet exist is an approach not found in the literature. Eliciting requirements in this way eliminates the schedule, budget, and legal constraints that typically apply to KM projects.

By following the RE methods outlined in this study, a strong set of user needs emerged from the collective input of high caliber FBI Training Division employees. However, the same RE methodology is applicable to the large KM development projects of other CALOs who struggle with building RE consensus among its stakeholders.

Taken together, the captured user needs have described effectively the problem scope of the proposed KM system by defining the criteria for actual software requirements. Future efforts should concentrate on the validation and prioritization of these software requirements in preparation for a formal statement of work (SOW) or request for proposal (RFP).

Chapter Four

Validation Study

Abstract

Large organizations have trouble keeping their ever-changing business processes and policies aligned with their training programs, which are largely based on learning content locked inside static lesson plans. One particular large organization recognized the need for a knowledge management (KM) solution and took steps to elicit user needs from a group of training experts. A subsequent study was conducted to enable the validation, fine-tuning, and prioritization of the natural language software requirements, accomplished by reaching consensus among eight of the most important training stakeholders in the organization–all without having to conduct any face-to-face meetings. The successful validation and prioritization of the 33 user-based requirements were achieved using techniques derived from Hybrid Delphi, preference-based goal modeling, and hierarchical cumulative voting.

Keywords

Requirements validation, requirement prioritization, user needs, knowledge management, corporate learning, systems development, instructional systems design, Delphi technique

Introduction

Large organizations that manage numerous business processes have unique training needs. Many of these organizations have begun implementing the principles of business process management (BPM), which seeks to make an organization more efficient by modifying workflow and eliminating wasteful tasks (Trkman, 2009). However, training is an often overlooked need when decisions are made to change business processes or policies. If these changes affect the entire organization, then many thousands of employees may need to be trained on the new policies and tasks (Reijers, Mans, & van der Toorn, 2009). Unfortunately, organizational policies and business processes rarely are mapped to the existing training products that support them (Hawryszkiewycz, 2005). This oversight makes it difficult to locate and modify training content to reflect organizational changes.

The current validation study was designed to follow up on a previous effort to elicit user needs from a panel of experts in the Training Division of the Federal Bureau of Investigation (FBI). By employing requirements elicitation methods derived from Laporti, Borges, and Braganholo (2009), the experts were asked to submit detailed “stories” expressing their needs for a KM system that manages training curricula. Their collective feedback was deconstructed into story fragments, transformed into activity themes, and categorized into generally desired outcomes. The result was a detailed Soft Systems Methodology (SSM) model (Coughlan & Macredie, 2000) that visualized the interrelated goals representing the user needs for the KM system.

However, generating effective software specifications is not the goal—the real goal is to create effective software (Japenga, 2011). To solve a realistic problem, software requirements need to be validated and prioritized, distinguishing those that are mandatory for the function of the system from those that may be preferred. These judgments may be based on schedule, cost, and system limitations. Therefore, the current study was designed to validate and prioritize the requirements developed from the previous elicitation study. For simplicity, these will be referred to as the previous study and the current study. The remainder of this paper is organized as follows: A set of definitions is provided to clarify the scope and context of the research. Next, the purpose of the study and research questions are introduced, followed by a review of the literature and a discussion of methods used. Finally, a summary of the research findings is presented along with a summary and considerations for future work. Appendix A is provided to assist the reader in recalling pertinent acronyms and definitions.

Definitions

A business process (BP) is a segment of defined activity in an organization. Having a specific beginning and end, a BP is a set of tasks that logically are related to fulfill an organization's objective (Trkman, 2009). Through the use of symbols and arrows, BP diagrams define how work is done and who does it. Due to its size and diversity, the workforce of a large organization participates in hundreds of processes every day (Reijers, Mans, & van der Toorn, 2009). In the FBI Training Division, each person's contribution to a business process is referred to as a job task (JT), which is the construct used to map to instructional materials used to train that person. To avoid confusing participants with an unfamiliar term, JT was used in place of BP throughout the study.

An organizational policy (OP) refers to an organization's documented guidance to the workforce. OPs articulate outcomes and strategic goals that are, in practice, fulfilled by BPs (Trkman, 2009). OPs define the parameters and context for JTs, and may be produced internally or adopted from a higher authority such as a parent organization. Statutes, regulatory guidelines, employee handbooks, and standard operating procedures are examples of OPs. An OP may take the form of an electronic or physical document and convey the specific rules, protocols, or directives approved by executive managers. The FBI manages several hundred OPs through its Policy and Guidance Library (FBI, 2011).

Finally, a learning object (LO) is a construct used to chunk educational material into smaller units for content management in online learning environments (Beck, 2005). An LO may be defined broadly as a container of learning materials that manages instructional content and resources maintained by an organization. However, like other organizations, the FBI typically manages this type of information in the form of a lesson plan (LP). And while an LO is probably a more accurate term to describe the functionality envisioned in the KM system, in the current study the more familiar term LP was used to refer to the anticipated collection point for JTs, OPs, and instructional products—PowerPoint slides, job aids, and demonstrations—needed to communicate the JTs and OPs to the learner.

Purpose

Large organizations have many business processes—sometimes thousands of them—that need to be communicated effectively to their employees (Reijers, Mans, & van der Toorn, 2009). For organizations undergoing transition, modifying the training products that correspond to business processes becomes an endless exercise in trying to catch up to the latest policy or technical implementation that affects those processes. Indeed, depending on the organization's blended training approach, every change in a business process could require a corresponding change in a policy manual, job aide, user guide, online help section, PowerPoint presentation, computer-based training module, instructor-led training unit, or train-the-trainer component. Hundreds of business process changes easily can breed thousands of adjustments to training products, and the chore of making these adjustments often is attempted without a means to find exactly where in the curriculum the changes need to be made. For large organizations the coupling of policies, processes, and training content is no trivial task.

Due to their sizable workforces and number of business processes, large organizations seek efficient mechanisms to capture and disseminate knowledge to the organization. An effective training management solution is needed to facilitate how an organization's policies and business processes are communicated to its employees. The current study sought to create a set of natural language software requirements for a knowledge management (KM) system—one that will coordinate an organization's learning systems with its BPs and OPs. Specifically, the goal was to produce a high-quality, validated set of prioritized human language requirements that will inform the development of the envisioned KM system. To attain this goal, this study was guided by the following research question:

- What is the relative importance of the requirements necessary to inform the development of a KM system that coordinates learning programs with policies and business processes?

This question helped to focus the review of the literature and the selection of methods applied to the requirements elicited from the previous study. The following section summarizes current research pertaining to techniques for both requirements validation and achieving group consensus.

Background

After software requirements are elicited from stakeholders, the next logical step is to validate those requirements to ensure that they meet user needs (Nuseibeh & Easterbrook, 2000). However, for large-scale KM systems not all of the requirements may be implemented into the final product. Therefore, the requirements also must be prioritized to maximize the fulfillment of technical constraints, business aspects, and crucial stakeholder preferences (Perini, Ricca, & Susi, 2009).

The process of validating a set of requirements is a difficult task for two reasons. The first reason was suggested by Nuseibeh and Easterbrook (2000), who explained that getting to the truth regarding a good set of software requirements is similar to developing a good scientific theory. The requirement descriptions not only must be testable empirically, but they should also be refutable. In fact, inaccurate requirements should be identified by trying to falsify them during testing. Consider a sample requirement: “A lesson plan (LP) will contain links to the organizational policies (OPs) that provide the authority for the content being taught in the LP.” During validation the participants may be asked, “Are there any instances where an LP should

NOT require a link to an OP?" If the answer is yes, then the requirement may need to be modified to say "A lesson plan (LP) will include *the ability to link* to the organizational policies (OPs) that provide the authority for the content being taught in the LP." In the modified version, the user is not required to define links to the OP(s) when creating the LP.

A second area of difficulty lies in the attempt to maintain agreement among all stakeholders, especially when they have divergent goals. The goals of executive managers may be very different from those of mid-level managers and end users in terms of costs, accountability, security, and performance of the new system. Political and social pressures are constantly at work, shaping the outcome of requirements negotiation. Nuseibeh and Easterbrook (2000) reviewed several models of negotiating requirements, but realized that since all parties will not be pleased with all requirements, the overall objective should be to accommodate the most important goals identified by each stakeholder.

In the field of requirement engineering there is a difference between system requirements and software requirements (Japenga, 2011). System requirements describe the required hardware, networking components, and other aspects of the system. Software requirements focus exclusively on the features and functionalities associated with the software, where the end users concentrate, and where most of the complexity is encountered. For the purposes of this paper, "requirements" and "user needs" will be used interchangeably to refer to the natural language descriptions of the envisioned KM software functions and not to the more technical statements used to bind software developers to activities specified in contracts. Regardless of the technical granularity of software requirements, there is a considerable amount of literature-based guidance on how to best evaluate them.

The Institute of Electrical and Electronics Engineers (IEEE) publishes industry standards in many areas of hardware and software development (see www.ieee.org). Their most recent set of recommendations regarding software requirements specification (SRS) is contained in the document known as *IEEE STD 830-1998* (IEEE, 1998). In this standard the IEEE outlines the benefits for developing software requirements—including the establishment of common agreements between customer and developer, the reduction of development effort, the facilitation of estimating costs and schedules—and providing the basis for validation and compliance testing. They also recommended that several issues be addressed in SRS development, such as functionality, attributes, and design constraints, and they offered specific characteristics that should be included in each requirement. In essence, requirements should be:

- *Correct* – Accurately representing what the customers desire in the system.
- *Unambiguous* – Open to only one interpretation.
- *Complete* – All that is needed for the developer to build a particular feature.
- *Consistent* – Using the same terminology across descriptions.
- *Ranked for importance and/or stability* – Separating priority goals from nice-to-haves.
- *Verifiable* – Including measurable standards.
- *Modifiable* – With the ability to change the same terms and definitions across multiple requirements.
- *Traceable* – The ability to track requirements to overarching guidelines, policies, and artifacts of requirements elicitation, such as interviews and surveys.

Despite its detailed guidance regarding the characteristics of requirements, the IEEE standard offers very little in terms of *how* the requirements should be developed or validated. It simply recommends that customers (users) and suppliers (developers) should work together on defining requirements, so that “the functionality, interfaces, performance, and other attributes and constraints of the software are not predefined, but rather are defined jointly and subject to negotiation and change" (p. 14). Fortunately, there are many studies that have suggested best practices for accomplishing effective requirements validation. In general, they have focused on two areas: quality assurance and prioritization.

Denger and Olsson (2005) observed that many of the quality assurance (QA) activities of a software development project typically are performed at the testing and maintenance stages. However, without conducting QA on the software requirements, developers could be building a system based on the wrong requirements. Denger and Olsson (2005) stressed that all subsequent steps of the software development process are influenced by the requirements that are defined for the system. Therefore, the requirements need to be subjected to QA from various perspectives, including:

- *User view* – The requirements need to reflect what the user requires of the final system.
- *Product view* – The system must be described in such a way that it can be developed efficiently.
- *Manufacturing view* – The requirements need to adhere to certain standards that can be measured.
- *Value-based view* – The requirements must provide a basis for relating value to cost.

According to Denger and Olsson (2005), the quality of requirements themselves is only part of an overall QA strategy for requirements engineering. One also must consider available resources, risks, time schedule, and organizational aspects as they relate to the software project. The current study offers many advantages in these areas in that the project is relatively low-risk (i.e., human lives are not dependent on its outcome), it has no defined timetable, and the participants are willing volunteers who seek to improve work life for themselves and for the organization.

During the previous elicitation study, conceptual models were created using the Soft Systems Methodology developed by Checkland (1998). These models illustrated how the elicited software requirements relate to each other. Liaskos, McIlraith, Sohrabi, and Mylopoulos (2011) expanded the idea of requirements concept mapping to include user preferences. Their framework distinguished between mandatory and non-functional requirements for a system. The latter describes "nice-to-have" requirements, the priorities of which seldom are understood or documented as they relate to mandatory requirements.

In the goal model developed by Liaskos, et al. (2011), software tasks are specific, measurable activities that lead to goals. Borrowing from the example presented by Liaskos, et al., sample tasks could be Obtain Credit Card Number, Obtain Credit Card Authorization, and Charge Credit Card. Goals, on the other hand, are states or conditions that are desired, such as Payment Done Via Credit Card. Their goal model depicts arrows indicating precedence (sequencing) between the tasks and arrows between each of the tasks and the goal indicating an AND-decomposition, meaning that all of the tasks must be completed in order to achieve the goal. From the goal, other arrows are drawn to preferences, such as Expedited Process, Payment Flexibility, and Customer Convenience. One interesting area explored in the current study was the effect these preference links had on the prioritization of requirements.

After the requirements are further refined and clarified, Denger and Olsson (2005) suggested that a QA analysis be applied to the requirements list. The first approach they suggested involves an inspection process. An inspection checklist can be useful for uncovering defects, but the checklist must be project-specific and accompanied by significant guidance. Scenario-based inspections are more effective, according to Denger and Olsson (2005), because they provide the context required to properly evaluate requirements. By inspecting requirements through the lens of realistic scenarios, the participants also may verify the traceability of the requirements back to the problems they are meant to solve.

The second QA approach suggested by Denger and Olsson (2005) is the application of requirements-based testing. They insisted that test planning and the creation of test cases should be completed as soon as the requirements are defined. The process for defining test cases identifies many types of defects, because if test engineers cannot derive acceptance test cases from the requirements, then they will need to be modified. Traceability is also important in test creation, as tests need to be tracked to requirements so that regression testing may be performed if requirements are modified.

Assigning priorities to requirements is essential given that cost, schedule, or system constraints may preclude some requirements from being implemented in the final software solution. Karlsson, Wohlin, and Regnell (1998) evaluated six different methods for prioritizing software requirements. They concluded that the analytic hierarchy process (AHP) was the most effective method, although it had problems with scaling to larger sets of requirements. AHP involves making pair-wise comparisons between each of the requirements. In addition, an importance factor is assigned to each of the comparisons, so that a value of 1 would indicate equal importance while a value of 9 would indicate an extreme difference in importance. For a software project having n requirements, a person would need to make n(n-1)/2 pair-wise comparisons. Therefore, in the study conducted by Karlsson, et al. (1998), a total of 78 comparisons were required to prioritize the 13 requirements they examined using the AHP method. It is easy to see how prioritizing a larger requirement set in this way could consume time and resources, although computer-based tools can make the comparison process a little easier (Perini, et al., 2009).

Hierarchy AHP is a variation of AHP that was also reviewed by Karlsson, et al. (1998), although it did not perform as well as AHP for their requirement set. With hierarchy AHP a larger set of requirements is sorted into general categories, which are placed into a hierarchy.

Because only requirements at the same level in the hierarchy are compared to each other, this method greatly reduces the number of pair-wise comparisons that are required.

Dealing with this type of multi-level requirements prioritization, Berander and Svahnberg (2009) examined two methods of calculating priorities using the hierarchy techniques. In their study, lower-level requirements (LLRs) were organized under more general high-level requirements (HLRs). Rather than using the pair-wise comparisons of AHP to achieve a ranking of LLRs within blocks of HLRs, they used a cumulative voting technique to distribute 100 tokens among LLRs. This technique is called hierarchical cumulative voting (HCV). The researchers then conducted an empirical study comparing two different methods of employing HCV.

One prioritization method involved multiplying the relative priority of each LLR (P_{LLR}) by the priority of its parent HLR (P_{HLR}) in order to arrive at an absolute priority relative to all other LLRs. However, the researchers noted that this method favors HLRs with fewer requirements. The bias was caused by the fact that the priority tokens were distributed among fewer requirements. To remedy this problem, a compensation factor was added to the equation. The researchers chose to multiply the factor P_{HLR} x P_{LLR} by a compensation factor (C_{HLR}) equal to the number of LLRs comprising an HLR block. In other words, an HLR block containing two LLRs would merit a compensation factor of 2, and an HLR block containing ten LLRs would merit a compensation factor of 10, thus compensating for the bias favoring low-LLR blocks. In this way, the absolute priority of an LLR was calculated by multiplying P_{HLR} x P_{LLR} x C_{HLR}. Berander and Svahnberg (2009) reported that participants preferred the compensation factor for hierarchical prioritization of requirements.

In order to employ a group of experts to validate and prioritize requirements, some type of method is needed to harness the collective knowledge of that group. Bolger and Wright

(2011) refer to this activity as a process of "aggregating judgment" (p. 1500). In some studies, group opinions on requirements are captured and tallied in real time by having experts log into computers simultaneously (e.g., Berander & Svahnberg, 2009 and Perini, et al., 2009). In the current study, however, it was unlikely that all of the desired group of experts would be available to participate at the same time. Therefore, other consensus-building techniques were sought in the literature.

Graefe and Armstrong (2011) conducted an experiment to compare four different methods: Face-to-face meetings, Nominal groups, Delphi, and Prediction markets. Although the researchers did not find statistically significant differences in accuracy among the four techniques, they did observe qualitative differences in implementation and participant ratings of the methods. Face-to-face meetings involve relatively unstructured interactions, while Nominal groups employ a mix of individual contributions and group discussions. Both of these techniques are vulnerable to live-meeting, participant-related biases, such as dominant personalities and "groupthink." On the other hand, prediction markets involve the anonymous participation of experts who have a contrived stake in the outcome of the event they are predicting. However, in the Graefe and Armstrong (2011) study, the prediction markets technique was given the least favorable rating due to its complexity.

Delphi is a survey technique by which participants anonymously contribute their opinions or estimates, which are aggregated and provided back to the group in multiple rounds. In the study by Graefe and Armstrong (2011), the Delphi technique performed the best out of the four methods on specific experimental questions and provided the greatest improvement in group accuracy of estimates compared to participants' prior individual estimates. Furthermore, participants were most confident in the results produced by Delphi, which also scored high in other rating categories such as *freedom to participate, time well-spent*, and *satisfaction*.

Bolger and Wright (2011) also highlighted the benefits of Delphi, claiming that its multiple rounds of anonymous surveys and feedback enable participating experts to “converge on the truth” (p.1500). Drawing upon a social science analysis, these researchers established that Delphi overcame several of the nonconstructive pressures to change opinions that are found in traditional face-to-face collaborations. Instead, Delphi enabled participants to state positions and provide justification for those positions without fear of embarrassment or being dominated by overconfident participants.

Finally, a unique consensus building method was proposed by Landeta, Barrutia, and Lertxundi (2011), who combined aspects of the Face-to-face meeting, Nominal group technique (NGT), and Delphi to create a Hybrid Delphi. This technique involved two stages. The first was a face-to-face stage, where a facilitator elicited stories and reflections from an expert focus group, which led to questions that were posed to the group using modified NGT activities. Activities in this stage produced outcomes similar to those of the elicitation methodology used in the previous FBI study. (However, the previous study did not require face-to-face meetings.)

The second stage involved two rounds of Delphi surveys to the same group, during which an ordered list of proposals was produced. Landeta, et al. (2011) noted how participants were able to follow the established criterion, respond in a reasonable amount of time, and “think out their answers” (p. 1633) using an asynchronous survey method. This Hybrid Delphi method was tested successfully on three real cases during their investigation.

Similar to what was experienced with the requirements elicitation methods in the previous study, the validation of requirements in the current study utilized a variety of techniques that were tailored to the project (Perini, et al., 2009). The next section will describe the methods, which combine many of the techniques that were uncovered in the above literature review.

Methods

The methods were implemented in three phases. First, the user needs elicited in the previous FBI study were organized into a logical goal model (Liaskos, et al., 2011). Next, stakeholder participants were identified from those who represent a cross section of potential users of the proposed system. Finally, data collection was conducted using techniques derived from Hybrid Delphi (Landeta, et al., 2011), preference-based goal modeling (Liaskos, et al., 2011), and hierarchical cumulative voting (Berander & Svahnberg, 2009).

Organizing user needs

Starting with the SSM model developed in the previous elicitation study, a diagram similar to the goal model presented by Liaskos, et al. (2011) was developed to capture the tasks, goals, and preferences of the proposed KM system. In essence, lower level tasks were organized under higher level goals, which led to preference outcomes for the proposed KM system. Unlike their model, however, the "++" and "--" notations were abandoned in favor of a simpler, more hierarchical structure to represent dependencies. Figure 1 depicts a small segment of the actual goal model, and in this paper it will be used to simplify the explanation of techniques used to validate and prioritize the software goals and tasks. The actual tasks, goals, and preferences were derived from the activity themes and categories developed from the user-based stories of the previous study and will be discussed in the results section.

In the model, it is possible that more than one goal can lead to a preference, and conversely, more than one preference could result from a single goal. Based on the findings of Liaskos, et al., the model was kept to fewer than 50 elements to avoid confusion. The model not only depicts the webbed relationships between goals and preferences but it also exhibits a goal-based hierarchical structure, which is not present in the goal model of Liaskos, et al. (2011). The model served two primary purposes: To structure smaller sets of information that were provided to the participants for their review and to guide the final priority calculations for each user need.

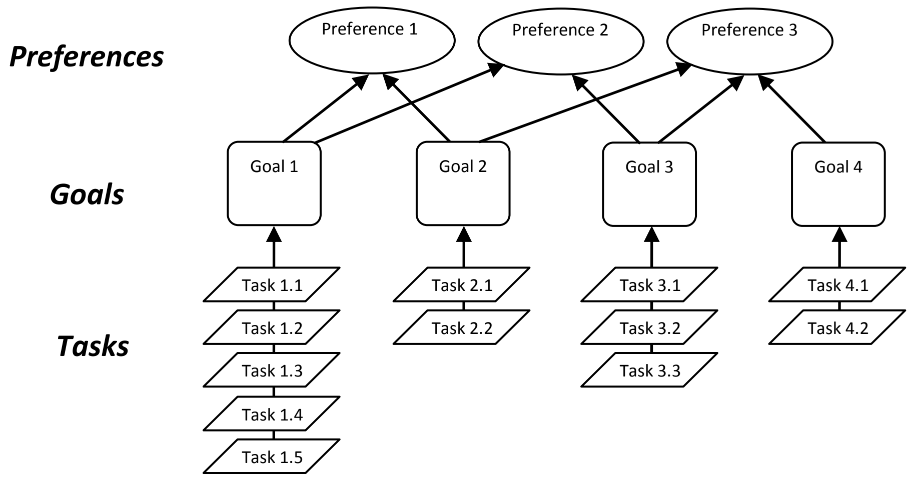


Figure 1: A segment of a sample goal model.

Selection of participants

Several academic and professional colleagues were recruited for participation. Landeta, et al. (2011) tested their Hybrid Delphi technique in three real cases, all of which involved six to ten expert participants. Therefore, eight participants were recruited from the following positions: Instructional Systems Specialist, Front Line Instructor, Instructor Supervisor, Training Program Manager, Training Executive Manager, and Software Systems Designer. All experts provided their written consent to participate in the study and were told that the data they submit would be reported only in the aggregate and not associated with their names or positions.

Each participant was sent an introduction letter, which highlighted the benefits of participation. The letter was designed to appeal to participants who may be motivated by the opportunity to help define a KM system that will make their jobs easier. Volunteer participation is the most desired type of participation because it tends to increase the ability to generalize from the findings (Berander & Svahnberg, 2009).

The participants were experienced practitioners with previous knowledge of the domain (Perini, et al., 2009), as well as a stake in the outcome of the project. In fact, having familiarity with the domain made additional training unnecessary (Berander & Svahnberg, 2009). However, Bolger and Wright (2011) suggested that participants should not only be experts, but should also be diverse in their perspectives, avoiding too much similarity of opinion. The eight FBI employees who agreed to participate included a seasoned frontline instructor, who regularly designs and delivers training; one supervisory instructional systems specialist, an expert in instructional systems design (ISD); three unit chiefs who supervise instructors in three different training programs (and who deliver training themselves); one unit chief who regularly answers division-level data calls regarding FBI training and policy; a unit chief with significant technical expertise in learning management systems; and a supervisory information technology specialist with a Ph.D. in computer science. The variety of experts enable what Denger and Olsson (2005) called perspective-based reading (PBR), whereby requirements are inspected by stakeholders representing a variety of positions and interests.

Data collection

Based on a review of current research, the Delphi technique was deemed to be an effective way to obtain a reliable group opinion from experts. Although a specific example of using Delphi to validate and prioritize user needs (or software requirements) could not be found in the literature, the technique has been used for "the obtaining of ranked lists of problems, practices, solutions… regarding a specific phenomenon using the tacit knowledge of professional experts" (Landeta, et al., 2011, p. 1629). This statement accurately reflects the aims of the current study.

Delphi was used to collect data, which included individual rankings and weightings of requirements, along with the rationale that supports each judgment. The rationale is important for the discussion of validation and prioritization because the quality of the rationale should serve as a "good cue to truth" (Bolger & Wright, 2011, p. 1503). Also, guidance found in the literature was adopted in the administration of the Delphi technique. For example, there were no more than two rounds of voting per Delphi phase, each of which lasted no more than two weeks (Landeta, et al., 2011). Also, each round was engineered to require as few judgments as possible (Bolger & Wright, 2011). To keep the number of judgments to a minimum, therefore, data were collected using three separate phases of the Delphi technique.

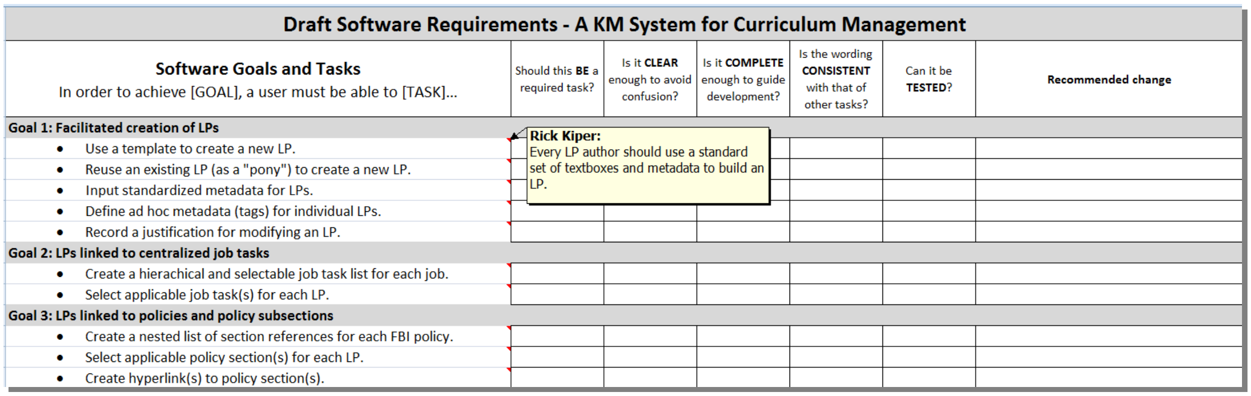
Delphi Phase 1: Quality assurance and verification

Denger and Olsson (2005) suggested that a combination of quality assurance (QA) techniques should be applied to software requirements in a manner that is specific to the project. During this data collection phase, a checklist of questions was provided to prompt participants to inspect the requirement statements for defects relating to the previously mentioned IEEE (1998) quality measures. Because certain quality measures—such as ranking and traceability—were to be controlled by the study itself, the remaining measures were reworded to ask the following questions about each task:

- Should this BE a required task?
- Is it CLEAR enough to avoid confusion?
- Is it COMPLETE enough to guide development?
- Is the wording CONSISTENT with that of other tasks?
- Can it be TESTED?

To create an effective inspection checklist these questions were added as columns to a spreadsheet containing a list of the goals and tasks extracted from the goal model (see Figure 2). To clarify the difference between goals and tasks the following explanation was provided to the Goals and Tasks column header: "In order to achieve [GOAL], a user must be able to

[TASK]…" By substitution, for example, the participant understood that "In order to achieve [Goal 1: Facilitated creation of LPs], a user must be able to [Task 1.1: Use a template to create a new LP]." In addition, a mouseover comment providing a brief explanation of each task description was added to each task in the spreadsheet.



Draft Software Requirements - A KM System for Curriculum Management						
Software Goals and Tasks In order to achieve [GOAL], a user must be able to [TASK]...	Should this **BE** a required task?	Is it **CLEAR** enough to avoid confusion?	Is it **COMPLETE** enough to guide development?	Is the wording **CONSISTENT** with that of other tasks?	Can it be **TESTED**?	**Recommended change**
Goal 1: Facilitated creation of LPs						
• Use a template to create a new LP.						
• Reuse an existing LP (as a "pony") to create a new LP.						
• Input standardized metadata for LPs.						
• Define ad hoc metadata (tags) for individual LPs.						
• Record a justification for modifying an LP.						
Goal 2: LPs linked to centralized job tasks						
• Create a hierachical and selectable job task list for each job.						
• Select applicable job task(s) for each LP.						
Goal 3: LPs linked to policies and policy subsections						
• Create a nested list of section references for each FBI policy.						
• Select applicable policy section(s) for each LP.						
• Create hyperlink(s) to policy section(s).						

Figure 2. Quality Assurance checklist for software requirements.

Utilizing standard enterprise e-mail, participants were provided with the spreadsheet and asked to complete two activities with respect to the list: 1) Inspect the tasks (representing software requirements) to ensure they meet quality standards provided in the checklist, and 2) If the answer is "no" to any quality measure, provide a recommended change to the task. In accordance with the Delphi method, participants were asked to respond by e-mail to the researcher to eliminate any influence from other participants. Their recommended modifications were accommodated, their rationale was summarized, and both were returned to the entire group for verification. The requested modifications, along with the subsequent e-mail discussions between researcher and participants, provided a context-rich set of clarifications that were codified for each task and made available to the group during the next part of the study. The prioritization of the requirements was accomplished using the next two phases of Delphi.

Delphi Phase 2: Prioritization of goals

Referring to their goal model, Liaskos, et al. (2011) suggested that the identified preferences be assigned numerical weights in a way similar to cumulative voting that uses "tokens" (Berander & Svahnberg, 2009). Liaskos, et al. also noted that assigning an absolute value to these weights is not as important as capturing their importance relative to each other. Therefore, in the first round of this phase of Delphi, participants were asked to assign numerical weights (using 100 tokens) to each of the preferences, along with a justification for each value chosen. As Bolger and Wright (2011) suggested, the participants were asked to provide feedback that included specific examples that illustrate reasons for the judgment. To facilitate the weighting and feedback process another spreadsheet was created so that both numerical and text data could be provided with a simple e-mail reply.

After the scores for each preference were received by the expert panel they were averaged and returned to the group along with any justifications that were provided. The range of scores (the highest and lowest score) for each preference was also provided to the group to aid in the evaluation. In the second round of this Delphi phase, each participant was asked to review the scores and justifications and then provide new scores for each preference. The new scores after the "re-voting" were averaged and then plugged into the preferences portion of the goal model. For illustrative purposes, a simple example of scores for preferences is depicted in Figure 3. (In reality, 6 preferences were identified, along with 11 goals and 33 tasks, which will be discussed in the Results section.) In keeping with the "tokens" idea, the points from each preference were divided among the goals that support it. Goals that support multiple preferences received their share of the points from that preference. The resulting scores defined the prioritization for the goals, which is the starting point for the next Delphi phase.

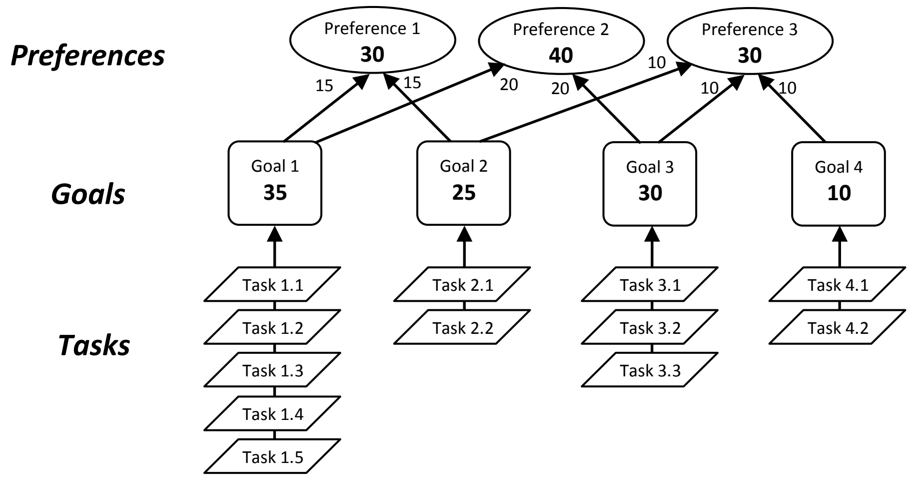


Figure 3. Assigning priorities to goals using sample data.

Delphi Phase 3: Prioritization of requirements

The previous Delphi phase combined two prioritization methods as the preference weighting technique guided by the research of Liaskos, et al. (2011) led to the prioritization of the goals, or higher level requirements (HLRs), called for by Berander and Svahnberg (2009). This was the starting point for prioritizing the subordinate tasks, which will now be designated as lower level requirements (LLRs). The finished translation of terms is depicted in Figure 4. In heeding the advice of Berander and Svahnberg (2009) the final model includes no more than seven LLRs per HLR.

In this data collection phase participants were again requested to assign 100 token-based, weighted ratings to each of the LLRs. However, those ratings were made relative to LLRs residing in the same HLR block. Again, to illustrate the technique, sample scores are depicted in Figure 4 with LLR scores in each block adding up to 100. As in the previous phase of Delphi, there were two rounds of e-mail communications to settle the relative prioritization of LLRs. Following the method preferred by Berander and Svahnberg (2009), the absolute priority for each LLR was calculated by multiplying the relative priority of each LLR (P_{LLR}) by the priority assigned to its HLR (P_{HLR}) by a compensation factor (C_{HLR}) corresponding to the number of LLRs in the block. For example, the final priority for the sample LLR 3.2 would be 10 x 30 x 3 = 900. The next section will describe the actual results of both the voting and the calculations.

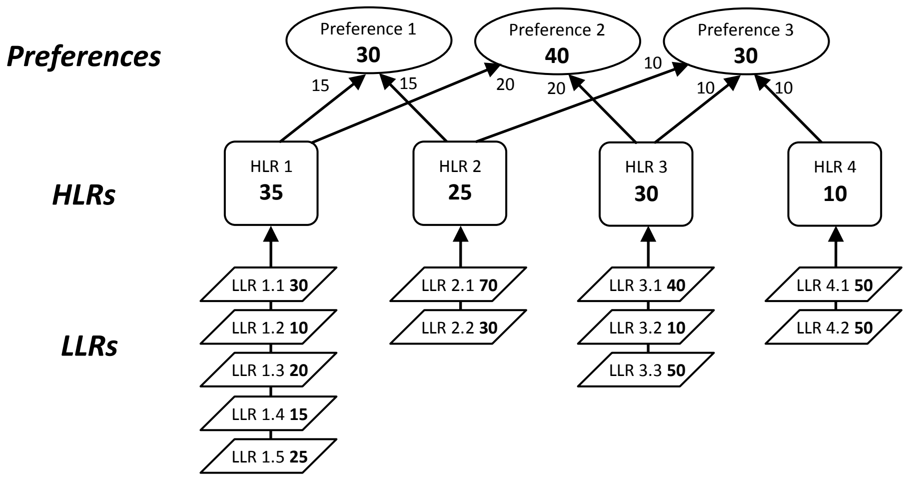


Figure 4. Assigning priorities to lower-level requirements (LLRs) using sample data.

Results

Delphi Phase 1: Quality assurance and verification

During the QA and verification activities of Phase 1, it was anticipated that organizing tasks by their end states (goals) and preferences would reveal conflicts and dilemmas that trigger further discussion and clarification of requirements (Liaskos, et al., 2011). As a result of participant feedback, the wording for several tasks needed to be modified to clarify the meaning of the tasks without becoming overly technical. For example, Task 2.1 originally read, "Create a hierarchical and selectable job task list for each job" and the mouseover comment elaborated by stating, "Representatives from the Curriculum Management Section need the ability to define a list of job tasks (business processes) based on the most recent job task analysis." Three of the participating experts had questions about this task.

One participant took issue with the Curriculum Management Section being named as the entity in charge of creating the job task list. Another did not understand whether the KM system or a user (through an interface) would create the job task list. Another wanted a subject matter expert (SME) to be part of the explanation of who was allowed to create the job task list. As a result the task was changed to read, “Create and manage a selectable job task list for each job” and the task explanation became, “SMEs or specific users authorized by Training Division need the ability to define a list of job tasks (business processes) that will be selectable when authoring LPs.” In addition, the following comments were summarized by the researcher as a result of the e-mail interactions:

> With regard to the job tasks, the KM system needs to provide two capabilities: One to create/modify the list of job tasks and the other to pull from those job tasks while creating the lesson plan. The creating action [Task 2.1] should be completed by an "authorized user" and the selecting action [Task 2.2] should be completed by a "common user," or the one who creates the LP.

Many similar comments were generated as a result of the negotiation of meaning between the participants and the researcher. These comments provided so much rich context that it was decided to add them as a separate column in the goals and task list that was sent out to the group for final review. They also became a useful aid to the participants when they were later asked to prioritize those tasks (i.e., during Delphi Phase 3).

Delphi Phase 2: Prioritization of goals

During this phase, the expert panel was asked to focus on the relative importance of the preferences, or general outcomes, of the envisioned KM system. After a brief explanation of the token-based voting process the participants were asked to evaluate six preferences. To help clarify the meaning of each preference, the following definitions were provided. The definitions were based on the findings of the previous elicitation study.

- **Effective Development of Lesson Plans (LPs)** – The user develops standardized lesson plans in a way that is easy to edit, attach supplementary materials, link to job tasks and policies, and make available for collaborative input.

- **Availability of LP Versions** – Modified versions of LPs are justified, archived, and available for retrieval.

- **Flexible Reporting of Training Information** – Administrative information contained in LPs and training programs is available for a variety of query and reporting needs, both internal and external to Training Division.

- **Consistent Accountability for Training Process** – The process of developing, approving, and delivering training is consistent with policy and transparent to those who have a need-to-know.

- **Convenient Access to Training Content** – The actual content of training products is searchable and easily retrievable for users who have been authorized access.

- **Facilitated Compliance with FLETA Requirements** – The authors of LPs and training program managers comply with FLETA requirements with minimal impact to the training mission.

The participants were asked to provide justifications to accompany the scores they assigned to each preference. The justifications helped to explain the significant differences between the lowest and highest scores of some of the preferences. For example, the third preference, *Flexible Reporting of Training Information*, earned a score of 5 from one participant and 20 from another. The first participant stated, “This is important, but within the six choices, not as important… I know that statistics can be manipulated to show what a person wants to see….” The second participant commented, “Increased oversight—key to future success—don't want to lose our delegated approval authorities from DOJ.” Another preference with a wide scoring range was *Facilitated Compliance with FLETA Requirements*, which earned a low score of 6 and a high score of 30. Justifications such as “Complying to standards may or may not offer valuable ROI” and “We cannot lose accreditation” helped to explain the difference in scoring.

During the second Delphi round the participants were asked to reconsider each of their scores based on the group’s average score, the scoring range, and the justifications compiled from the first round. During this round of “re-voting” many participants changed their scores while a few firmly held their ground, especially on the low and high scores for the fifth and sixth preferences (see Table 1). One particularly volatile preference was *Facilitated Compliance with FLETA Requirements*. One participant, apparently influenced by the positive rationale of others, changed the score from 20 to 24. However, another participant reduced her score from 25 to 20 based on the justifications, offering the following argument:

> I do not agree with the "we cannot lose accreditation" statement, and thus, I still would not rate it any higher than I have in my re-voting. I do agree with the idea that the perceived need to comply with FLETA standards seems to lack a solid ROI in many aspects.

Another low scoring participant offered the following commentary on the same preference:

> This is a double-edged sword. Of course now that we have joined FLETA, we cannot lose that accreditation. The perception and consequences would be devastating within the federal law enforcement community. However, while accreditation does force us in some areas to do the right thing, it also creates a tremendous amount of workload documenting what we already do very well. And in some cases, it forces us to do things differently even if the new value is less than the current value.

These candid discussions about preferences demonstrated the effectiveness of the Delphi technique, as the participants considered each others' input while arriving at a group consensus. Indeed, after the second round of this Delphi phase the scores were moderated (the range was reduced) for three out of the six preferences as depicted in Table 1.

Voting Summary for Preferences						
	Round 1			**Round 2**		
Preference	Low	High	Mean	Low	High	Mean
Effective DEVELOPMENT of Lesson Plans (LPs)	10	25	21.7	10	25	**18.3**
Availability of LP VERSIONS	10	20	13	10	15	**12.2**
Flexible REPORTING of Training Information	5	20	13.8	10	20	**17.7**
Consistent ACCOUNTABILITY for Training Process	5	25	15.5	10	25	**16.3**
Convenient ACCESS to Training Content	10	24	15.7	10	24	**16.3**
Facilitated Compliance with FLETA REQUIREMENTS	6	30	20.3	6	30	**19.2**

Table 1. Results of preference voting.

Overall, the mean scores did not change significantly from the first round to the second. There was some movement of importance away from the *Development*, *Versions*, and *FLETA Requirements* preferences to those supporting *Reporting*, *Accountability,* and *Access*, but the largest difference was an increase of 3.9 points for *Reporting*. Those who did not change their high and low scores on the *FLETA* preference, for example, effectively cancelled each other out in the mean score, but others reduced their rating slightly. Furthermore, the mean scores were distributed more evenly between all six preferences, as indicated by a mere 7 point difference between the lowest mean score, *Versions*, and the highest, *FLETA Requirements*.

Although these Delphi rounds did not identify user preferences with extraordinary scores, an important part of this exercise was that the preferences themselves were discussed and the justifications were documented in a context that made sense to the user. Indeed, these discussions will assist software developers with understanding the user needs. The scores from the preferences were used to calculate the final prioritization of the requirements, which is discussed in the next section.

Delphi Phase 3: Prioritization of requirements

During the last phase, the participants were asked to evaluate the importance of tasks associated with each goal. The voting on tasks in this phase was a natural extension of the voting they had just completed in the previous phase. However, in this phase the participants were afforded 100 tokens for each of the 11 goals. The participants were required to distribute their tokens between as many as five tasks per goal or as few as two. This simplified the voting somewhat and may have contributed to the reduced number of justifications that accompanied the voting. Another factor could have been the presence of detailed comments that were added to the task explanations after the Phase One discussions. One participant noted, "I didn't add many new comments as your new statements helped address any previous concerns." In any case, the voting conducted on tasks did not spawn the type of strongly worded justifications that were submitted during the preference voting.

Nevertheless, during the task voting the participants asked additional clarifying questions, some of which resulted in further adjustments to the task descriptions. For example, one participant submitted this issue:

> Still a bit confused between the stated task and the explanation used to describe the task. For example Task [6.2] here talks about Routing … and the explanation speaks to a way of keeping historical records (or logs) of who the collaborators were. For developers I am not sure they would make the transition from “ability to route an LP” to “provide a record of who collaborated on an LP.” Consider using the term “Log” where you want the system to capture historical events/actions for later use in other ways such as reports or other actions….

This e-mail exchange prompted the rewording of Task 6.2 to be “Route LP to collaborators and record activity log.” Similarly, Task 10.2 was modified to read “Route LP to approvers and record activity log.” In more formal software requirements these two activities likely would appear in separate requirement statements. However, from the perspective of the participants these two ideas go hand in hand. To separate them for voting would have caused unnecessary confusion.

As with the preference voting phase, in the second round of Delphi voting the participants were provided the low score, high score, and mean score for each of the tasks, along with submitted justifications. However, not only was the number of task voting justifications fewer than those in the preference voting, but also few of the mean scores awarded to the tasks changed substantially from the first round of voting to the second (see Table 2). This is somewhat surprising, as some of the task groups were given a wide range of scores during the first round. For example, the initial scoring for Task 2.1 (“Create and manage a selectable job task list for each job”) and Task 2.2 (“Select applicable job task(s) for each LP”) each resulted in a 30 point difference between their high and low scores. The first round mean scores for the two task groups were 42.5 and 57.5, respectively. During the second round the point range remained the same, and the mean scores separated only slightly to 41.9 and 58.1, respectively, indicating that the participants strongly felt that the ability for a user to select from a list of job tasks was more important than being able to create and modify the job task list itself.

Some tasks that saw movement in scoring were those of Goal 7. Tasks 7.1 and 7.2 scored 56.3 and 43.7, respectively, during the first round and separated to 60.1 and 39.9 during the second round. Upon further reflection, participants may have felt that “Input standardized data for training programs” (Task 7.1) was a more essential requirement than “Create multiple views of program data” (Task 7.2), which may be considered a nice-to-have feature in a future build.

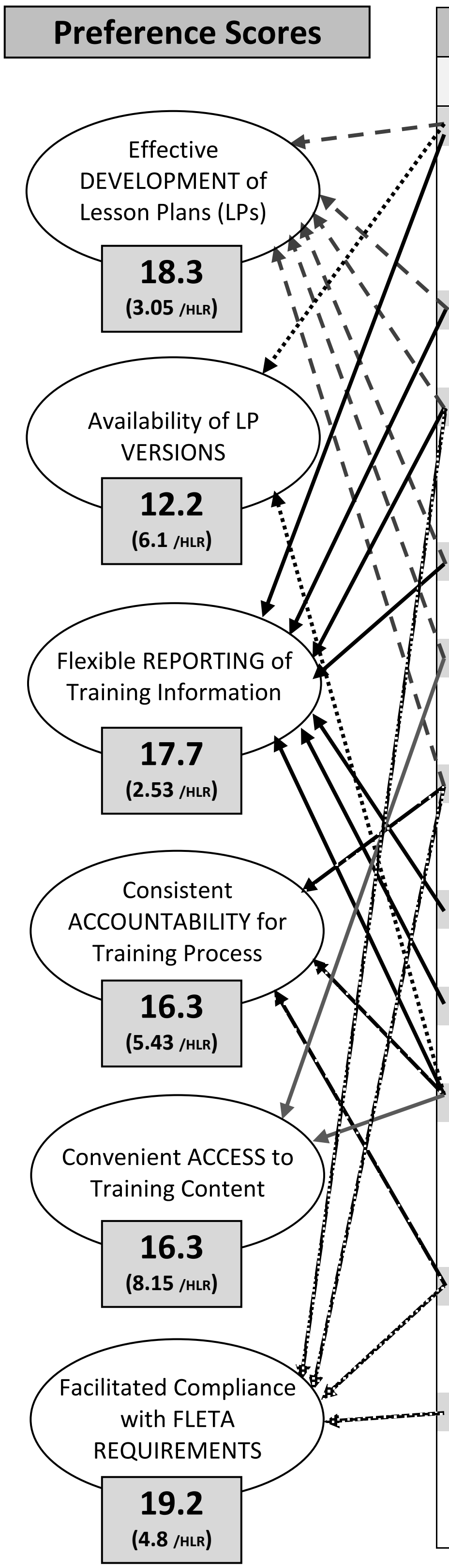
Voting Summary for Task Priorities						
	Round 1			Round 2		
Goals and Tasks	Low	High	Mean	Low	High	Mean
Goal 1: Facilitated creation of LPs						
1.1 Use a template to create a new LP.	20	30	24.3	15	30	**24.8**
1.2 Reuse an existing LP (as a "pony") to create a new LP.	20	22	20.5	20	30	**21.6**
1.3 Input standardized data for LPs.	10	27	18.0	10	25	**17.6**
1.4 Apply ad hoc metadata (tags) for individual LPs.	10	15	11.7	10	20	**15.3**
1.5 Record a justification for modifying an LP.	12	40	25.5	10	30	**20.7**
Goal 2: LPs linked to centralized job tasks						
2.1 Create and manage a selectable job task list for each job.	20	50	42.5	20	50	**41.9**
2.2 Select applicable job task(s) for each LP.	50	80	57.5	50	80	**58.1**
Goal 3: LPs linked to policies and policy subsections						
3.1 Create and manage a selectable and nested list of section references for each FBI policy.	25	60	45.0	30	60	**44.3**
3.2 Select applicable policy section(s) for each LP.	5	50	26.3	5	50	**28.0**
3.3 Create hyperlink(s) to policy section(s).	10	45	28.7	10	40	**27.7**
Goal 4: Learning objectives tracked to LPs						
4.1 Define terminal and enabling learning objectives for each LP.	25	75	50.0	25	75	**51.4**
4.2 Use learning objectives to organize LP content.	25	75	50.0	25	75	**48.6**
Goal 5: LPs linked to instructional materials and to other LPs						
5.1 Upload instructional materials, such as PowerPoint slides.	15	50	28.7	15	50	**35.6**
5.2 Create link to online resources, such as Intranet web sites.	15	35	27.5	10	35	**26.9**
5.3 Create link to other LPs.	15	70	43.8	15	70	**37.5**
Goal 6: Mechanism for collaboration and review						
6.1 Select collaborators for an LP.	30	60	39.5	20	60	**40.7**
6.2 Route LP to collaborators and record activity log.	5	40	27.0	5	40	**23.9**
6.3 Enable collaborators to provide feedback on LP content.	30	35	33.5	25	40	**35.4**
Goal 7: Program-level data reported to training managers						
7.1 Input standardized data for training programs.	40	85	56.2	40	75	**60.1**
7.2 Create multiple views of program data.	15	60	43.8	25	60	**39.9**
Goal 8: Instances of training are recorded						
8.1 Input standardized data for instances of training delivery.	10	60	42.5	20	60	**41.7**
8.2 Archive instances of training delivery.	40	90	57.5	40	80	**58.3**
Goal 9: Training content available to authorized users						
9.1 Save and access LPs and training program data from a single user interface.	5	30	19.8	5	30	**18.4**
9.2 Track versions of LPs.	20	30	23.2	10	30	**23.0**
9.3 Apply access controls to stored training data.	20	40	28.8	20	40	**27.7**
9.4 Search training content across LPs and programs.	20	50	28.2	20	40	**30.9**
Goal 10: Training development process that follows established policy						
10.1 Review policy governing training development.	10	30	20.0	10	40	**22.9**
10.2 Route LP to approvers and record activity log.	40	80	57.5	20	80	**51.1**
10.3 Conduct annual review of LPs.	10	40	22.5	10	40	**26.0**
Goal 11: FLETA documentation generated from system						
11.1 Identify FLETA-related metadata from LPs and programs.	30	40	35.0	25	40	**33.3**
11.2 Input data into FLETA-specific forms.	20	30	23.8	20	30	**24.9**
11.3 Generate FLETA statistics and reports.	20	30	25.0	20	32	**26.0**
11.4 Save and archive reports.	10	20	16.2	5	20	**15.8**

Table 2. Results of task voting.

After the Round 2 (final) mean score was calculated for each task's priority relative to its goal (see Table 2), the only remaining activity was to calculate the *absolute* priority of each task as it compares to tasks that support other goals. The first step was to distribute the scores for the preferences among the goals that support them. The relationships between goals and preferences were established by the previous study and are depicted in Figure 5.

At this point it is useful to refer to tasks as LLRs and to goals as HLRs. Figure 5 is functionally the same as the previous goal diagram (see Figure 4). However, for the sake of presentation it has been modified such that the preferences (and their scores) appear on the left side while the HLRs and LLRs are listed on the right. The arrows indicate relationships between HLRs and preferences, and their styles have been altered to reduce confusion.

To calculate the priority of an HLR (P_{HLR}), one needs simply to add its share of the score for each preference that it supports. For example, Figure 5 illustrates how HLR 1 supports the *Development*, *Versions*, and *Reporting* preferences, which scored 18.3, 12.2, and 17.7 during the preference voting. Each preference distributes its points equally among the goals that support it. In this case, the *Development*, *Versions*, and *Reporting* preferences distributed a per-HLR share of 3.05, 6.1, and 2.53, respectively. Adding these numbers resulted in a P_{HLR} of 11.68 for HLR 1. All of the other HLR priorities were calculated in the same way, by adding up their share of the preference scores.



Priorities for Goals and Tasks

P_{HLR}	Goals (HLRs) and Tasks (LLRs)	P_{LLR}
11.68	**HLR 1: Facilitated creation of LPs**	
	LLR 1.1 Use a template to create a new LP.	**24.8**
	LLR 1.2 Reuse an existing LP (as a "pony") to create a new LP.	**21.6**
	LLR 1.3 Input standardized data for LPs.	**17.6**
	LLR 1.4 Apply ad hoc metadata (tags) for individual LPs.	**15.3**
	LLR 1.5 Record a justification for modifying an LP.	**20.7**
5.58	**HLR 2: LPs linked to centralized job tasks**	
	LLR 2.1 Create and manage a selectable job task list for each job.	**41.9**
	LLR 2.2 Select applicable job task(s) for each LP.	**58.1**
10.38	**HLR 3: LPs linked to policies and policy subsections**	
	LLR 3.1 Create and manage a selectable and nested list of section references for each FBI policy.	**44.3**
	LLR 3.2 Select applicable policy section(s) for each LP.	**28.0**
	LLR 3.3 Create hyperlink(s) to policy section(s).	**27.7**
5.58	**HLR 4: Learning objectives tracked to LPs**	
	LLR 4.1 Define terminal and enabling learning objectives for each LP.	**51.4**
	LLR 4.2 Use learning objectives to organize LP content.	**48.6**
11.2	**HLR 5: LPs linked to instructional materials and to other LPs**	
	LLR 5.1 Upload instructional materials, such as PowerPoint slides.	**35.6**
	LLR 5.2 Create link to online resources, such as Intranet web sites.	**26.9**
	LLR 5.3 Create link to other LPs.	**37.5**
13.28	**HLR 6: Mechanism for collaboration and review**	
	LLR 6.1 Select collaborators for an LP.	**40.7**
	LLR 6.2 Route LP to collaborators and record activity log.	**23.9**
	LLR 6.3 Enable collaborators to provide feedback on LP content.	**35.4**
2.53	**HLR 7: Program-level data reported to training managers**	
	7.1 Input standardized data for training programs.	**60.1**
	7.2 Create multiple views of program data.	**39.9**
2.53	**HLR 8: Instances of training are recorded**	
	8.1 Input standardized data for instances of training delivery.	**41.7**
	8.2 Archive instances of training delivery.	**58.3**
22.21	**HLR 9: Training content available to authorized users**	
	LLR 9.1 Save and access LPs and training program data from a single user interface.	**18.4**
	LLR 9.2 Track versions of LPs.	**23.0**
	LLR 9.3 Apply access controls to stored training data.	**27.7**
	LLR 9.4 Search training content across LPs and programs.	**30.9**
10.23	**HLR 10: Training development process that follows established policy**	
	LLR 10.1 Review policy governing training development.	**22.9**
	LLR 10.2 Route LP to approvers and record activity log.	**51.1**
	LLR 10.3 Conduct annual review of LPs.	**26.0**
4.8	**HLR 11: FLETA documentation generated from system**	
	LLR 11.1 Identify FLETA-related metadata from LPs and programs.	**33.3**
	LLR 11.2 Input data into FLETA-specific forms.	**24.9**
	LLR 11.3 Generate FLETA statistics and reports.	**26.0**
	LLR 11.4 Save and archive reports.	**15.8**

Figure 5. The distribution of preference votes to supporting HLRs.

After the 100 tokens from the preferences were distributed among the eleven HLRs the absolute priority of the LLRs was ready to be computed. The absolute priority of an LLR was calculated by multiplying the priority of its parent HLR (P_{HLR}) by the relative priority of the LLR within its group (P_{LLR}) by a compensation factor (C_{HLR}), which is the number of LLRs comprising an HLR block (Berander & Svahnberg, 2009). As discussed earlier, the calculation P_{HLR} x P_{LLR} x C_{HLR} is designed to compensate for HLR groups with fewer LLRs. The final ranking of each LLR was based on this calculated score as depicted in Table 3.

HLR	P_{HLR}	LLR	P_{LLR}	C_{HLR}	P_{HLR} x P_{LLR} x C_{HLR}	**Final LLR Ranking**
HLR 1	11.68	LLR 1.1	24.8	5	1448.32	**7**
		LLR 1.2	21.6		1261.44	**10**
		LLR 1.3	17.6		1027.84	**14**
		LLR 1.4	15.3		893.52	**17**
		LLR 1.5	20.7		1208.88	**12**
HLR 2	5.58	LLR 2.1	41.9	2	467.60	**28**
		LLR 2.2	58.1		648.40	**22**
HLR 3	10.38	LLR 3.1	44.3	3	1379.50	**9**
		LLR 3.2	28.0		871.92	**18**
		LLR 3.3	27.7		862.58	**19**
HLR 4	5.58	LLR 4.1	51.4	2	573.62	**24**
		LLR 4.2	48.6		542.38	**25**
HLR 5	11.2	LLR 5.1	35.6	3	1196.16	**13**
		LLR 5.2	26.9		903.84	**16**
		LLR 5.3	37.5		1260.00	**11**
HLR 6	13.28	LLR 6.1	40.7	3	1621.49	**5**
		LLR 6.2	23.9		952.18	**15**
		LLR 6.3	35.4		1410.34	**8**
HLR 7	2.53	LLR 7.1	60.1	2	304.11	**29**
		LLR 7.2	39.9		201.89	**33**
HLR 8	2.53	LLR 8.1	41.7	2	211.00	**32**
		LLR 8.2	58.3		295.00	**31**
HLR 9	22.21	LLR 9.1	18.4	4	1634.66	**4**
		LLR 9.2	23.0		2043.32	**3**
		LLR 9.3	27.7		2460.87	**2**
		LLR 9.4	30.9		2745.16	**1**
HLR 10	10.23	LLR 10.1	22.9	3	702.80	**21**
		LLR 10.2	51.1		1568.26	**6**
		LLR 10.3	26.0		797.94	**20**
HLR 11	4.8	LLR 11.1	33.3	4	639.36	**23**
		LLR 11.2	24.9		478.08	**27**
		LLR 11.3	26.0		499.20	**26**
		LLR 11.4	15.8		303.36	**30**

Table 3: Calculating the priority ranking of lower-level requirements (LLRs).

It is not surprising that the highest priority LLRs are found in the HLR 9 block "Training content available to authorized users." That particular HLR garnered the highest percentage of preference tokens by supporting four different preferences, including half of those available from the *Access* preference. This is consistent with the written justifications from the participants, who assigned high value for these preferences and LLRs (tasks). On the other hand, the lower ranked LLRs were associated with HLR 7 "Program-level data reported to training managers" and HLR 8 "Instances of training are recorded." This also is not surprising because each of those HLRs were able to capture only a small share of the tokens belonging to the *Reporting* preference. Also, the compensation factor (C_{HLR}) did not help their ranking as there were only two LLRs for each of those HLRs.

What is interesting is to examine LLRs like 10.2 "Route LP to approvers and record activity log," which emerged from its fellow 20th and 21st ranked LLRs to be rated as the 6th most important requirement. Despite its parent HLR ("Training development process that follows established policy") receiving a below average share of the preference points, LLR 10.2 earned more than half of the votes of the three LLRs in the HLR 10. This could mean that the idea of having electronic routing and approval of LPs has special importance despite being associated with an underperforming HLR that supports only two preferences.

Summary and Future Considerations

The study fulfilled its purpose and research goal by producing a rich, agreed-upon list of validated and prioritized software requirements that satisfy the user needs for a proposed KM system. It did so by conducting a structured QA review followed by a combination of voting activities. All work was accomplished by applying the Delphi consensus-building technique through e-mail communications.

The primary purpose of the envisioned KM system was to enable the organization to coordinate its training programs with its policies and business processes (or, in this case, its job tasks). Such a gap in its management of curriculum created for the subject organization several challenges in the areas of Development, Versions, Reporting, Accountability, Access, and FLETA Requirements, which became preference categories identified in the previous study. For example, being unable to link lesson plans to policies meant that training content could not be *accessed* by searching on policy topics, training managers could not *report* on which policies were being trained, and *FLETA requirements* could not be satisfied by mapping lesson *versions* to policies that were in force at the time. Likewise, because job task lists were not available for lesson plan *development*, approvers could not hold lesson plan authors *accountable* for content that accurately and completely fulfilled business needs in the organization.

Participants weighed the relative importance of these problems (and their solutions) as they considered the overall preferences, or outcomes, of the proposed KM system as well as its functional goals and tasks (HLRs and LLRs, respectively). They drew upon their own experience and skill sets, as well as those of their colleagues, to form opinions and assign relative values to user needs that define a KM system that would be useful to all of them. The prioritized LLR list of Table 3 represents a culmination of their efforts.

Although validated techniques were used successfully to validate and prioritize the final list of user needs, caution should be exercised when interpreting the priorities. The rankings provide general guidance about which requirements are more important than others, but they should not be used as a strict standard for inclusion into the final KM project. In other words, just because LLR 7.2 "Create multiple views of program data" ranked last among the 33 requirements does not mean that it should be the first one to be eliminated, if indeed some requirements must be cut. The relationships between HLRs and preferences are a matter of judgment generated by consensus, but a few changes to those relationships could have a big impact on the final priority list. In addition, it seemed that participants with the more extreme positions during the Delphi voting were less likely to change their positions. A few changes in those votes could have altered the results as well.

Nevertheless, the outcomes from this study validate and contribute to research in the area of consensus building as well as requirements validation and prioritization. For example, the application of the Delphi method facilitated the resolution of conflict between stakeholders (Nuseibeh & Easterbrook, 2000), but it was accomplished completely via e-mail. The results add to the body of knowledge in the area of processes that affect opinion change in Delphi, which is a sparsely treated subject in the literature according to Bolger and Wright (2011). Moreover, this research answers the call for more investigation into customizing and combining QA inspections (Denger & Olsson, 2005) as e-mail interactions between the researcher and participant contributed to rich descriptions that greatly clarified the requirements.

Finally, the combination of prioritizing techniques demonstrates a novel way of producing prioritized software requirements. Indeed, using the Delphi method for requirements prioritization does not appear in the literature, although it is used to achieve other group judgments (Landeta, et al., 2011). Also, the method of assigning priority weights to preferences (Liaskos, et al., 2011) for the purpose of prioritizing goals, and ultimately lower-level requirements (Berander & Svahnberg, 2009), likewise has not been found in the literature. And although these techniques may not be scalable to large KM projects encompassing hundreds of requirements, they certainly are generalizable to other similar-sized software applications and to other large organizations.

Including the appropriate stakeholders as participants ensured that the KM system would be useful to a variety of users at Training Division. Embedded with their e-mailed votes and written justifications were several positive comments regarding the techniques and hope for the future KM system: "I appreciate the request for a second review—especially given the significant differences…"; "It would be fabulous to have a system that would handle these tasks!!"; "…if this existed, it would be a good thing!"

The output from this validation study was a prioritized set of software requirements formulated in natural language, which is the language of the customer (Denger and Olsson, 2005). Future work on this project includes the translation of those requirements into a formal version to be operationalized using standardized semantics (Letier & Lamsweerde, 2002). At the same time, standardized training metadata must be defined to support the new digital library (Hicks, Perkins, & Maurer, 2007), and existing policy documents must be examined to identify granular mapping points (Douglas, 2009) for linking to LPs. The task explanations and detailed comments captured during this study will aid substantially in the transformation of the user needs into formal software requirements. After this last step the requirements will be ready to present as part of a formal request for proposal (RFP).

Chapter Five

Conclusions, Implications, and Recommendations

Introduction

Large organizations have trouble creating and modifying training products to reflect their ever-changing business processes and corporate polices, resulting in ineffective and irrelevant training programs. The overall objective of the research project was to develop a high level, user-based set of criteria defining a KM system that coordinates an organization's processes and policies with its learning programs. To accomplish this goal, the project was divided into three individual studies. The first study established the theoretical foundation for the project by examining the literature for similar examples of the problem as well as any progress made towards solutions. The second study used these findings to inform the process of eliciting a strong set of user needs from a group of experts representing an organization's stakeholders. The third study validated these user needs by applying a QA review and prioritization to the list, resulting in a design specification that represents the most important features of the envisioned KM system. The outcome of these studies led to valuable conclusions, implications, and recommendations for future work.

Conclusions

The theoretical study explored how the existing literature addresses the problem of coordinating learning programs with business processes and organizational policies. The review uncovered research describing consequences associated with the problem, the theoretical basis for addressing the problem, and studies that offer partial solutions to the problem.

Training products that are not tied to an organization's policies and processes can become outdated very quickly. When a policy or process cannot be traced back to the instructional materials meant to communicate it to the workforce, then a change in that policy or process has little chance of being reflected in the current training. Outdated training content leads to ineffective training delivery, which can cost an organization in terms of wasted employee time (Barnum, 2002) as well as legal liability (Sorenson, 2002).

According to the literature, large organizations often change their strategies and policies, which generate changes in how they do business (Koschmider & Oberweis, 2007). As a result, the workforce needs to be trained on the new policies and processes. However, simply placing the training content in an electronic environment does not solve the problem because the alignment between a learning object repository or eLearning system to current corporate practices is not enforced (Martin, Leyking, & Wolpers, 2008; Newton & Doonga, 2007; Zhang & Su, 2007). In fact, neither the computer managed instruction (CMI) standard nor the Sharable Content Object Reference Model (SCORM) specification provide any way of connecting learning objects to policies or business processes. At the same time, the Business Process Modeling Language (BPML) does not facilitate the connection between business process models to either policies or training.

The literature also indicated that learning theory and knowledge management strategies should inform the development of technologies used to define business processes and policies and communicate them to the workforce. Employees have built for themselves mental schemas (Driscoll, 2005) and concept maps (Chang & Chang, 2008) to enable them to make sense of how business processes take place. These schemas may be exploited during a training needs analysis to help define training that is specific job roles (O'Brien and Hall, 2004). Generating and codifying these schemas form the basis for building KM solutions that transfer the knowledge between people in an organization (Hlupic, Pouloudi, & Rzevski, 2002).

Finally, the theoretical study revealed areas where researchers only addressed a portion of the problem. For example, Hawryszkiewycz (2005) recognized that training programs must be customized and adapted to changing corporate business processes, but he did not propose a tracking mechanism to keep them synchronized. Caetano, Pombinho, and Tribolet (2007) demonstrated how role-based competencies could be associated with business processes, but stopped short of suggesting a way to map them to policies or training. Similarly, Reijers, Mans, and van der Toorn (2009) developed a novel way of depicting complicated business process models, but they did not attempt to connect those models with any policies that authorized them or the instructional products that teach them. Finally, David, David, and David (2011) offered a skill set inventory to help a training provider focus on job-related tasks rather than theory. However, they did not mention the need to map those tasks to training products for gap analysis and content updates.

The output from the theoretical study exposed several gaps in the existing training-related KM research documented in the literature. The gaps pointed to the need for a technical mechanism to bring together an organization's training products, processes, and policies. The successful identification and support for this need became the starting point for the elicitation study.

The purpose for the elicitation study was to utilize stakeholders in a large government agency to identify high level software requirements for a KM system that would manage links between learning products (lesson plans), business processes (job tasks) and organizational policies. Specifically, the intent of the research was to determine the user needs regarding KM processes and technologies required to create those associations, and the methods used to acquire the user needs from on-board employees.

Seven participants were recruited from a larger group of active instructors and training managers in the FBI Training Division. Each participant was a domain expert as well as a user representative (Chakraborty, Sarker, & Sarker, 2010). Consistent with requirements elicitation (RE) studies found in the literature (Chakraborty, Sarker, & Sarker, 2010; Coughlin & Macredie, 2002; Nuseibeh & Easterbrook, 2000; Saiedian & Dale, 2000), a combination of RE methods were used to capture user needs from the participants.

First, each contributing member of the expert group was individually e-mailed a short set of open-ended questions that elicited detailed, candid responses regarding their experiences with training management. This elicitation method proved to be a very efficient way of collecting raw input from very busy people. Their "stories" (Laporti, et al., 2009) included negative on-the-job experiences as well as positive suggestions as to how training management problems may be resolved. Next, their stories were subjected to a content analysis to identify common features called activity themes. Each of the 37 activity themes was then associated with one or more categories, which represented high level outcomes (or benefits) of the envisioned KM system.

At times, participants needed to be reminded of the goals of the study and that not all training management issues (such as content ownership) may be resolved with a technical solution. Nevertheless, nearly all responses converged on the primary disconnect between lesson plans, job tasks, and policies. Other identified activity themes either related to this disconnect or were deemed essential to a KM system that would address the problem.

The webbed relationships between the activity themes and the categories were depicted in two different diagrams based on Soft Systems Methodology (Checkland, 1998; Coughlan & Macredie, 2000). One diagram depicted the current state of affairs and the other illustrated what a successful KM system would be required to do. Both figures were sent back to the expert panel members for their feedback and the diagrams were modified accordingly.

The outcome from the elicitation study was a high-level set of interrelated user needs that were traceable back to the original input from the experts. The defined user needs effectively addressed the goal of connecting lesson plans with job tasks and policies. The fact that the specification list was user-generated increases the likelihood of user buy-in for the future KM system and the traceability of the requirements ensures their accuracy (IEEE, 1998).

The purpose for the third study was to validate the user needs elicited from the previous study and determine their relative importance. An eight member expert panel, composed of instructors, training managers, and software specialists, was used to validate the user needs through three phases of e-mail-based collaboration. Each phase involved two rounds of Delphi interactions (Bolger & Wright, 2011; Landeta, et al., 2011).

The first phase involved a quality assurance (QA) review of the existing set of required software tasks that emerged from the elicitation study. An excel spreadsheet helped to facilitate the QA review by organizing these tasks into logical goals, imbedding descriptive explanations for every task, and then providing specific criteria that required a simple “yes” or “no” answer. If a task did not meet a particular criterion, the participant was asked to provide a recommended change. This exercise produced an unexpectedly detailed series of explanatory e-mails between the researcher and participants, which prompted the rewriting of several tasks. The e-mail interactions also provided rich descriptions and context that served to inform the next two phases of activity.

In the second phase participants were asked to assign relative importance to six preferences, or general outcomes, of the proposed KM system. These preferences were adopted from the categories identified in the previous study. The expert panel used a token-based voting technique (Liaskos, et al., 2011) to distribute 100 points among the preferences, and several members provided a justification for their votes. In general participants voiced very strong opinions on the importance of the preferences, which demonstrated the effectiveness of this Delphi technique for enabling members to provide quality input based on personal and professional experience. When the votes and justifications were compiled in a spreadsheet and shared with the group during the second round re-voting, the experts referred to each others' previous votes and justifications – to either adjust their own votes or to express their disagreement and thereby defend their scores. Regardless of which side was argued, these interactions accurately represented the issues involved, and served to provide context for what was important to the users.

The voting on preferences successfully prepared the participants for a similar voting activity during the third Delphi phase. This time, the participants voted on the individual tasks – representing user needs – the descriptions of which were fine-tuned during the first phase. Based on participant comments, the clarity of the tasks was probably the reason for the reduced debate regarding their assigned votes and justifications. Each member was allowed 100 tokens to distribute among tasks supporting a single goal, thus limiting the number of voting decisions to a maximum of five tasks per goal. Again, a spreadsheet was useful in being able to represent the mean scores, the range of scores, and justifications from the first round so that participants could adjust their re-votes accordingly. However, the second round of this phase neither produced much movement of votes nor inspired much debate among participants.

To arrive at the final (absolute) prioritization of the software requirements, all of collected data were used. A relational model was developed, based on the associations between activity themes and categories defined in the elicitation study. In the new model these concepts became known as software goals and preferences, respectively. The mean score obtained for each of the preferences in Phase Two were divided among the goals that supported them (Liaskos, et al., 2011), so that each goal was assigned its share of the original votes (tokens). Software tasks (also adopted from activity themes) that were grouped under goals were already assigned mean scores (or priorities) from Phase Three. Berander and Svahnberg (2009) considered these lower level requirements (LLRs) and higher level requirements (HLRs), and following their method the priority for each LLR was multiplied by the priority of its HLR and the number of LLRs assigned to that HLR. The product of these three numbers represented the absolute priority – or relative importance – for each LLR. Thus the goal of the third study was achieved.

The preceding discussion illustrates both the strengths and weakness of the research project as a whole. Methods were adopted from a variety of research sources, and the combination of those methods resulted in novel ways to develop software requirements using a group of experts. Indeed, using a story-based elicitation technique (Laporti, et al., 2009) to feed an SSM model (Coughlan & Macredie, 2000) was an effective way to transform raw, expert contributions into diagrams that represented high level user needs. In the same way, preference weighting influenced by the research of Liaskos, et al. (2011) was combined with the prioritization of the goals (HLRs) and tasks (LLRs) inspired by Berander and Svahnberg (2009). The use of these two techniques enabled the experts to evaluate importance at both the high level (general outcomes of the envisioned KM system) as well as the lower level (specific tasks required by the KM system).

However, the blending of methods came with a price. Different researchers refer to similar concepts using distinct terminology, so combining methods meant finding a way to translate terms that may have particular meanings relative to a study. For example, *activity themes* harvested from the raw stories of the elicitation study became known as software *goals* and *tasks* when creating relationship models in the validation study. Moreover, these constructs later became known as *HLRs* and *LLRs* respectively, as another method was applied to their prioritization. Similarly, activity theme *categories* from the elicitation study became known as *preferences* for purposes of voting on the importance of outcomes in the validation study. Finally, all of these concepts were loosely referred to as high level *user needs*, *software requirements*, or *software specifications*, to avoid the more formal, technical descriptions that most participants could not understand. Future projects of a similar nature should strive to clarify these important terms.

Implications

With regard to managing large training programs, several benefits may be taken from this research. That learning objects should be coordinated with business processes and policies is an opinion found in the literature as well as expressed by two expert panels. Maintaining links to processes and policies not only establishes the relevance and authority of the lesson plan, but it also ensures that the latest policies and processes are actually being trained to the workforce. With effective BP-to-LP links, reports could be generated to provide insight into how well business processes (or job tasks) are being covered by training products. In the same way, OP-to-LP reports could determine the extent to which training programs are facilitating compliance with specific policies.

For leaders of an organization similar to the one examined in this research, the findings will help them not only to understand the nature of problems that plague their training programs but also to develop a technical solution to solve them. Maintaining a repository of electronic lesson plans, for example, would be more effective if the lesson plans could be standardized, searchable, linked to BPs and OPs, and put into workflow to track collaboration and approval.

Aside from greatly increasing efficiency, such functionality would undoubtedly meet many of the administrative, operational, and strategic requirements of the organization.

With a carefully designed KM system, efficient search and reporting features would facilitate the location and modification of instructional materials that have been outdated by changing processes and policies. Data calls for accountability and compliance would be streamlined by access to metadata embedded into lesson plans. Indeed, nearly everyone associated with an organization's training program – curriculum managers, instructors, supervisors, course designers, and executives – would benefit from a KM system that implements the findings of this research.

The process of arriving at the KM software requirements themselves could be considered a secondary contribution of the research. The quality of software requirements is as good as the group of participants who develop them. Unfortunately, the best people are most often the busiest people, and are usually so tied down to their jobs that they cannot commit to the hours – and sometimes days and weeks – of meetings required to harvest their expertise. Consider a face to face meeting of ten people, for example. If no one talks over each other and everyone has equal time to speak, then each participant will spend only 10% of the meeting time providing input. For the individual, therefore, 90% of the meeting is spent *not* providing input. Of course, face to face meetings could involve a significant amount of non-verbal communication, contextual details, and negotiating the meaning of concepts. However, if one of the primary goals of a collaborative effort (such as requirements elicitation) is to collect raw input, it seems far more efficient to use a method that enables participants to use 100% of their time providing raw input. This objective may be achieved using asynchronous communication.

Indeed, software requirements are typically developed using methods that involve a facilitator and a group of subject matter experts who attend a series of face to face meetings

(Coughlan & Macredie, 2002). The Delphi method offered a way to collect input from experts without requiring face-to-face meetings. But although the Delphi method commonly has been used to build consensus through asynchronous communication (Bolger & Wright, 2011; Graefe & Armstrong, 2011; Landeta, et al., 2011), an example could not be found in the literature where it had been used to develop software requirements.

In this project the successful use of Delphi e-mail communications, whereby participants are asked to share their stories and vote on requirements – in their own time and without distractions from other participants – enabled the best people to participate. Moreover, the anonymity of the data collected ensured that opinions could be shared without fear of intimidation or repercussions. The resulting data from open-ended questions were often difficult to analyze and compile, but they provided a very accurate representation of the user desires for the proposed KM system.

In summary, the three study approach to this research provided a no-cost, low-risk method for an organization to define user needs for a future KM system with committing resources to the project. Avoiding the high costs and risks of typical requirements elicitation efforts (Davey & Cope, 2008; Laporti, et al., 2009), this project identified the best contributors, maximized their time, facilitated their expert responses, and distilled their collective wisdom into a set of validated, prioritized requirements. The user-defined, natural language requirements are traceable back to the original input from the participants, which will provide context for the future interpretation and clarification of the requirements.

Recommendations

The organization used as the subject for this investigation was provided a validated and prioritized list of requirements that represent the needs of the population of future users. It is recommended that the organization now determine whether its existing IT systems could be leveraged to meet any of the identified requirements. The owners and maintainers of those systems should be involved in the discussion, during which the requirements may be clarified or reworded if required. Eventually, a budget and timeline will need to be established to build the KM system and account for any functional gaps.

Some of the biggest challenges, however, will be to determine specific metadata for lesson plans, as well as how to link the lesson plans to processes (job tasks) and policies. Stakeholders will need to decide which metadata should describe training programs versus the lesson plans, and which type of data will be collected regarding each delivery of the training programs. The job task list needs to be finalized and assigned a clear owner, so that it may be modified to accommodate future organizational needs. Finally, existing policies must be examined to determine the most appropriate level of granularity (e.g., sections, subsections) to which lesson plans should link. These linkages will determine the effectiveness of reporting capabilities that reveal training gaps for both job tasks and policies.

As previously described, a contribution of this research was the demonstration of an efficient means to collect raw input from a group of expert contributors. Considering the positive results from the elicitation and validation activities, the e-mail-based Delphi method could be expanded, perhaps, to include a larger problem scope that would produce a larger number of software requirements. To avoid overwhelming participants, however, the problem scope could be broken down into discreet areas so that contributors are asked to make decisions on a limited number of issues at any given time. Multiple teams of experts could be used, each one simultaneously concentrating on a separate problem area, and then they could switch to another problem area. The experts in each problem area would help generate SSM diagrams which could then be compared as the expert groups rotate through the problem areas. The multiple group method would expand the population of expert contributors and increase confidence in the resulting requirements list, without increasing the workload for a group at any particular time.

It is recommended that future training-related KM projects follow the methods described in this three study investigation. It would be interesting to see how other large organizations adopt these methods to identify their training management problems and develop software requirements to address those problems. Once a new KM system is deployed, a usability study could be used to measure exactly how much time and resources are saved through its implementation. Finally, future research projects should pursue consensus-building methods that include the most knowledgeable and motivated participants without impacting their operational obligations and work schedules. The project described in this paper offers an effective strategy to accomplish these goals.

Appendix A

Acronyms and Definitions

AHP	Analytic Hierarchy Process
BP	Business Process
BPM	Business Process Management
BPML	Business Process Modeling Language
BPMN	Business Process Modeling Notation
CALO	Company, Agency, or other Large Organization
CMI	Computer Managed Instruction
CMS	Course Management System
COTS	Commercial Off The Shelf
EPSS	Electronic Performance Support System
ERP	Enterprise Resource Planning
FLETA	Federal Law Enforcement Training Accreditation
HCV	Hierarchical Cumulative Voting
HLR	High-level Requirement
IEEE	Institute of Electrical and Electronics Engineers
IRB	Institutional Review Board
JAD	Joint Application Design
JT	Job Task
KM	Knowledge Management
LLR	Low-level Requirement
LO	Learning Object

LP	Lesson Plan
NGT	Nominal Group Technique
OLE	Online Learning Environment
OP	Organizational Policy
QA	Quality Assurance
RE	Requirements Elicitation
RFP	Request for Proposal
SCORM	Sharable Content Object Reference Model
SSM	Soft Systems Methodology
SRS	Software Requirements Specification
TEL	Technology-enhanced Learning
TNA	Training Needs Analysis

References

Barnum, Carol M. (2002). *Usability Testing and Research.* New York: Pearson Education.

Beck, R. J. (2005). Learning objects: an introduction. *Center for International Education*. University of Wisconsin – Milwaukee. Modified June 2, 2005. Retrieved March 23, 2006 from http://www.uwm.edu/Dept/CIE/AOP/learningobjects.html.

Berander, P. and Svahnberg, M. (2009). Evaluating two ways of calculating priorities in requirements hierarchies – An experiment on hierarchical cumulative voting. *The Journal of Systems and Software, 82,* 836-850.

Bertot, J.C. (2012). Author information pack. *Government Information Quarterly*. Retrieved March 21, 2012 from http://www.elsevier.com/wps/find/journaldescription.cws_home/620202/authorinstructions.

Bolger, F. and Wright, G. (2011). Improving the Delphi process: Lessons from social psychological research. *Technological Forecasting & Social Change, 78,* 1500-1513.

Byrd, T.A., Cossick, K.L., and Zmud, R.W.(1992). A synthesis of research on requirements analysis and knowledge acquisition techniques. *MIS Quarterly*, 1992. 117-138.

Caetano, A., Pombinho, J., & Tribolet, J. (2007). Representing Organizational Competencies. SAC'07, ACM, 1257-1262.

Capuano, N., Gaeta, M., Ritrovato, P., and Salerno, S. (2008). How to integrate technology-enhanced learning with business process management. *Journal of Knowledge Management, 12*, 6, 56-71.

Chakraborty, S., Sarker, S. and Sarker, S. (2010). An exploration into the process of requirements elicitation: A grounded approach. *Journal of the Association for Information Systems, 11,* (4), 212-249.

Chang, S. L. and Chang, Y. (2008). Using online concept mapping with peer learning to enhance concept application. *The Quarterly Review of Distance Education, 9,* (1), 17-27.

Checkland, Peter B. (1998). *Systems Thinking, Systems Practice*, John Wiley & Sons Ltd. 1981, 1998. ISBN 0-471-98606-2.

Coughlan, J. and Macredie, R. (2002). Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering, 7,* 47-60.

Cross, J. (2003). Workflow-based elearning and the bottom line. *Internet Time Group.* Posted on July 17, 2003. Retrieved October 1, 2006 from http://www.internettime.com/blog/archives/000551.html#000551.

Davey, B. and Cope, C. (2008). Requirements elicitation – what's missing? *Issues in Informing Science and Information Technology, 5,* 543-551.

David, F. R., David, M. E., and David, F. R. (2011). What are business schools doing for business today? *Business Horizons, 54,* 51-62.

Denger, C. and Olsson, T. (2005) Quality assurance in requirements engineering. In: A. Aurum & C. Wohlin (Eds), *Engineering and managing software requirements* (pp 163-185). Berlin, Germany: Springer.

Douglas, I. (2009). An object and performance framework for implementation of web-based knowledge sharing technology. *Journal of Theoretical and Applied Electronic Commerce Research, 4,* 1, 57-71.

Driscoll, Marcy P. (2005). Psychology of Learning for Instruction (3rd ed.). Boston, MA: Allyn and Bacon.

Ellis, T. J., and Levy, Y. (2010). A guide for novice researchers: Design and development research methods. *Proceedings of Informing Science & IT Education Conference (InSITE) 2010,* 107-118.

FBI (2006). Business process reengineering investigative results (version 1.1). *Federal Bureau of Investigation.* Published June 15, 2006. This is an Unclassified/For Official Use Only (FOUO) document not available to the public.

FBI (2011). Domestic Investigations and Operations Guide. *Federal Bureau of Investigation.* Published October 15, 2011. Retrieved May 11, 2012 from http://vault.fbi.gov/FBI%20Domestic%20Investigations%20and%20Operations%20Guide%20%28DIOG%29/fbi-domestic-investigations-and-operations-guide-diog-2011-version.

Feldstein, M. (2006). There's no such thing as a learning object. *eLearn Magazine, 2006,* 5. 2-3.

Fischer, L. (2002). BPML 1.0 Final and Business Process Modeling Notation WD. *OASIS Open Pages*. Modified November 13, 2002. Retrieved February 22, 2006 from http://xml.coverpages.org/BPML10-BPMN.html.

Gay, L.R., Mills, G.E., and Airasian, P. (2009). *Educational Research.* New Jersey: Pearson Education.

Graefe, A. and Armstrong, J.S. (2011). Comparing face-to-face meetings, nominal groups, Delphi and prediction markets on an estimation task. *International Journal of Forecasting, 27,* 183-195.

Gupta, S. and Bostrom, R. P. (2006). End-user training methods: What we know, need to know. *SIGMIS-CPR '06, April 13-15, 2006.* 172-182.

Hawryszkiewycz, I.T. (2005). A framework for integrating learning into business processes. *SEARCC 200: ICT: Building Bridges Conference.* 23-28.

Hicks, E.A., Perkins, J., and Maurer, M.B. (2007). Application profile development for consortial digital libraries: An OhioLINK case study. *Library Resources & Technical Services, 51,* 2, 116-131.

Hlupic, V., Pouloudi, A., and Rzevski, G. (2002). Towards an integrated approach to knowledge management: 'Hard', 'soft' and 'abstract' issues. *Knowledge and Process Management, 9,* 2, 90-102.

IEEE Recommended Practice for Software Requirements Specifications, IEEE Standard 830, 1998.

Israel, J. W. (2012). Why the FBI can't build a case management system. *Computer, 45,* (6), 73-80.

Japenga, R. (2011). How to write a software requirements specification. *Microtools, Inc.* Retrieved March 5, 2012 from www.microtoolsinc.com/Howsrs.php.

Kankanhalli, A., Tan, B.C.Y. (2004). A review of metrics for knowledge management systems and knowledge management initiatives. *Proceedings of the 37th Hawaii International Conference on System Sciences, HICSS37*, IEEE Computer Society, Washington, DC.

Karlsson, J., Wohlin, C., and Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology, 39,* 939-947.

Kay, D. (2003). E-learning: Drivers, developments, and decisions. *Multimedia Information Technology, 9,* 26-29.

Kiper, J. R. (2008). Online strategies for teaching business processes in large organizations. *Journal of Instruction Delivery Systems, 22,* 14-18.

Koschmider, A. and Oberweis, A. (2007). How to detect semantic business process model variants? *SAC'07,* March 11-15, 2007, 1263-1264.

Krause, M. (2012). Vigilance Fatigue in Policing. *FBI Law Enforcement Bulletin, December 2012,* 3-11.

Landeta, J., Barrutia, J., and Lertxundi, A. (2011). Hybrid Delphi: A methodology to facilitate contribution from experts in professional contexts. *Technological Forecasting & Social Change, 78,* 1629-1641.

Laporti, V., Borges, M.R.S. and Braganholo, V. (2009). Athena: A collaborative approach to requirements elicitation. *Computers in Industry, 60,* 367-380.

Liang, H. and Xue, Y. (2004). Coping with ERP-related contextual issues in SMEs: a vendor's perspective. *Journal of Strategic Information Systems, 13* (2004), 399-415.

Liaskos, S., McIlraith, S.A., Sohrabi, S., and Mylopoulos, J. (2011). Representing and reasoning about preferences in requirements engineering. *Requirements Engineering, 16,* 227-249.

Loucopoulos, P. (2012). About this journal. *Requirements Engineering*. Retrieved March 30, 2012 from http://www.springer.com/computer/swe/journal/766.

Martin, G., Leyking, K., and Wolpers, M. (2008). Business process-driven learning. *IST-Africa 2006 Conference and Exhibition*, Pretoria, South Africa, May 2006. IIMC International Information Management Corporation, 2006. ISBN: 1-905824-01-7.

Martínez-Ortiz, I., Sierra, J., and Fernández-Manjón, B. (2009). Language-driven, technology-enhanced instructional systems design, *Learning in the Synergy of Multiple Disciplines, Lecture Notes in Computer Science,* 2009, Volume 5794/2009, 725-731.

May, D. and Taylor, P. (2003). Knowledge management with patterns. *Communications of the ACM*. July 2003. 46,7, 94-99.

Miller, J. (2005). FBI studying ways to improve bureau's business lines. *Washington Technology.* Modified 09/23/2005. Retrieved January 26, 2006 from www.washingtontechnology.com/news/1_1/daily_news/27021-1.html.

Newton, R. and Doonga, N. (2007). Corporate e-learning: Justification for implementation and evaluation of benefits. A study examining the views of training managers and training providers. *Education for Information, 25,* 111-130.

Nuseibeh, B. and Easterbrook, S. (2000). Requirements engineering: A roadmap. *ICSE '00 Proceedings of the Conference on The Future of Software Engineering,* 37-46.

O'Brien, E. and Hall, T. (2004). Training needs analysis - The first step in authoring e-learning content. *2004 ACM Symposium on Applied Computing*. 935-939.

Perini, A., Ricca, F. and Susi, A. (2009). Tool-supported requirements prioritization: Comparing the AHP and CBRank methods. *Information and Software Technology, 51*, 1021-1032.

Raybould, B. (1995). Performance support engineering: An emerging development methodology for enabling organizational learning. *Performance Improvement Quarterly, 8,* (1), 7-22.

Reijers, H.A., Mans, R.S., and van der Toorn, R.A. (2009). Improved model management with aggregated business process models. *Data & Knowledge Engineering, 68,* 221-243.

Saiedian, H. and Dale, R. (2000). Requirements engineering: Making the connection between the software developer and customer. *Information and Software Technology, 42,* 419-428.

Scheinholtz, L.A. and Wilmont, I. (2011). Interview patterns for requirements elicitation. *Requirements Engineering: Foundation for Software Quality 17th International Working Conference, REFSQ 2011,* 72-77.

Song, S., Nerur, S., and Teng, J.T.C. (2007). An exploratory study on the roles of network structure and knowledge processing orientation in work unit knowledge management. *The DATA BASE for Advances in Information Systems, 38,* 2. 8-26.

Sorenson, S. M. (2002). Training for the long run: a cost-benefit perspective. *Business News Publishing Co*. Posted in May, 2002. Retrieved October 15, 2006 from http://findarticles.com/p/articles/mi_m0BPR/is_5_19/ai_87078406.

Sun, J. (2011). Human choice between computerized and traditional methods: Assessing tool readiness from the activity perspective. *Computers in Human Behavior, 27,* 391-401.fig

Trkman, P. (2009). The critical success factors of business process management. *International Journal of Information Management* (2009), doi: 10.1016/j.ijinfomgt.2009.07.003.

Turban, E. and Aronson, J. E. (2001). Decision Support Systems and Intelligent Systems (6th edition). New Jersey: Prentice Hall.

Van den Akker, J. (1999). Principles and methods of development research. In J. van den Akker, *et al* (eds). *Design Approaches and Tools in Education and Training.* Dordrecht: Kluwer Academic Publishers.

Virine, L. and Rapley, L. (2003). Visualization of probabilistic business models. *Proceedings of the 2003 Winter Simulation Conference.* 1779-1786.

Wei, C., Hu, P.J., and Chen, H. (2002). Design and evaluation of a knowledge management system. *IEEE Software.* May/June 2002. 56-59.

Wickramasinghe, N., Fadlalla, A., and Sharma, S.K. (2004). The KM infrastructure: making implicit knowledge assets explicit. *Proceedings of the 37th Hawaii International Conference on System Sciences.* 1-8.

Zhang, H. & Su, H. (2007). Reforming computing education with new web technologies. *The Journal of Computing Sciences in Colleges, 23,* 150-156.

Zwass, V. (2012). Journal profile. *Journal of Management Information Systems.* Retrieved March 23, 2012 from http://www.jmis-web.org/profile/.