Presented to the Interdisciplinary Studies Program:

UNIVERSITY OF OREGON
APPLIED INFORMATION MANAGEMENT

Applied Information Management
and the Graduate School of the
University of Oregon
in partial fulfillment of the
requirement for the degree of
Master of Science

CAPSTONE REPORT

Combining Agile Software Development Practices and Knowledge Transfer Systems to Support Knowledge Sharing Between Subject Matter Experts (SMEs) and Software Developers

Stephan A. Hancock
Software Development Manager
Rentrak Corporation

University of Oregon
Applied Information
Management
Program

February 2012

Continuing Education
1277 University of Oregon
Eugene, OR 97403-1277
(800) 824-2714

Approved by

_____

Dr. Linda F. Ettinger
Senior Academic Director, AIM Program

Combining Agile Software Development Practices and Knowledge Transfer Systems to Support Knowledge Sharing Between Subject Matter Experts (SMEs) and Software Developers

Stephan A. Hancock

Rentrak Corporation

Abstract

This annotated bibliography reveals that knowledge transfer systems, when combined with selected agile software development practices, may help overcome barriers that prevent knowledge sharing between subject matter experts (SMEs) and software developers. Social aspects of agile development practices (pair programming) and approaches to knowledge transfer (community of practice) are shown to address barriers including hoarding, linguistics, and bureaucracy. The goal is to mitigate risks associated with potential loss of intellectual capital related to legacy systems.

Keywords: agile software development, knowledge transfer, community of practice, legacy system, pair programming, subject matter expert

Table of Contents

Introduction

Problem Area

Knowledge is defined in this study within the context of organizations and the individuals within them and refers to "not truth, but effective performance: not 'what is right' but 'what works' or even 'what works better' where better is defined in competitive and financial contexts" (Demarest, 1997, p. 375). Knowledge is described in two dimensions: (a) *explicit*, knowledge that can be systematically communicated, and (b) *tacit*, knowledge based in specific context, action, or experience (Nonaka, 1994). Both tacit and explicit dimensions are key concepts in organizational knowledge and come together to create a knowledge network, which is individual knowledge amplified and crystalized by the organization (Nonaka, 1994); within this network tacit knowledge is viewed as bound to the individual, and explicit knowledge is codified in the company (Nonaka, 1994).

Knowledge within a company takes one of three forms: (a) embedded within an object, (b) held within individuals (as either tacit or explicit knowledge), or (c) held within a community, (also as either tacit or explicit knowledge) (Wasko & Faraj, 2000). According to Wasko and Faraj (2000), organizational knowledge is the sum of these parts: (a) knowledge embedded in objects comes, for example, in the form of documents and databases, (b) knowledge existing in the form of what each individual knows as it applies to the company such as content types of corporate databases or working knowledge of legacy computer systems, and (c) knowledge existing within the community "in the form of routines and shared languages, narratives and codes" (p. 158). Bontis (1999) describes organizational knowledge, when collected from people, routines, and the communities within a company, as *intellectual capital* (IC) (p. 444). Intellectual capital is defined as tangible and intangible assets held by the company

in the form of explicit knowledge and tacit knowledge (Rus & Lindvall, 2002, p. 29). The goal of this study is to examine how to utilize agile software development practices (Beck et al., 2001; Cockburn & Williams, 2000) to convert the tacit knowledge held by individuals who are part of an agile development team into more widely-held community knowledge (Crawford, Castro, & Monfroy, 2006), thus reducing the potential for loss of intellectual capital when a key subject matter expert is no longer employed at the company (Massingham, 2008). In this study *agile software development* is defined as a set of principles that embody twelve concepts designed to allow self-organizing development teams to deliver software early and within short time frames, while working closely with the business customer directly (Beck et al., 2001).

Knowledge management (KM) is defined by Davenport and Prusak (as cited in Bjørnson & Dingsøyr, 2008) as "a method that simplifies the process of sharing, distributing, creating, capturing, and understanding a company's knowledge" (sect. 2.1, para 1). Alavi and Leidner (2001) create a framework for knowledge management consisting of four categories: (a) creation, (b) storage and retrieval, (c) transfer, and (d) application (p. 115). This study focuses on the third category, *knowledge transfer* which is the sharing, or movement of one group or individual's experience onto another group (Argote & Ingram, 2000). Knowledge transfer consists of seven basic processes: (a) creation, (b) identification, (c) collection, (d) organization, (e) sharing, (f) adaptation, and (g) usage (O'Dell & Grayson, 1998, p.7). There are various types of systems that facilitate knowledge transfer within an organization including "face-to-face interactions (planned or ad hoc), mentoring, job rotation, and staff development" (Alavi & Leidner, 1999, p. 3) or company-based community of practice (Wasko & Faraj, 2000, p. 160). To remain competitive, an organization needs to transfer knowledge (Argote & Ingram, 2000). However, Disterer (2001) reports that there are barriers to knowledge transfer including (a)

obstacles faced by individuals such as loss of power, uncertainty as to the value of the knowledge held by the individual, and a lack of motivation to share, and (b) social barriers such as organizational bureaucracy, and misalignment between individual and corporation paradigms.

KM with regard to software engineering is more specifically defined as a process that includes strategies to prevent or marginalize risk caused by attrition of subject matter experts (SMEs) (Rus & Lindvall, 2002, p. 29). The process requires a steep learning curve to understanding the knowledge, and often involves rework or repetition when lessons are not learned (Rus & Lindvall, 2002, p. 29). Within that definition, the role of the subject matter expert is of central interest in this study.

Agile software development practices show that "there are strong social forces at play in agile teams that underscore the value of agile methodologies" (Whitworth & Biddle, 2007, p 35). One of the four central values of the agile software development process "emphasizes the relationship and communality of software developers and the human role reflected in the contracts", suggesting that social contact was a goal of agile since the beginning (Abrahamsson, Salo, Ronkainen, & Warsta, 2002, p. 11). There are differences between specific forms of agile development practices, but most include the following attributes: (a) regular updates or releases within a limited time frame; (b) adaptive, light-weight requirements (stories); (c) short, iterative development cycles; (d) continuous testing; and (e) collective working style and ownership, including stakeholders and developers (Abrahamsson, Salo, Ronkainen, & Warsta, 2002, p.14). Considering the perspective of Pawlowski and Robey (2004) that knowledge transfer is achieved within the enterprise as a "process of social participation in which members interact with more experienced members who convey both tacit and explicit knowledge through personal contact"

(p. 649), the assumption underlying this study is that knowledge transfer can be improved by combining knowledge transfer and agile software development practices.

One KM system that encourages knowledge transfer is that of community of practice (Wasko & Faraj, 2000, p. 160). A community of practice (CP) is defined as "groups of people informally bound together by shared expertise and passion for a joint enterprise" (Wenger & Snyder, 2000, p. 139). How might this CP definition align with the practices of agile software development? The following two examples taken from the agile process demonstrate the potential. Whitworth and Biddle (2007) state that although agile teams may attach specific technological elements to individuals, these individuals regularly share with the team any progress made and allow the team to revisit the subject in enough depth to create a pool of common knowledge. Further, they believe that "agile teams allow for an extremely high level of social support and accountability during software development" (Whitworth & Biddle, 2007, p. 3); this software development community within an organization shows common values and understanding held by followers of an agile culture (p. 6). In addition, one of the practices commonly used in agile software development is *pair programming* (PP). According to Cockburn and Williams (2000), PP supports the process of sharing knowledge through:

- Continuous reviews of the code creates tacit knowledge of the system undergoing change or development;
- Problem solving, whereas the programming pair both can address an issue, discuss options and potentially develop solutions;
- Learning, in which "knowledge is constantly being passed between partners, from tool usage tips (even the mouse), to programming language rules, design and programming idioms, and overall design skill" (p. 7); and

- Team building, the effect caused by people learning to work together, rapid sharing of lessons learned, a reduction of hidden agendas, and frequent group communication (p. 8).

Purpose

The purpose of this annotated bibliography is to present literature that addresses the topic of how agile software development practices (Beck et al., 2001; Cockburn & Williams, 2000) may potentially apply to the process of knowledge transfer (Argote & Ingram, 2000) in small- and mid-sized companies (U.S. Small Business Administration, 2011) in order to relieve the risks to loss of intellectual capital posed by subject matter experts while at the same time dispersing knowledge to the wider development community. This study addresses selected practices used within agile software development, with interest in practices that involve face-to-face interactions between SMEs and other developers. In addition, focus is on two of the most widely-used agile methodologies, Scrum and XP (Begel & Nagappan, 2007).

This study is designed to help IT managers to identify agile approaches with the best potential to support the process of knowledge transfer between SMEs and the software development teams. Specific examples show instances where certain agile software development practices tie into knowledge transfer systems, (e.g. community of practice) in ways that relieve the risks to loss of intellectual capital.

The larger context for the bibliography is framed by exploring the concepts of *tacit* and *explicit* knowledge (Nonaka, 1994), the subject matter expert (SME) (Davenport & Prusak, 2000), and knowledge management (KM) (Davenport & Prusak, 2000). Once the context for the bibliography is established, a selected set of *agile software development practices* is presented as a way to enable knowledge transfer in a business environment. For example, the set includes examination of the use of *pair programming* (PP) as a way to support one form of knowledge

transfer, known as *community of practice* (CP) (Wenger & Snyder, 2000). Community of practice is defined and possible connections are made (Kahkonen, 2004) between basic concepts of CP and the selected practices used in agile software development.

Research Questions

Main question: What agile software development practices offer the best potential to support knowledge transfer of legacy systems between software development subject matter experts and the wider development community (including novice software developers) in mid-sized companies?

Sub-questions include:

What are the key reported potential risks to loss of intellectual capital when a SME is no longer able to work in their area of expertise (Boehm, 1991; Bontis, 1999; Crawford, Castro & Monfroy, 2006; Massingham, 2008)?

- What are the key reported barriers to knowledge transfer between a SME and other developers (Argote & Ingram, 2000; Connelly, Zweig, Webster & Trougakso, 2010; Cromity & Stricker, 2011; Disterer, 2001; Whitworth & Biddle, 2007)?
- Which agile practices provide mechanisms most applicable for legacy system knowledge transfer (Pawlowski & Robey, 2004; Vanhanen, Lassenius & Mantyla, 2007; Visaggio, 2001; Wasko & Faraj, 2000; Wenger & Snyder, 2000)?
- Which knowledge management systems have processes and practices that align with the processes and practices of agile software development methodologies (Abrahamsson, Salo, Ronkainen, & Warsta, 2002; Beck & Andres, 2004; Begel & Nagappan, 2007; Cockburn & Williams, 2000; García, Amescua, Sánchez & Bermón,

2011; Demarest, 1997; Kahkonen, 2004; Wenger & Snyder, 2000; Williams & Kessler, 2003)?

Audience/Significance

This study is written for IT managers who already use or who want to know more about using agile software development practices to help mitigate the risks associated with potential loss of intellectual capital related to legacy system knowledge held by a small number of subject matter experts. For this study, a legacy system is considered any asset that is difficult or risky to replace but holds high economic value to the company (Visaggio, 2001). When a company has a SME dedicated to working with a legacy system, the risk associated with the SME (Kong & Thompson, 2008; Marentette, Johnson, & Mills, 2009) adds to the fragility of the legacy system. Cromity and Stricker (2011) state that "stakeholders want to remove silos in an effort to improve workflow processes and the efficiency of decision making" (p. 167); by removing the silos, a company can "decrease redundancy and save time for those who track the same or similar information" (Cromity & Stricker, 2011, p. 168).

Delimitations

Topic focus. The context for this study focuses on *knowledge management* (Davenport & Prusak, 2000), particularly how to manage the tacit knowledge (Nonaka, 1994) that may be held by subject matter experts, and how such knowledge can be shared among the software development community within a company. C*ommunity of practice* (Wenger & Snyder, 2000) is examined as a framework with strong similarities to certain aspects of agile software development practices, with a narrow focus on *pair programming* (Williams & Kessler, 2003).

This study does not examine the complete set of agile software development practices, but rather focuses on useful practices from the two most widely-used methodologies, Scrum and

XP (Begel & Nagappan, 2007). Also, agile practices, for example test-driven development from XP (Beck & Andres, 2004) and the sprint retrospective from Scrum (Schwaber, 2009), while potentially proving useful to create explicit knowledge elements from tacit mores, are excluded from the study; the reason to limit the practices under review is twofold: (a) to control the scope, and (b) to accommodate the availability of published materials.

The study also examines three predetermined potential risks associated with the loss of intellectual capital, related to the role of the subject matter expert: (a) risks associated with legacy IT systems, (b) the risks associated with a limited point-of-contact subject matter expert, and (c) limits within small and medium companies to have adequate personnel to help keep the knowledge distributed through the development team.

Audience. The audience for this study is any IT manager, particularly in small or medium companies with limited IT development staff, who is interested in using agile practices to reduce the risks of legacy systems that are understood by solitary subject matter experts within their teams. The study will be more useful to managers with some knowledge of the agile process.

Time frame. Two different time frames are used for literature collection, depending on the type of literature needed. One time frame is for foundational research, which includes works published between 1994 and 2004. For information on Knowledge Management, Davenport and Prusak's (2000) book, *Working Knowledge: How Organizations Manage What They Know,* or Nonaka's (1994) article, *A Dynamic Theory of Organizational Knowledge Creation,* is selected to provide key definitions and frameworks. For agile software development, *The Agile Manifesto* (Beck et al., 2001) provides general background and *Extreme Programming Explained: Embrace Change* (Beck & Andres, 2004) provides foundational material for XP.

The second time frame is for works that build upon and extend the ideas provided in the foundational materials. This more recent material is limited to publications published within the past seven years which refine or realign foundational perspectives in ways useful to this study. A good example of a work that combines aspects of two different topics addressed in this study, that is knowledge transfer and agile process, is *Managing Knowledge in Development of Agile Software* (Bari & Ahamad, 2011).

Selection criteria. Literature is selected from search results performed through Google Scholar, the University of Oregon Library Portal, or IEEE Xplore. All journal article results are from juried or professional sources (Leedy & Ormrod, 2001). When selecting book-based materials, more latitude is allowed, though additional checking is made into the author's professional or academic standing via other published works.

Selection of foundational articles is based on keyword matches and the number of citing articles. Foundational articles include older writings necessary to develop the context for the study. Selection of specific topic works is based on keyword searches with preference towards works written in the last seven years.

Literature from trade magazines or informational web sites is used to gain statistics or certain definitions. In these cases, the sites are considered acceptable when they are the final word on the subject, for example, using the U.S. Small Business Administration site when defining business size (2011).

Preview of the Reading and Organizing Plan

Reading plan. The reading plan for this study is designed to examine selected references in relation to a main question and a set of supporting sub-questions. These questions pertain to the purpose of the study, and provide critical keywords and phrases to guide the examination.

References pertain to four larger content areas: (a) subject matter experts in small to medium sized organizations (Connelly, Zweig, Webster & Trougakso, 2010; Cromity & Stricker, 2011), (b) agile software development practices (Beck et al, 2001; Cockburn & Williams, 2000; Williams & Kessler, 2003), (c) knowledge management/knowledge transfer systems (Davenport & Prusak, 2000; Kong & Thomson, 2008; Nonaka, 1994), and (d) risks associated with legacy systems (Visaggio, 2001).

All references are initially qualified through the evaluation criteria described by Bell (2009), and consist of several key considerations researchers use in evaluating creditability of materials: authority, reputation, objectivity, quality, coverage, and currency. References are restricted to three different types, each of which requires a different approach to qualifying the value of the reading:

- Peer-reviewed journal articles: Read the abstract, introduction, and conclusion, followed by a scan of the list of references;
- Trade magazines or website articles: Read the introduction and the first paragraph under a sub-heading, scan the rest of the main article, review headings in side-bars, and review any references; and
- Books: Read the preface and introduction, scan the table of contents, read initial paragraphs of chapters that appear relevant, and scan chapter summaries.

Each reference undergoes a deeper reading through a process similar to content analysis (Busch, De Maret, Flynn, Kellum, Le, Meyers, Saunders & White, 2005). Conceptual analysis, as defined by Busch et al. (2005) involves a series of steps to analyze and code content by examining literature through the use of specified keywords and phrases (Busch et al, 2005). For

this study, keywords and phrases are determined in direct relation to concepts represented in each research questions and sub-question. The steps in a conceptual analysis process are:

1. Determine the depths of the analysis, for example, using single terms or phrases;
2. Determine how may concepts to evaluate;
3. Determine if the analysis is concerned with term frequency or existence;
4. Determine if keyword or phrase matching is general (forms of the word or similar concepts) or specific (exact matches only);
5. Create rules for coding that maintain consistency across materials;
6. Determine what is done with irrelevant text, especially in how such text may alter the coding;
7. Process the text through the coding using the previous six dimensions to help maintain consistency and coherency; and
8. Analyze and record your result in the annotated bibliography as indicated in the organization plan.

Organization plan. Information derived during the deep reading analysis is presented thematically (University of North Carolina, n.d.) in the Annotated Bibliography section of the document. Themes are related to the concepts embedded in the research questions; they include: (a) agile software development, providing foundational information to address the purpose of the study (Beck et al., 2001; Cockburn & Williams, 2000), (b) knowledge management (Davenport & Prusak, 2000) and knowledge transfer (Argote & Ingram, 2000), also providing foundational information to address the purpose of the study, and (c) risks related to legacy systems and subject matter experts (Boehm, 1991; Bontis, 1999; Crawford, Castro & Monfroy, 2006; Massingham, 2008), and (d) potential solutions involving aspects of knowledge management and

agile software development (Pawlowski & Robey, 2004; Vanhanen, Lassenius & Mantyla, 2007; Wasko & Faraj, 2000).

Definitions

A body of definitions is essential in any field of specialized knowledge. This study discusses software development from disciplines of agile with names such as *Scrum* or *XP* that convey no obvious meaning on their own. Knowledge management also has a specialized vocabulary including terms such as *knowledge transfer*, *tacit* and *explicit knowledge* and *subject matter experts*. This section of the document provides a set of definitions to ensure the literature is read within the specific context presented in this study.

Agile Manifesto is a series of statements made by leaders in several agile systems that define the values underlying light-weight development practices and includes: (a) "individuals and interactions over processes and tools," (b) "working software over comprehensive documentation," (c) "customer collaboration over contract negotiation," and (d) "responding to change over following a plan" (Beck et al., 2001, sect 1. para 2).

Agile Software Development is a set of principles that embody twelve concepts designed to allow self-organizing development teams deliver software early and within short time frames, while working closely with the business customer directly (Beck et al., 2001).

Community of Practice is a knowledge management system that consists of "groups of people informally bound together by shared expertise and passion for a joint enterprise" (Wenger & Snyder, 2000, p. 139).

Cross-Training is defined as "a strategy in which each team member is trained on the tasks, duties, and responsibilities of his or her fellow team members" (Volpe, Cannon-Bowers & Salas, 1996, sect 2, para 1).

Explicit Knowledge is knowledge that can be systematically communicated (Nonaka, 1994).

Intellectual Capital is defined as tangible and intangible assets held by the company in the form of explicit knowledge including documentation, procedures, correspondence, and patents, and tacit knowledge including skills, corporate knowledge, and experience (Rus & Lindvall, 2002, p. 29).

Knowledge Management (KM) is defined by Davenport and Prusak (as cited in Bjørnson & Dingsøyr, 2008) as "a method that simplifies the process of sharing, distributing, creating, capturing, and understanding a company's knowledge" (sect. 2.1, para 1).

Knowledge Network per Nonaka is knowledge held by individuals that is amplified and crystalized within an organization (1994).

Knowledge Transfer is the movement of one group or individual's experience onto another group (Argote & Ingram, 2000).

Legacy Systems are assets that have significant value but are difficult to replace. Some reasons why such systems are difficult replace include the effort to retrain users, the risks of cost or schedule overruns in replacing the system or possible loss of necessary functionality that users are expecting (Visaggio, 2001).

Lightweight software development methodology is the term used to describe agile software development methods prior to the creation of the Agile Manifesto (Constantine, 2002).

Pair programming (PP) is a practice used in agile software development in which "two persons design, code and test software together at one computer actively communicating with each other" (Vanhanen, Lassenius & Mantyla, 2007, sect 1, para 1). Further aspects of PP enable a set of developers to build trust and teamwork, exchange knowledge, and create an enhanced learning environment (Williams & Kessler, 2003).

Small and Mid-Sized Companies, in the United States, have no strict definition. Small-sized software developmentcompanies are determined by industry, in which a cap of 25 million dollars (US) is set by the U.S. Small Business Administration (2011) for "computer programming, data processing and systems design."

Software Development Method refers to a framework used "to organize and structure how software development activities should be performed, and in what order" (Scacchi, 2002, sect 2.1, para 1).

Subject Matter Expert is an "individual who exhibits the highest level of expertise in performing a specialized job, task, or skill within the organization" (iSixSigma, 2011, para 1.).

Tacit Knowledge is knowledge based in specific context, action, or experience (Nonaka, 1994).

Research Parameters

This section of the document establishes the parameters used to design the study. The study is designed as an annotated bibliography; information presented in the study is grounded in selected literature. This section has six major components:

- *Keywords,* including words and phrases used through search engines to find reference materials and why those words were selected;
- *Search Patterns,* detailing search engines used, methods of accessing information, and some delimitations placed on searches;
- *Evaluation Criteria,* detailing precisely how each located reference is reviewed and evaluated to determine if the artifact is appropriate for this study (Bell, 2009);
- *Documentation Approach,* describing how found references are tracked and stored; and
- *Reading Plan,* describing how each article is read and analyzed, in relation to the specific concepts embedded with the research questions and the larger purpose of study (Busch, De Maret, Flynn, Kellum, Le, Meyers, Saunders & White, 2005).
- *Organization Plan,* describing the thematic approach taken in the Annotated Bibliography to present the results of the deep reading analysis (University of North Carolina, n.d.).

Keywords

The keywords selected are part of the vernacular common within agile development practices and knowledge management systems. Keywords related to agile practices are found in *Extreme Programming Explained: Embrace Change* (Beck & Andres, 2004), *Working Knowledge: How Organizations Manage What They Know* (Davenport & Prusak, 2000), and

web resources including the *Agile Manifesto* (Beck et al., 2001). Some of the terms are part of this researcher's professional knowledge base, including *risk management*, *subject matter expert*, *legacy systems*, and *knowledge management*. These keywords lead to high-level subject articles including *Software Risk Management: Principles and Practices* (Boehm, 1991) found through Google Scholar. The resulting search often uncovers keyword-specific sections for the discovered articles, which are added to the list of keywords used in searching for additional source material. Another useful technique for mining articles is to find foundational studies based on keywords and then explore through Google Scholar the list of related works or articles that use the foundational study in a citation. Lookup sites including the IEEE Computer Society portal (http://www.computer.org) provides an IEEE terms (keyword) section to and also includes a section of references within the IEEE catalog which cite any source document.

The selection of the initial set of keywords based on the above methodology includes:

- Terms related to agile software development: agile, extreme programming, scrum, XP, spiral, pair programming;
- Terms related to value of knowledge held within a corporation: intellectual capital, knowledge brokering, knowledge markets;
- Terms related to the larger research problem context (siloed technology experts): expert, silo, subject matter expert, SME (also a common term meaning Small to Medium Enterprise, so less helpful), legacy systems;
- Terms related to direct means of knowledge transfer: (personal, not methodological): tutor, coach, cross-train;
- Terms related to knowledge management: knowledge management, knowledge sharing, knowledge transfer, community of practice; and

- Well known leaders in agile community: Ken Schwaber, Ward Cunningham, Alistar Cockburn, Kent Beck.

Search Patterns

Keywords are used to search initially in Google Scholar, UO Library Portal, and IEEE Computer Society. Google Scholar provides access other portals including IEEE Xplore, CiteSeer, Science Direct, and EBSCO Host. In cases where complete versions of articles are not found, the UO Library Portal serves as an alternative search engine. If content is not found directly in electronic form (PDF, raw ASCII text, or HTML pages), then the UO Interlibrary Loan (ILLiad) portal is used; if articles are not available through the ILLiad portal, then the material is dropped from consideration. Three articles are only available through ILLiad:

- Design Guidelines for Software Processes Knowledge Repository Development (García, Amescua, Sánchez, & Bermón, 2011)
- Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts, Findings and Research Methods Used (Bjørnson & Dingsøyr, 2008)
- Silo Persistence: It's not the Technology, it's the Culture! (Cromity & de Stricker, 2011)

Articles are searched and subdivided into two categories: (a) those that are deemed to be foundational and (b) those that address the specific purpose. The article “Software Risk Management: Principles and Practices” by Boehm (1991) is considered foundational; it is frequently cited in other selected literature. Foundational articles are limited to works published in the last twenty-five years. References addressing the specific purpose are limited to works published between the years 2005 and 2011. Selected references apply to software development practices in small or mid-sized companies in the United States (U.S. Small Business

Administration, 2011) that use, or wish to begin using, agile practices within their software development teams.

Evaluation Criteria

References and sources are evaluated using the criteria listed in *Critical Evaluation of Information Sources* (Bell, 2009). According to Bell, the five key considerations researchers use in evaluating credibility of materials include: (a) the author's and publisher's authority on the subject matter including credentials; reputation; and institutional associations, (b) the author's objectivity including fairness to alternative viewpoints and clearly stated biases, (c) overall quality including clarity; flow; structure; and grammar, (d) coverage in that the article extends or updates prior works or fills gaps in research, and (e) sufficiently current material (2009).

Authority. Authority is determined by the author's academic standing or the author's professional standing in the field of study based on reputation (Bell, 2009). Publication authority is based on the source found in acknowledged scholarly publications or professional journals. Some publications sources, for example the U.S. Small Business Administration (2011), are considered authoritative as representatives of regulated organizations.

Objectivity. Abstracts, introductions and conclusions are reviewed to evaluate source materials in determining publication goals and underlying biases (Bell, 2009). Subsequent examination focuses on the amount and type of referenced materials used in the material, indications of alternative points of view, and the quantity and quality of works that reference the material.

Quality. Material is reviewed to determine if the framing of information, clarity of divisions within the document, and quality of the writing is sufficient structurally. Quality also pertains to how well the research materials align with other peer viewpoints in the topic of study;

for example, does the author show reasonable understanding of the material and relevant perspectives held in the arena of the topic. Other quality checks against materials include evaluations of methodology, quality of referenced materials, and determination that the author is avoiding assumptions (Bell, 2009).

Coverage and currency. Coverage focuses on two questions that address how well the material serves the researcher: (a) does the material update other source materials, and (b) does the material support or add to the researcher's arguments (Bell, 2009)? Currency applies to using up-to-date materials, when necessary. Currency also applies towards determining if the information has later revisions or editions (Bell, 2009).

Documentation Approach

For each research topic, keywords are selected and then applied through the chosen search engines: Google Scholar, the University of Oregon Library Portal, or IEEE Xplore. Results are evaluated and reviewed as per the evaluation criteria. Potential candidate articles are recorded in EndNote X4; citation information gathering is performed through citation generation links at most of the target websites, with each citation reviewed manually and updated if incomplete; all other references are loaded manually into EndNote. Copies of the abstract or publisher description of the reference are verified as recorded or entered manually and cleaned of transcription errors.

Reading and Organizing Plan

Reading plan. This reading plan is designed to analyze selected references in four larger content areas: (a) subject matter experts in small to medium sized organizations (Connelly, Zweig, Webster & Trougakso, 2010; Cromity & Stricker, 2011), (b) agile software development practices (Beck et al., 2001; Cockburn & Williams, 2000; Williams & Kessler, 2003), (c)

knowledge management/knowledge transfer systems (Davenport & Prusak, 2000; Kong & Thomson, 2008; Nonaka, 1994), and (d) risks associated with legacy systems (Visaggio, 2001). References are initially reviewed in relation to selection criteria and credibility is determined using the key evaluation criteria described by Bell (2009) including authority and reputation of the publisher and author, objectivity of the information, quality of the writing, and the coverage and currency of the material. References used in this study are limited to written material from peer-reviewed journal articles, trade magazines or websites, or books; each source type uses a distinct approach towards reviewing the content as stated in the reading and organization plan preview.

References that meet the selection and evaluation criteria are analyzed through the process of conceptual analysis as described by Busch et al. (2005); the process consists of these eight steps:

1. Determine the depths of the analysis, for example, using single terms or phrases: phrases relevant to the concepts embedded in the research questions such as *subject matter expert* and *pair programming* are used during analysis. Some phrases require creation of subtle variations, such as *agile software development practices* versus *agile software development methods*.

2. Determine how many concepts to evaluate: There are four larger content areas as listed above; each content area is equated to concepts that are articulated as specific keywords and phrases that are used to code the reference. For example, *agile software development practices* includes keywords for *pair programming*, *Scrum*, and *XP*. At most each larger concept is restricted to five or fewer sub-concepts. Further, new keywords or phrases are added to the

keyword list as they emerge during the analysis process, when appropriate. New keywords may require already processed references to be re-evaluated.

3. Determine if the analysis is concerned with term frequency (i.e., count) or occurrence (i.e., meaning) in the literature: Each reference is evaluated to determine if any of the keywords or phrases exist in order to extrapolate meaning; frequency is not a relevant parameter. Tracking consists of denoting matched keywords or phrases; when a match is identified, the context is reviewed to better understand specific meaning.

4. Determine if keyword or phrase matching is general (forms of the word or similar concepts) or specific (exact matches only): General matching is used through the analysis with keywords added if new qualifiers are discovered. Variations of the word form are not considered new keywords unless combined with other words to create a new phrase.

5. Create rules for coding that maintain consistency across materials: Table 1 shows a sample of the translation rules for this study:

Table 1

Translation Coding Rules (sample)

Context	Word or Phrase	Alternatives
Experts	Subject matter expert	SME
Agile	Agile software development	Scrum, XP, agile development, agile practice, agile method
Agile	Pair programming	pairing
Knowledge Management	Knowledge transfer	knowledge sharing, community of practice, knowledge brokering, knowledge market

6. Determine what is done with irrelevant text, especially in how such text may alter the coding: Unrelated content is excluded from the study.

7. Process the text through the coding using the previous six dimensions to help maintain consistency and coherency: Each reference is coded through identification of, the applicable set of keywords and phrases, determined in relation to the concepts addressed in the research questions and the purpose of the study. Keyword and phrase searches are performed manually; identified keywords or phrases are recorded in the notes sections of the reference in EndNote.

8. Analyze and record your result in the annotated bibliography as indicated in the organization plan; see below.

Organization plan. The references are organized thematically in the Annotated Bibliography section of this paper, "around a topic or issue, rather than the progression of time" (University of North Carolina, n.d., para. 23). The first theme addresses the sub-questions (a) which agile practices provide ways to transfer legacy knowledge and (b) which agile methodologies align with systems of knowledge management. References provide background information to (a) acquaint the reader with agile methodologies (Abrahamsson, Salo, Ronkainen, & Warsta, 2002; Beck et al., 2001), (b) present the processes and practices used in several selected methodologies (Beck & Andres, 2004; Schwaber, 2009), and (c) report the values or benefits of the agile practices within these same selected methodologies (Cockburn & Williams, 2000; Vanhanen, Lassenius & Mantyla, 2007). References from this section provide insights into the sub-question on which agile practices provide ways to transfer legacy knowledge and the sub-question on which agile methodologies align with systems of knowledge management.

The second theme addresses the sub-questions of (a) which potential risks to intellectual capital occur when a SME is no longer able to work in their field of expertise, and (b) what barriers to knowledge management transfer exist between a SME and other developers. References provide insights into knowledge, knowledge management, and knowledge transfer

systems. Some references provide information on the basics of knowledge and knowledge management within an organization (Davenport & Prusak, 2000; Nonaka, 1994) to provide background information for the reference question on the types of knowledge that can be transferred. Several references explain the value of knowledge within a company which establishes a foundation to understand knowledge related risks, a key component to the sub-question addressing knowledge loss impacts (Bontis, 1999; Kong & Thomson, 2008). Other foundational context provides background information on knowledge transfer (Argote & Ingram, 2000; Pawlowski & Robey, 2004) including community of practice (Wasko & Faraj, 2000; Wenger & Snyder, 2000).

The third theme investigates the sub-questions on (a) barriers to knowledge transfer between a SME and other developers, and (b) the potential risks concerning loss of intellectual capital that occur when a SME is no longer working in their area of expertise. This theme defines risk within IT systems and what risk represents, then looks into specific risks related to subject matter experts (Cromity & Stricker, 2011), knowledge management (Connelly, Zweig, Webster & Trougakos, 2010; Disterer, 2001; Massingham, 2008), and legacy systems (Visaggio, 2001). The areas of risk defined here build upon information provided in the first two themes, especially knowledge management.

The final theme presents references that provide potential solutions to the main research question: Which agile software development practices offer the best potential to support knowledge transfer of legacy systems between software development SMEs and other developers? References in this section combine two or more concepts and address (a) managing software in a development environment (Bari & Ahamad, 2011; Crawford, Castro & Monfroy, 2006; García, Amescua, Sánchez & Bermón, 2011), and (b) the ways that agile development

engages community of practice (Kahkonen, 2004) and socialization (Whitworth & Biddle, 2007) in order to provide potential solutions.

Annotated Bibliography

The purpose of this annotated bibliography is to present literature that addresses the topic of how agile software development practices (Beck et al., 2001) may potentially apply to the process of knowledge transfer to relieve the risks to loss of intellectual capital posed by subject matter experts while at the same time dispersing knowledge to the wider development community. This study addresses selected practices used within agile software development, with interest in practices that involve face-to-face interactions between SMEs and other developers.

The annotated bibliography contains key references that apply to one or more of the research questions of this study. The information is presented *thematically* (University of North Carolina, n.d.) in a way that addresses the research questions:

- The first theme examines selected agile software development practice, and addresses the sub-questions on (a) the agile practices that may provide ways to transfer legacy knowledge and (b) the agile practices that may align with some knowledge management systems;
- The second theme provides information on knowledge management and transfer systems, and addresses the sub-questions on (a) potential risks to intellectual capital when the SME is no longer able to work in their field of expertise, and (b) the barriers between a SME and other developers that may inhibit knowledge transfer;
- The third theme examines the sub-questions on (a) knowledge transfer barriers, risks associated with legacy systems and (b) the loss of intellectual capital that can occur when a SME is no longer available to work in their chosen field of expertise; and

- The fourth theme provides potential solutions to the main research question on which agile software development practices best facilitate knowledge transfer between a SME and other developers for legacy systems.

Theme 1: Selected Practices Related to Agile Software Development

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods: Review and analysis*. Espoo [Finland]: VTT. doi: 10.1.1.161.5931

Abstract. The aim of the article is to systematically review the existing literature on agile software development methodologies with three main purposes. First, the authors propose a definition and a classification of agile software development approaches. Second, they analyze ten software development methods that can be characterized as being "agile" against the defined criteria. Third, the authors compare the agile software development methods and highlight their similarities and differences within their defined criteria.

Credibility. The publisher, VTT, is a not-for-profit research firm located in Finland. It produces approximately 2000 publications, most of which appear in peer-reviewed journals or conference proceedings (VTT, n.d.). Abrahamsson is a professor at Free University of Bozen-Bolzano, and adjunct professor at the University of Oulu. Salo is a researcher with a PhD in Pharmacy from the University of Kuopio, but has co-written a number of well-referenced articles on agile software development. The content is clearly laid out, with ten different agile methodologies analyzed through identical sub-topics. The material is cleanly written using proper grammar and structure.

Summary. This article provides a ground-up approach to understanding agile software development methods. It begins by describing the overarching themes to agile methods and then covers each of them methodically with sections on “process, roles and responsibilities, practices, adoption and experiences, scope of use and current research” (p. 18). After describing each of the ten methodologies, a table compares methods based on general key points, features, and drawbacks of each methodology. The conclusion of

the article shows that agile software development methods all derive from elements of the Agile Manifesto (Beck et al., 2001) but differ in focus.

This document provides relevant background information on some agile software development methods as these are described in this study. High-level definitions for methodologies such as Scrum and XP are derived from the document content.

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change*. Boston, MA: Addison-Wesley.

Abstract. This book offers the starting point of understanding of the Extreme Programming (XP) agile software development methodology. This methodology is considered by the authors as a lightweight approach that is not for everyone, but it offers a new approach into developing software. The book discards older software development norms, and is (at the time of writing) considered to be potentially controversial.

Credibility. Andres has a BS degree psychology with a focus on organization behavior and decision analysis. (InformIT, n.d.) She worked with Beck on early social aspects of XP. Beck has a MS in computer science and one of the cosignatories of the Agile Manifesto. Beck has published books and articles on the subjects of XP, test driven development, and patterns. This book is written in first person, and incorporates frequent anecdotes and metaphors. There are some biases within the material towards using XP, but there is a chapter explaining when it would be inappropriate to use XP in an organization. There is an annotated bibliography that is well organized by subject; some references are more whimsical (i.e. The Princess Bride motion picture) and some discuss interesting point-of-views from non-IT business (i.e. Disney Animation: The Illusion of

Life describing team structure and change). The book is designed for a professional IT worker or manager to read.

Summary. This book provides detailed coverage of the Extreme Programming (XP) methodology as described in this study. The writing is designed for easy readability with frequent metaphors and anecdotes while describing the critical aspects to this development process. There are some references related to collective code or system ownership that enable knowledge transfer within the developer community; it also addresses how collective ownership helps defer the risk of knowledge loss (i.e., intellectual capital). This book further establishes pair programming as an important agile practice and part of the XP methodology and defines how it is used in a XP setting.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Principles behind the agile manifesto. Retrieved November 6, 2011, from http://agilemanifesto.org/principles.html

Abstract. The Agile Software Development Manifesto is the result of a two-day conference between leading members of various lightweight software development methodologies such as XP, Scrum, and Crystal. The final statement is frequently referred to in literature regarding agile software development methods (cited in over 300 works per Google Scholar).

Credibility. The materials are the output from a two-day conference between leaders of many agile software development methodologies. These leaders are frequently published authors in one or more fields of IT and regular speakers at IT-related conferences and include: (a) Mike Beedle, co-author of *Scrum Agile Software Development* with Ken Schwaber; (b) Arie van Bennekum, an early adopter and proponent for the DSDM

(Dynamic Systems Development Method) agile methodology; (c) Alistair Cockburn, creator of the Crystal agile software development method; (d) Ward Cunningham, co-contributor to the XP methodology and creator of Framework for Integrated Testing (FIT) and the wiki system concepts; (e) Martin Fowler, several IT professional concept oriented books such as *Analysis Patterns*, *Refactoring*, and *Planning Extreme Programming*; (f) Jim Highsmith, creator of the Adaptive Software Development method; (g) Ron Jefferies, the first Extreme Programming coach and co-author of *Extreme Programming Installed*; (h) Robert Martin, author of *Agile Software Development: Principles, Patterns, and Practices*; (i) Ken Schwaber, co-creator of Scrum and co-author of *Scrum Agile Software Development* with Beedle; (j) Jeff Sutherland, co-creator of Scrum; and a number of other leaders in early forms of agile practices.

Summary. The Agile Manifesto is the synthesis of multiple perspectives held among the early "lightweight" software development methodology leaders who found common ground between their process approaches over a two-day conference. The Agile Software Development Manifesto sets the underlying concepts used by all agile software development methodologies. This document provides a simple and direct means to understand why agile is important and what preferences agile methods maintain. One principle, face-to-face communication within the team and with the stakeholders, conveys the importance of knowledge transfer within agile practices as the notion is addressed in this study.

Begel, A., & Nagappan, N. (2007, 20-21 Sept. 2007). *Usage and perceptions of agile software development in an industrial context: An exploratory study.* Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, 2007.

Retrieved from http://research.microsoft.com/en-us/um/redmond/groups/hip/papers/agiledevatms.pdf

Abstract. This paper reports the results of an empirical study into the use and perceptions of agile software development at Microsoft. The paper showed that one-third of the respondents use agile methodologies to some extent in their work, usually a form of Scrum. The paper also reports favored practices of improved communication, quicker releases, and increased development flexibility, but respondents are concerned about too many meetings, agile/non-agile team coordination, and scaling to larger projects.

Credibility. Begel has a PhD in computer science from the University of California, Berkeley and works in Human Interactions in Programming Group at Microsoft (Microsoft Research, n.d.). Nagappan has a PhD from the North Carolina State University and works in the Empirical Software Engineering Research Group at Microsoft (Microsoft Research, n.d.). The paper is part of a conference proceeding for the First International Symposium on Empirical Software Engineering and Measurement in 2007. The paper shows no bias towards any particular agile practice or methodology in the report. Data is collected using anonymous web-based forms; all participants were developers, managers, or managers of managers.

Summary. The paper provides real-world examples of agile practices and methodologies used within the setting of a large corporation, Microsoft. Empirical data gathered anonymously provides information related to this annotated bibliography as follows: (a) teams use agile practices for both new and legacy development, (b) pair programming is the least-used and least-likely-to-be-used practice, (c) the use of agile practices at

Microsoft is growing at a continuing rate, and (d) the top benefit to use of agile methodologies is improved communication.

Cockburn, A., & Williams, L. (2000). *The costs and benefits of pair programming.* Paper presented at the eXtreme Programming and Flexible Processes in Software Engineering -- XP2000, Cagliari, Sardinia, Italy. doi: 10.1.1.26.9064

Abstract. Pair or collaborative programming is where two programmers develop software side by side at one computer. Using interviews and controlled experiments, the authors investigated the costs and benefits of pair programming. They found that for a development-time cost of about 15%, pair programming improves design quality, reduces defects, reduces staffing risk, enhances technical skills, improves team communications and is considered more enjoyable at statistically significant levels.

Credibility. Cockburn received his PhD from the University of Oslo, is a recognized expert in use cases, and creator of the agile methodology Crystal. Williams received her PhD in computer science from the University of Utah and a co-founder of the Agile 2000 conference. The paper builds arguments in favor of pair programming through eight clearly described factors before concluding. The language is informal at times, the intent of which is to convince the audience of the merits of pair programming, but it is clearly written and well referenced. The paper evolved into a chapter in the book *Extreme Programming Examined*, published by Addison Wesley in 2001.

Summary. This paper provides insight as to why a manager or developer would wish to use the agile practice of pair programming in their development environment. Of the eight investigative paths explored in the paper, the following pair programming benefits apply directly to this annotated bibliography: (a) continuous code reviews, (b) pair

members learning from each other, (c) more effective team communication, and (d) the reduced risk associated with staff loss. The paper addresses two key concepts embedded in the research questions in this study: (a) tying the agile practice of pair programming to knowledge transfer, and (b) reducing the risk associated with the loss of a staff member. It also provides a definition to the pair programming practice.

Schwaber, K. (2009). *Agile project management with scrum.* [Adobe Digital Editions version]. Retrieved from http://multco.eblib.com/patron/FullRecord.aspx?p=488725

Abstract. Scrum co-creator Ken Schwaber identifies the real-world lessons, both successes and failures, culled from his years of experience coaching companies in agile project management. This book helps the reader understand how to use Scrum to solve complex problems and drive better results by delivering more valuable software faster.

Credibility. Schwaber co-created the Scrum agile process with Jeff Sutherland in the early 1990s and is one of the original signatories on the Agile Manifesto. The book is published by Microsoft Press. The chapters lay out basic Scrum process, roles, artifacts, and flow; these chapters provide case studies to give examples to the reader. The book contains bias towards using the Scrum methodology, but also contains examples of success and failures within the methodology including sections indicating when the method would not be applicable. The book is not peer-reviewed.

Summary. This book provides a high-level coverage of the overall Scrum process, designed to educate people new to Scrum. Chapters extensively use case studies to illuminate the author's points and include elements of success and failure. The information applies to this annotated bibliography in that it provides significant insight into Scrum; understanding Scrum helps address the research questions on (a) which agile

practices can be used for legacy knowledge transfer, and (b) which practices best align with knowledge transfer systems.

Vanhanen, J., Lassenius, C., & Mantyla, M. (2007). *Issues and tactics when adopting pair programming: A longitudinal case study.* Paper presented at the International Conference of Software Engineering Advances. doi: 10.1109/ICSEA.2007.48

Abstract. This paper presents experiences from a two-year study of adopting pair programming (PP). The paper examines the five tactics used by the company in adopting PP (a) the creation of simple PP guidelines, (b) the use of a PP champion, (c) making the use of PP voluntary, (d) creating a positive atmosphere for PP, and (e) instituting a separate PP room. The resultant surveys report that PP considerably surpassed developers' preconceptions of PP, but also uncovered issues with the selected company's infrastructure, that is a lack of adequate workspace to properly support PP. At the end of the study, a majority of the developers thought that PP should be utilized for more than the approximate 10% of development effort.

Credibility. Vanhanen has a D.Sc. and is a researcher and instructor at the Department of Computer Science and Engineering at Aalto University, Finland. Lassenius has a DSc, is a professor (pro tem) at Aalto University, and is a member of the university's Software Process Research Group. Mantyla also has his DSc from Aalto University and works in the same research group as Lassenius. The paper is based on the use of four surveys, initially closed questions, but with open questions added later, over a two-year period to investigate the implementation of PP at a Finish company. Sections show related works, research methodology, results with analysis, and discussion on the study including

lessons learned. Language is formal and well referenced throughout. The study does not exhibit any bias for or against PP.

Summary. This paper provides supplemental information on starting pair programming (PP) within a company; five distinct tactics are described to aid implementing the practice. The paper describes some initial barriers to PP within the company surveyed and notes improvements to acceptance of PP when the barriers are removed. The paper also discusses limited aspects of knowledge transfer as an effect of using PP, though mostly as a side-point. The research uses a qualitative approach to assess the effectiveness of the practice over time.

Williams, L., & Kessler, R. R. (2003). *Pair programming illuminated*: Addison-Wesley Professional.

Abstract. In the agile software development practice of pair programming, two programmers work side-by-side at one computer, continuously collaborating on the same design, algorithm, code, and test. It produces a higher quality of code in about half the time than that produced by the summation of their solitary efforts. The practice is not without it difficulties, as people and personalities are involved, particularly people who are accustomed to working alone. Pair programming provides numerous benefits but requires careful thought and planning.

Credibility. Williams has a PhD in computer science from the University of Utah and currently serves as a Professor in the Computer Science Department of the College of Engineering at North Carolina State University. Kessler has a PhD in computer science from the University of Utah; he serves as the Associate Director and a Professor for the university's Computer department. The book is written for the IT professional with

leanings towards newcomers to the practice of pair programming (PP) and includes two chapters dedicated to case studies of PP implementation. The book is cited by over 300 articles, per Google Scholar.

Summary. This book codifies the agile software development practice of *pair programming* (PP). The authors provide arguments and guidelines for PP from the perspectives of management, developers, and educators. There are significant discussions on some specific agile methodologies including Scrum and Test Driven Development to limited degrees and more significantly into the subject of XP. Research questions addressed by this study include: (a) questions related to knowledge transfer via constant learning between pairs in topics of system, styles, and approaches; (b) the specific knowledge transfer systems known as *community of practice*, a positive side effect of pair programming caused by ad hoc community formations via regularly rotating pairs; (c) handling of experts in pairing situations and knowing when pairing should be regulated so as to not impinge on the experts more critical work; and (d) understanding how PP aligns with knowledge management to achieve best practices in both domains.

Theme 2: Knowledge Management, Knowledge Transfer, Community of Practice, and Intellectual Capital

Alavi, M., & Leidner, D. (1999). *Knowledge management systems: Emerging views and practices from the field.* Paper presented at the Thirty-second Annual Hawaii International Conference on System Sciences, Hawaii. doi: 10.1109/HICSS.1999.772754

Abstract. This study provides an analysis of practices and outcomes of knowledge management systems (KMS) and the nature of KMS as they are evolving in fifty organizations. The findings suggest that interest in KMS across a variety of industries is very high, the technological foundations are varied, and the major concerns revolve around achieving the correct amount and type of accurate knowledge and garnering support for contributing to the KMS.

Credibility. Alavi is the Vice Dean for Faculty and Research and a professor of information strategy at Emory University. Leidner is a professor of information systems at Baylor University. The conference proceeding was in the 32nd year at the time of the writing; it is currently in the 45th year. The survey encompasses twelve countries with responses from a balanced variety of IT managers representing different industries including finance, manufacturing, government, energy, and other sectors. The paper conforms to standard headers and sections including introduction, definition/description, methodology, findings, discussion and conclusion, and references. The material is clear and does not require special knowledge to understand the content.

Summary. Knowledge management systems (KMS) establish a means with which to convert tacit knowledge into explicit knowledge. This study looks into how 50 different companies worldwide understood and approached KMS. It also discusses barriers to

knowledge transfer, though limited to only IT management concerns. The outcome of this study helps define a KMS as the concept is used in this study, uncovers concerns by managers in implementing KMS systems and strategies, and signifies the importance of technology to implement any system. The information within this study is presented to help build an understanding for managers who do not already implement a KMS and also to help those managers understand some of the expected barriers to a knowledge management system and knowledge transfer.

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, *25*(1), 107-136.

Abstract. There is growing interest in treating knowledge as a significant organizational resource, which IT needs to codify into a knowledge management system (KMS). The objective of KMS is to support creation, transfer, and application of knowledge in organizations. Knowledge and knowledge management are complex and multi-faceted concepts. This article provides a review and interpretation of knowledge management literature in different fields with an eye toward identifying the important areas for research. The article presents a detailed process view of organizational knowledge management with a focus on the potential role of information technology in this process. Drawing upon the selected literature and the analysis of knowledge management processes, the article discusses several important research issues surrounding the knowledge management processes and the role of IT in support of these processes.

Credibility. Alavi is the Vice Dean for Faculty and Research and a professor of information strategy at Emory University. Leidner is a professor of information systems

at Baylor University. The article is published in MIS Quarterly, a peer-reviewed journal. The sections of the article are clear with divisions for introduction, literature reviews, and conclusion. There are several tables and graphics to aid in understanding.

Summary. This article includes three primary sections, (a) introduction, (b) literature reviews, and (c) summary. The section on literature reviews covers three distinct aspects on knowledge management: (a) knowledge and the firm, in which alternative views of knowledge are examined; (b) the possible role technology plays in knowledge management systems; and (c) research themes requiring further study. The review provides detail as presented in this study in the Introduction section of the paper to help understand the main aspects of knowledge management. The section on alternative views of knowledge presents different taxonomies used to represent knowledge and knowledge management, providing key data for understanding these systems. This article also introduces the KMS of *community of practice*, which has ties to research questions addressed in this study related to both agile processes and knowledge management.

Argote, L., & Ingram, P. (2000). Knowledge transfer: A basis for competitive advantage in firms. *Organizational Behavior and Human Decision Processes, 82*(1), 150-169. doi: 10.1006/obhd.2000.2893

Abstract. The article examines the foundations of knowledge transfer in organizations and argues that the creation and transfer of knowledge are a basis for competitive advantage in firms. The article posits that by embedding knowledge in interactions involving people, organizations can both effect knowledge transfer internally and impede knowledge transfer externally and thereby gain a competitive advantage over other firms.

Credibility. Argote is the Director of the Center of Organizational Learning, Innovation and Performance and a professor of organization behavior and theory at the Carnegie Mellon Tepper School of Business. Ingram has a PhD from Cornell University and serves as a professor of business at the Columbia Business School. The article is well-referenced with 25% of the content being references. The article is well written and clear; the reader does not require special knowledge to understand the contents of this article.

Summary. The article begins with a definition of knowledge transfer as a way to frame the context, and then proceeds to define means with which to transfer knowledge; these definitions are used in this study. This article provides some background on types of knowledge held within companies. The article offers information into the way that knowledge is stored within a corporation including within individuals, knowledge tools, and tasks. The article does not mention *intellectual capital* directly, though the section on knowledge as a competitive advantage holds similar ideas as Bontis (1999). Two research questions in this study are explored within information provided in this reference: (a) the question on barriers to knowledge transfer, which ties directly to the section on *Moving Knowledge by Moving Reservoirs and Networks*, and (b) the question related to agile practices and how they may work with knowledge transfer mechanism, especially in the sections modifying networks and reservoirs and the factors affecting knowledge transfer. Ideas related to the second question demonstrate the importance of groups and group communication, similar to agile pair programming and Scrum daily stand-up meetings.

Bjørnson, F. O., & Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology, 50*(11), 1055-1068. doi: 10.1016/j.infsof.2008.03.006

Abstract. This article provides a systematic review that identifies empirical studies of knowledge management initiatives in software engineering, and discusses the concepts studied, the major findings, and the research methods used. Within software engineering related industries, the article identifies 29 empirical studies and 39 reports of lessons learned. The majority of empirical studies relate to technocratic and behavioural aspects of knowledge management, while there are few studies relating to economic, spatial and cartographic approaches.

Credibility. Bjørnson has a PhD from the Norwegian University of Science and Technology, where he defended his thesis titled *Knowledge Management in Software Process Improvement*; he has co-authored over 17 scientific papers over the last 8 years. Dingsøyr is an adjunct associate professor at Norwegian University of Science and Technology. The article is a review of empirical studies of literature on the subject of knowledge management in a software engineering context. The article defines background information, the method of research including research parameters and qualifiers, results, discussion, and a conclusion. There are 119 citations noted in the article.

Summary. This article provides a review of literature that examines two different schools of thought on knowledge management (KM), (a) technocratic and (b) behavioral, to investigate what concepts, findings, and research exists on KM in a software engineering setting. The article assists in understanding the body of work on software development knowledge management that organizations are taking. One section ties to the research question as to which KM systems tie to agile practices in the discussion on

agile software development methods as they relate to tacit knowledge. There are also relevant discussions on KM systems such as *community of practice*.

Bontis, N. (1999). Managing organisational knowledge by diagnosing intellectual capital: Framing and advancing the state of the field. *International Journal of Technology Management, 18*(5), 433. Retrieved from http://www.business.mcmaster.ca/mktg/nbontis/ic/publications/ijtmbontis.pdf

Abstract. Organizational knowledge is an important aspect of strategic competitive advantage for organization that can be measured in an organization's intellectual capital. Intellectual capital is conceptualized from a number of disciplines making the field a mosaic of perspectives. The article presents a literature review from a variety of managerial disciplines.

Credibility. Bontis has a PhD from the Ivey Business School of University of Western Ontario in the study of strategic management. The *International Journal of Technology Management* is a peer-reviewed, scholarly journal. The structure of the article provides an introduction, a section that establishes the premise, the literature review, several sections with specific topical discussion elements, the research to date, and a conclusion. The article includes 188 references. The article relies on the reader's knowledge of the general field related to the subject, including some of the referenced works.

Summary. The authors discuss the importance of intellectual capital, why it is not a fad, and some of the difficulties in measuring it. While this article does not expressly address the risks related to the loss of intellectual capital, it does help the reader understand the intangible value associated with specific corporate knowledge. The article also discusses

the value to group knowledge, an indirect tie to the knowledge management system *community of practice*.

Carlile, P. R., & Rebentisch, E. S. (2003). Into the black box: The knowledge transformation cycle. *Management Science, 49*(9), 1180-1195. Retrieved from http://www.jstor.org/stable/4134034

Abstract. The study examines how knowledge is integrated in complex technology and product development settings. By framing the task of knowledge integration as a cycle, the study highlights the inability of some knowledge transfer theories in explaining the consequences that arise from the path-dependent nature of knowledge. Through empirical evidence, the study outlines three stages of a knowledge transformation cycle to address limitations of knowledge integration within complex technology.

Credibility. Carlile has a PhD from the University of Michigan in organization studies and has published a number of articles on knowledge management. Rebentisch has a PhD from Massachusetts Institute of Technology (MIT) in the management of technological innovation and is a researcher at the Lean Advancement Initiative at MIT. The study was published in Management Science, a peer-reviewed management journal. The study is well referenced and broken out into an introduction, definition, research parameters, findings, discussion, and conclusion. The language used in the article relies on the reader having some background familiarity with the subject of knowledge management.

Summary. This study examines the difficulty of knowledge transfer within complex tasks based on three stages in the knowledge transformation cycle: (a) how the novelty of knowledge requirements limits knowledge reuse, (b) how mitigating between the uses of knowledge requires transformation of knowledge by those using or sharing the

knowledge, and (c) how representing and sharing the new knowledge is difficult, especially between specialty domains. For the purposes of this annotated bibliography, this study helps point out potential barriers to knowledge transfer between subject matter experts and other knowledge users. This study does not provide insights to address the problems, but highlights the types of difficulties one might expect and related deficiencies in current knowledge transfer systems.

Davenport, T. H., & Prusak, L. (2000). *Working knowledge: How organizations manage what they know*: Harvard Business Press.

Abstract. The book is a significant primer on knowledge management, establishing the language and core concepts used by knowledge management systems. Through the presentation of over 30 case studies with knowledge-rich firms, the authors examine how all types of companies can effectively understand, analyze, measure, and manage their intellectual assets, turning corporate knowledge into market value.

Credibility. Davenport has a PhD in sociology from Harvard University and currently holds the President's Chair in Information Technology and Management at Babson College. Prusak has an honorary PhD in information science from Long Island University and is the founder of IBM's Institute of Knowledge Management. The book is cited over 10,000 times, per Google Scholar. The material contains well-documented examples of how over 30 knowledge-rich organizations manage knowledge. The book is written for managers of knowledge workers.

Summary. This book examines knowledge within companies through numerous case studies. The book helps managers understand the best ways to use knowledge within a company, and includes practical advice as to how to approach building a corporate

knowledge management system from a general perspective. Sections on knowledge markets and knowledge transfer are applicable to the research questions posed in this study on knowledge transfer barriers and similarities between knowledge management systems and agile practices (from the KM side of the equation). Many chapters in this book provide useful insights into relevant knowledge management aspects.

Demarest, M. (1997). Understanding knowledge management. *Long Range Planning, 30*(3), 374-384. doi: 10.1016/s0024-6301(97)90250-8

Abstract. This article discusses the rise of knowledge management as a discipline, defining the relationship between knowledge management and traditional measures of firm performance such as marketplace innovation, internal efficiency, and profitability. The article describes some basic models for understanding how knowledge is created, embodied, and distributed within organizations; it also traces the connection between knowledge management and the infrastructure that supports it, particularly, new information technologies.

Credibility. Demarest has a MA in English from the University of South Carolina-Columbia and an Executive MBA from Stanford University. The journal *Long Range Planning* is a peer-reviewed publication focused on knowledge in the subject of strategy. The article defines several key concepts followed by suggestions for how to foster knowledge management within the corporation. The article is written in a style more in line with a popular journal than a research journal, utilizing a sizable number of bullet-points and numbered lists. The language of the article is clear and easy to understand to a management professional. The article cites references from other knowledge leaders including Polanyi and Nonaka.

Summary. This article helps with understanding the importance of knowledge and knowledge management within a corporation. There is some information related to risks associated with knowledge workers, especially subject matter experts, from a strategic management perspective; these risks include terms of employment and the possibility that a knowledge worker may possibly switch employers to a competitor. Both risks lend credence to the significance on the research question in this annotated bibliography related to the loss of intellectual capital when an expert is unable to work in their knowledge domain.

Ford, D. P., & Staples, D. S. (2008). What is knowledge sharing from the informer's perspective? *International Journal of Knowledge Management, 4*(4), 1.

Abstract. In this study, the authors examine how other researchers operationalize knowledge sharing, and they conduct a qualitative study to further understand this construct. The study develops a knowledge sharing and hoarding classification system in which six knowledge sharing behavioral categories are identified; recommendations for future knowledge management research are suggested.

Credibility. Ford has a PhD in Management information systems and organizational behavior from Queen's University, Ontario, Canada. Staples has a PhD in business administration from the Western Business School at the University of Western Ontario. This is a qualitative study with sections introducing the subject, examining the research methods, showing the results, discussing the results, and concluding. There are over 70 references from scholarly sources. The material is well written and clear to understand.

Summary. This study takes a qualitative approach to create recommended measures by which knowledge sharing can be discussed as a behavior for further research. For the

purposes of the annotated bibliography, this study aids in understanding both important aspects to knowledge transfer (as a means of knowledge sharing) as well as barriers to knowledge transfer (knowledge hoarding or hiding). The study identifies six distinct knowledge sharing behaviors and creates a descriptive text for each for future research.

Kong, E., & Thomson, S. B. (2008). An intellectual capital perspective of human resource strategies and practices. *Knowledge Management Research & Practice, 7*(4), 356-364. doi: 10.1057/kmrp.2009.27

Abstract. The authors propose a holistic overview of connections between three concepts: (a) intellectual capital (IC), (b) human resource management (HRM), and (c) strategic hum resource management (SHRM). The paper argues that not only are the three concepts closely connected, but also IC should play a key role in SHRM processes and HRM practices in organizations. The strategic connections also demonstrate that IC can be conceptualized as a holistic partner to both SHRM and HRM; thus, adding strong support for the need to measure IC accurately. The paper proposes a theoretical framework to illustrate IC, SHRM and HRM connections.

Credibility. Kong has a PhD in strategic management at Monash University, Australia, and is senior lecturer at the School of Management & Marketing, University of Southern Queensland, Australia. Thomson has a PhD from St. George's University, Grenada, West Indies, where he serves as an assistant professor for the Department of Business. The journal *Knowledge Management Research & Practice* is a peer-reviewed publication. The article has relevant definitions in the introduction followed by several discussion sections, a section for future research, and a conclusion. The article is well referenced, with approximately two of the nine pages being references.

Summary. The article divides intellectual capital (IC) into three components: (a) human capital, (b) social or relational capital, and (c) organizational or structural capital. From the human capital component, this article improve the reader's understanding of the desirability to maintain, through development and better deployment, experts instead of attempting to bring them into the organization. That point particularly addresses the research question on the potential risks to loss of IC. The second component, social capital, also works into the equation by showing how internal networks help spread knowledge by setting up both content and a similar context. The third component, organization capital, can establish a culture for learning as a strategy using *community of practice*, a knowledge transfer system.

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science, 5*(1), 14-37. Retrieved from http://www.jstor.org/stable/2635068

Abstract. This paper proposes a paradigm for managing the dynamic aspects of organizational knowledge creating processes. Its central theme is that organizational knowledge is created through a continuous dialogue between tacit and explicit knowledge. The nature of this dialogue is examined and four patterns of interaction involving tacit and explicit knowledge are identified. It is argued that while new knowledge is developed by individuals, organizations play a critical role in articulating and amplifying that knowledge. A theoretical framework is developed which provides an analytical perspective on the constituent dimensions of knowledge creation. This framework is then applied in two operational models for facilitating the dynamic creation of appropriate organizational knowledge.

Credibility. Nonaka has a PhD in business administration from the University of California, Berkeley, and is a Professor Emeritus at the Hitotsubashi University Graduate School of International Corporate Strategy, Japan. He has co-authored 20 books and approximate 40 articles in the field or organizational and strategic management. *Organization Science* is a peer-reviewed publication. The article is referenced by over 9,500 articles per Google Scholar. The article is well referenced and written to clearly explain the theoretical elements in a logical and consistent manner.

Summary. This article codifies the concepts necessary to understand organizational knowledge work through (a) creation, (b) growth of knowledge within the individual, creation of a community (self-organizing team) for knowledge creation, (c) sharing knowledge within the team and externally, and (d) the crystallization of the knowledge from the team in the form of a product or system. This article presents contextual information relevant to the larger set of research questions in this annotated bibliography related to knowledge.

Pawlowski, S. D., & Robey, D. (2004). Bridging user organizations: Knowledge brokering and the work of information technology professionals. *MIS Quarterly*, *28*(4), 645-672. Retrieved from http://www.jstor.org/stable/25148658

Abstract. This interpretive case study examines knowledge brokering as an aspect of the work of information technology professionals. The purpose of this exploratory study is to understand knowledge brokering from the perspective of IT professionals as they reflect upon their work practice. As knowledge brokers, IT professionals see themselves as facilitating the flow of knowledge about both IT and business practices across the boundaries that separate work units within organizations. A qualitative analysis of

interviews conducted with 23 IT professionals and business users in a large manufacturing and distribution company is summarized in a conceptual framework showing the conditions, practices, and consequences of knowledge brokering by IT professionals. The framework suggests that brokering practices are conditioned by structural conditions, including decentralization and a federated IT management organization, and by technical conditions, specifically shared IT systems that serve as boundary objects. Brokering practices include gaining permission to cross organizational boundaries, surfacing and challenging assumptions made by IT users, translation and interpretation, and relinquishing ownership of knowledge. Consequences of brokering are the transfer of both business and IT knowledge across units in the organization.

Credibility. Pawlowski has a PhD in computer information systems from Georgia State University and is currently an Associate Professor at Louisiana State University's Department of Information Systems and Decision Sciences. Robey has a doctorate in Administrative Science from Kent State University. This is a scholarly paper with an introduction, a list of related research, a description of the research methodology, the results of the research, a discussion, evaluation, and conclusion. Each section clearly entails relevant information in a clear manner with appropriate sub-headings. There are approximately 80 references, seeded throughout the document.

Summary. This article researches the role that IT plays in knowledge brokering as a facilitator in knowledge transfer throughout a corporation. This brokering is achieved through systems shared by different business units. Knowledge brokering, a form of knowledge transfer, has four specific practices: (a) crossing boundaries, (b) surfacing and challenging assumptions, (c) translation and interpretation, and (d) relinquishing

ownership of the knowledge. These practices apply internally as well as externally and may be applied to the research question in this study on legacy system knowledge transfer.

Wasko, M., & Faraj, S. (2000). "It is what one does": Why people participate and help others in electronic communities of practice. *The Journal of Strategic Information Systems*, *9*(2-3), 155-173. doi: 10.1016/s0963-8687(00)00045-7

Abstract. Organizations are finding that their members are often reluctant to exchange knowledge with others in the organization. This paper examines why. The article finds that organizations are treating knowledge as a private good, owned either by the organization or by organization members. The article proposes that knowledge can also be considered a public good, owned and maintained by a community. When knowledge is considered a public good, knowledge exchange is motivated by moral obligation and community interest rather than by narrow self-interest.

Credibility. Wasko has her PhD in management information systems from the University of Maryland, College Park, and is chair of the Department of Management, Information Systems and Quantitative Methods at the University of Alabama. Faraj has his doctorate in management information systems from Boston University. The publication is a peer-reviewed scholarly journal. This well referenced document provides introductory information, theoretical perspectives of knowledge (definitions and groundwork), the research direction (motivations for knowledge exchange), methods, results, discussion, and conclusion.

Summary. This article examines knowledge as being either a private good or a public good. Private good knowledge, that is knowledge held by individuals or embedded in

objects, typically can only be exchanged for something of equal or greater value as determined by the holder of the knowledge. Public good knowledge, that is knowledge held by community, is maintained instead by other factors such as a public duty or a desire to help care for the community. Several portions of the article focus on the knowledge transfer system called *community of practice*, a key aspect of research questions on knowledge transfer in this study, as well as insights into some barriers to such transfers (e.g. bad blood in the community or sub-standard exchange goods for knowledge).

Wenger, E. C., & Snyder, W. M. (2000). Communities of practice: The organizational frontier. *Harvard Business Review, 78*(1), 139-145. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=2628915&site=ehost-live&scope=site

Abstract. This article examines the knowledge management system known as *community of practice.* A community of practice is a group of people informally bound together by shared expertise and passion for a joint enterprise. People in companies form them for a variety of reasons--to maintain connections with peers when the company reorganizes; to respond to external changes such as the rise of e-commerce; or to meet new challenges when the company changes strategy. Members of a community of practice inevitably share knowledge in free-flowing, creative ways that foster new approaches to problems. Communities of practice can drive strategy, generate new lines of business, solve problems, promote the spread of best practices, develop people's skills, and help companies recruit and retain talent. The authors explain the steps managers need to take

in order to get communities going and to sustain them so they can become a central part of their companies' success.

Credibility. Wenger has a PhD in artificial intelligence and is a leading expert on the subject of *community of practice*. Snyder has a PhD in business administration from the University of Southern California. The article is published in the Harvard Business Review, a professional journal that is not peer-reviewed. The contents are clear and easy to follow with frequent examples of *community of practice* implementations in a variety of settings. There are no references listed in this article.

Summary. This article provides a variety of accounts where different business groups engaged in *communities of practice*. This article defines a *community of practice* as a type of knowledge management system designed for knowledge transfer, and provides examples as ways to cultivate such a community. The information found in this article provides some support for research questions in this study involving knowledge transfer and bear some resemblance to structures existing in agile software development methods such as Scrum.

Theme 3: Problems Caused by SME Loss, Barriers to Knowledge Transfer, and Risks Associated with Legacy Systems

Boehm, B. W. (1991). Software risk management: Principles and practices, *8*, 32-41. Retrieved from http://doi.ieeecomputersociety.org/10.1109/52.62930

Abstract. The emerging discipline of software risk management is described. It is defined as an attempt to formalize the risk-oriented correlates of success into a readily applicable set of principles and practices. The paper's objectives are to identify, address, and eliminate risk items before they become either threats to successful software operation or major sources of software rework. The basic concepts are set forth, and the major steps and techniques involved in software risk management are explained. Suggestions for implementing risk management are also provided.

Credibility. Boehm has PhD in mathematics from the University of California, Los Angeles, and an honorary ScD in computer science from the University of Massachusetts; he has worked as the Director for DARPA and has served on the boards of a number of scientific journals including IEEE Computer and IEEE Software. The article is written for computer science professionals. There are only five references listed, two of which are Boehm. The article uses a mathematical approach to mitigating risk as well as understanding risk analysis.

Summary. This article explores the concepts of risk and risk management, which provide background information into the research question on the risk of the loss of a subject matter expert by an organization. A correlation is made between successful software development work (in the form of projects) and good risk management techniques employed by the project manager. The article provides information for agile software

development managers about how to assess (identification, analysis, prioritization) and control risks (management, resolution, monitoring) during their project work.

Connelly, C. E., Zweig, D., Webster, J., & Trougakos, J. P. (2010). Knowledge hiding in organizations. *Journal of Organizational Behavior*, doi: 10.1002/job.737

Abstract. This article presents the results of three studies that together: (a) establish that knowledge hiding occurs in organizations, (b) develop a multidimensional measure to assess knowledge hiding and to distinguish this behavior from related constructs, and (c) focus on distrust as a key predictor of knowledge hiding in organizations. The goal of the article is to present a theory of knowledge hiding, which may deepen understanding of how knowledge is transferred within organizations, uncover the barriers to effective knowledge transfer, and provide a basis for future research.

Credibility. Connelly has a PhD from Queen's University and is an Associate Professor of Organizational Behavior and Human Resources Management at the DeGroote School of Business at McMaster University. Zweig has a PhD in Industrial/Organizational Psychology at the University of Waterloo and serves as an Associate Professor of Organizational Behavior and Human Resources Management in the Department of Management at the University of Toronto. Webster has a PhD from New York University and serves as a Professor of MIS in the School of Business at Queen's University. Trougakos has a PhD from Purdue University and is an Assistant Professor of Management in the Department of Management at the University of Toronto and the Rotman School of Management at the University of Toronto. The article uses an experience sampling methodology (ESM) to gather research data. The overall structure includes (a) an introduction; (b) the definition of knowledge hiding; (c) sections on the

three studies each including methods, results, and discussion; and (d) an overall, general discussion including topics on future research, research limitations, and implications for practice.

Summary. This article ties directly to the research question on barriers to knowledge transfer. The resulting data indicate that the practice of knowledge hiding, the antithesis of knowledge transfer, takes several different forms depending on the type of information requested, trust between the knowledge keeper and the potential recipient of knowledge, and the significance of the requested knowledge as it applies to task or job requirements of the knowledge keeper. Knowledge hiding behaviors are also varied, ranging from practices of (a) "playing dumb", (b) evasion, or (c) rationalized hiding (claiming inability to transfer knowledge or blaming others as to why transference could not occur). Another dimension uncovered in the study is that how an organization practices knowledge sharing has significant impact as to how often and the type of hiding that takes place generally.

Cromity, J., & de Stricker, U. (2011). Silo persistence: It's not the technology, it's the culture! *New Review of Information Networking, 16*(2), 167-184. doi: 10.1080/13614576.2011.619924

Abstract. Knowledge workers continue to have difficulties embracing commonly accepted philosophies of sharing knowledge. This article, a combination of content analysis and professional insight, reviews some of the primary technical and behavior barriers hindering the use of collaborative technology. This article provides practical insight concerning how enterprises can address the challenges of information silos, with or without technical knowledge management or Web 2.0 solutions.

Credibility. Cromity is a UX specialist for ProQuest Dialog and the Associate Editor for *New Review of Information Networking*, a peer-reviewed journal. De Stricker has a MA and MLS and is president of de Stricker and Associates, a consulting company for knowledge management projects located in Toronto, Ontario. The article is a literature review and contains an introduction, background information, the review of selected literature, methodology, a section on findings from the field (indicated as observations from real life), and a brief conclusion. There are 27 references in the article.

Summary. This article exposes a number of different barriers to knowledge transfer and relates directly to the research question on barriers in this annotated bibliography. Barriers come from two predominant classes: (a) technical including access, system, informational overload, and inadequate metadata; and (b) behavioral including attitude, age, culture, and motivations. The authors discuss barriers in relation to corporate culture, including technical and behavioral barriers. The article provides anecdotal evidence of these barriers as to how or why they create silos in industries as well as a literature review of more recent studies into the subject.

Disterer, G. (2001). *Individual and social barriers to knowledge transfer.* Paper presented at the 34th Annual Hawaii International Conference on System Sciences, Hawaii. doi: 10.1109/HICSS.2001.927138

Abstract. Knowledge transfer is difficult, in part due to people maintaining knowledge for their own benefit and sharing it only grudgingly. The causes are many including deep cultural traditions to garner individual recognition for work. Companies understand that knowledge sharing is a significant economic resource and is found to be critical to success. Through analysis drawn from literature and from the author's own research

experiences, this paper discusses various individual and social barriers that hinder people to share and transfer their knowledge. From this analysis, the author draws some suggestions to overcome these barriers.

Credibility. Disterer is a Professor of Economics Department at Fachhochschule Hannover, Germany. The conference at which this paper was presented is in the 46th year and ranks second highest from 18 different information science conferences. The conference proceeding has an introduction, sections discussing the individual and social barriers to knowledge transfer (the underlying problem), empirical evidence supporting their problem set, suggested countermeasures to the barriers, and a summary. The writing is clear and the information is easy to follow without a significant understanding of the concept of knowledge transfer.

Summary. This proceeding ties directly to the research question on barriers to knowledge transfer. It defines four barriers individuals have in transferring knowledge including: (a) the perceived loss of power that an individual feels when passing on exclusive knowledge, (b) the concern to disclose their knowledge if they are not confident that it is special or unique, (c) the uncertainty that the knowledge has any value, and (d) a lack of motivation to share knowledge (lack of reciprocation). There are also four social barriers including (a) language usage, referring to language specific to understanding the knowledge being transferred as well as spoken language (b) conflict avoidance, referring to instances where knowledge may be controversial or contrary and thereby risky to share, (c) bureaucracy within the organization, and (d) incompatibilities between individual and corporate norms. The proceeding provides countermeasures to address these barriers including fostering a *community of practice* within the company;

the *community of practice* describes one possible knowledge transfer system addressing the research question in this study on systems that align with agile software development practices.

Massingham, P. (2008). Measuring the impact of knowledge loss: More than ripples on a pond? *Management Learning, 39*(5), 541-560. doi: 10.1177/1350507608096040

Abstract. The impact of knowledge loss on an organization is a largely unexplored area of strategic management. This article reports the findings from an in-depth case study of an organization within the Australian Department of Defence, which suggests that lost human capital may produce decreased organizational output and productivity; lost social capital may reduce organizational memory; lost structural capital may diminish organizational learning; and lost relational capital may produce disrupted external knowledge flows. The study contributes a conceptual framework that measures the impact of knowledge loss on surviving employees.

Credibility. Dr. Massingham is Senior Lecturer and Director of the Centre of Knowledge Management at the University of Wollongong, Australia. The article is published in *Management Learning*, a peer-reviewed journal on learning and knowledge management in organizations. The article is well referenced with 34 citations. The research for the article is addressed through case studies of empirical inquiry. The article provides an introduction, a detailed section describing what is measured (the loss of human, social, structural, and relational knowledge), the research strategy, findings, a conceptual model, and the conclusion. This is a scholarly article that requires moderate knowledge in the field of study for comprehension.

Summary. This article delves into the impact of knowledge loss to an organization when an individual subject matter expert is no longer able to work in their field of knowledge. This information is researched through case studies into four aspects of intellectual capital within an organization: (a) social, (b) structural, (c) relational, and (d) human. This information applies to the research question in this annotated bibliography concerning risks associated with loss of intellectual capital. The research shows that the impact is lessened depending on a variety of circumstances. For example, when evaluating relational capital, the impact of SME loss is less when the relationship is defined through interactions between positions as opposed to individuals. Several useful frameworks are provided for managers of knowledge workers to reduce risk due to SME loss.

Theme 4: How to Best Align Selected Agile Software Development Practices in Support of Knowledge Transfer

Crawford, B., Castro, C., & Monfroy, E. (2006). Knowledge management in different software development approaches. In T. Yakhno & E. Neuhold (Eds.), *Advances in Information Systems* (Vol. 4243, pp. 304-313): Springer Berlin / Heidelberg. doi: 10.1007/11890393_32

Abstract. The software development community has a wide spectrum of methodologies to use when it decides to implement a software project, ranging from Tayloristic practices to Agile methods. Software development is a knowledge intensive activity and knowledge creation and sharing are crucial parts of the software development processes. This paper presents a comparative analysis between knowledge sharing approaches of Agile and Tayloristic software development teams and includes the authors' concerns about the development of Metaheuristics.

Credibility. Crawford has a PhD in computer science from the Technical University Federico Santa Maria, Chile and is a Professor at the School of Engineering at the Pontificia Universidad Católica de Valparaíso. Castro has a PhD in computer science from the University Henri Poincaré, France and serves as an Associate Professor in the Department of Information, Technical University Federico Santa Maria, Chile. Monfroy has a PhD in computer science from the University Henri Poincaré, France and is a Professor in the information department for the University of Nantes, France. The paper presents an (a) introduction, (b) information on how software engineering and knowledge management (KM) intersect, (c) a framework for KM in software engineering – process for agile or product for Tayloristic, (d) a discussion on metaheuristics as it applies to

software development optimization and reuse, (e) a discussion on knowledge sharing (transfer) as it supports agile or Tayloristic models, and (f) a conclusion. The document is well written but technical; an understanding of the concepts involved in metaheuristics improves understanding significantly. There are 18 references.

Summary. The information presented in this article applies to the research question on which knowledge management systems align with the practices and processes of agile software development. The authors point out that the tenants of knowledge management, collaboration, communication, and coordination, are part of software development products and process and therefore should work side by side. The article explores software reuse, which shows how high reuse leads to a different knowledge management problem, more explicit knowledge that is easy to search but not easy to transfer, and that more tacit knowledge approaches, such as those used in agile environments, provide better understanding of the knowledge. The article compares traditional (Tayloristic, such as Waterfall) methods and agile (particularly XP) methods on four dimensions: (a) gathering requirements and documentation, (b) training, (c) trust and freedom, and (d) team work and roles. The authors show how agile practices create an improved environment for knowledge sharing in software development endeavors.

Kahkonen, T. (2004, 22-26 June 2004). *Agile methods for large organizations - building communities of practice.* Paper presented at the Agile Development Conference, 2004. doi: 10.1109/ADEVC.2004.4

Abstract. Agile development practices respect tacit knowledge, make communication more effective, and foster the knowledge creation process. However the current agile methods, like XP, are focused on practices that individual teams or projects need, and the

use of the methods in organizations consisting of multiple cooperating teams is difficult. The community of practice theory suggests that large agile organizations should have various overlapping, informal cross-team communities. The paper studies three agile methods developed at Nokia that use facilitated workshops to solve multi-team issues. The paper explains how to use communities of practices theory and establishes why these methods work in multi-team settings. The results of this paper suggest that workshop practices that amass people from different parts of organizations to perform a specific well-defined task can be used effectively to solve issues that span over multiple teams and to build up communities of practice. This result suggests that the community of practice concept could provide a basis for adapting agile methods for the needs of large organizations.

Credibility. Kahkonen has a Masters in science, industrial engineering, and management from Finland's Lappeenrannan Teknillinen Yliopisto and has served as Senior Manager at Nokia and is currently Executive at Accenture. The proceeding includes an introduction, related research on teams and community of practice, a discussion on the three agile approaches that Nokia developed for cross-team communication, a discussion on communities of practice and the implications for agile software development methodologies, and a conclusion. The article is supported by 61 references. The article provides an understanding of the benefits and detriments to existing agile practices as well as alternative practices to foster cross-team knowledge transfer.

Summary. This article provides information related to three research questions found in this annotated bibliography: (a) which KM systems align with agile methods, (b) which agile practices provide mechanisms for knowledge transfer, and (c) which barriers to

knowledge transfer exist between the SME and other developers. Barriers described in this article apply to how agile techniques fail in their charter to transfer knowledge when dealing with large scale companies and projects; the article points out that some agile methods, such as XP, limit cross-team access to knowledge by the methods structure – that is, one customer plus a development team comprise the members involved in any knowledge transfer. The article touts using facilitated workshops in several forms as a means to achieve cross-team knowledge transfer. Facilitated workshops establish a number of factors that enable good *community of practice* operations: (a) continuity, (b) procedure – expectations and goals, (c) participation, (d) documentation artifacts, and (e) management support.

Whitworth, E., & Biddle, R. (2007, August 13, 2007-August 17, 2007). *The social nature of agile teams.* Paper presented at Agile 2007, Washington DC. doi: 10.1109/AGILE.2007.60

Abstract. Agile methodologies represent a people-centered approach to delivering software. This paper investigates the social processes that contribute to their success. Qualitative grounded theory was used to explore socio-psychological experiences in agile teams, where agile teams were viewed as complex adaptive socio-technical systems. Advances in systems theory suggest that human agency changes the nature of a system and how it should be studied. In particular, end-goals and positive sources of motivation, such as pride, become important. Research included the questions: How do agile practices structure and mediate the experience of individuals developing software? And in particular, how do agile practices mediate the interaction between individuals and the

team as a whole? Results support an understanding of how social identity and collective effort are supported by agile methods.

Credibility. Whitworth has a masters degree in human-computer interaction from Carleton University, Ottawa, Canada. Biddle is a full professor at Carleton University for the School of Computer Science and has a PhD from the University of Canterbury in computer science. This paper was part of the AGILE 2007 conference. This paper has an introduction, the theoretical framework, the methodology used (grounded theory), results, implications, and a conclusion. The article cites 37 references. Material is clearly written and the quotations from the study are logged so as to be anonymous (batch/participant number/paragraph). The reader does not need to have specialized knowledge though a deeper understanding of agile practices aids comprehension of the material.

Summary. This article addresses many of the questions posed in this study by exploring the positive and negative aspects to the social elements of agile software development. The information serves to create indicators in team communication, which is a key element knowledge sharing. Negative aspects of agile teams may even undermine communication, fostering feelings that may cause members to become more isolated and controlled by the team. Positive aspects of agile teams concern the strengths in communication including accountability, commitment to collective goals, cohesiveness of the team, and group identity.

Conclusion

This annotated bibliography looks at two larger information management systems, (a) knowledge management and (b) agile software development methodologies, to describe a means to support knowledge transfer between a subject matter expert (SME) and members of a software development team. Barriers that prevent knowledge transfer are presented along with risks related to the loss of intellectual capital. The social aspects of some agile software development practices are described with the goal to create an environment that improves knowledge sharing.

The bibliography examines 30 selected references, organized in four themes: (a) agile software development methodologies and practices; (b) knowledge management including knowledge transfer, communities of practice, and the value of intellectual capital; (c) the core problems including the risk inherent when the SME is unable to work in their specific knowledge domain, the risks of legacy systems in general, and barriers that inhibit knowledge transfer; and (d) how to align agile software development practices with knowledge transfer systems to potentially alleviate the barriers. The goal is to inform IT managers about how to use agile software development practices to mitigate the risks associated with potential loss of intellectual capital related to legacy system knowledge held by a small number of subject matter experts.

Understanding Agile Software Development Methodologies

Agile software development methods are rooted in early IT project management systems (Williams & Kessler, 2003, p. 172). The underlying precepts of what is deemed agile are codified in the Agile Manifesto, a set of twelve principles that value: (a) “individuals and interactions over processes and tools,” (b) “working software over comprehensive documentation,” (c) “customer collaboration over contract negotiation,” and (d) “responding to change over following a plan” (Beck et al., 2001). Prior to the development of agile

methodologies, software development approaches relied on heavy documentation and significant up-front design work (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). Methods including Waterfall, followed by the slightly more iterative Spiral, led to the development of agile methods including XP and Scrum, but still relied on the creation of explicit knowledge in the form of use cases or written requirements (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). As noted in this bibliography, written documentation is often a poor conveyance of knowledge and frequently does not equal understanding (Crawford, Castro, & Monfroy, 2006).

Each agile software development method has a unique approach to get work accomplished, embracing practices that fit the methodology goals. For example, Scrum is a method that uses short iterations, close communication between the developers and the customer (product owner), brief daily status meetings (i.e., daily stand-up meeting), and regular meetings to plan each work cycle and evaluate previous cycles to improve the process (Schwaber, 2009).

Some practices such as *pair programming* originated in one methodology, Extreme Programming (XP), but are easily used in other methodologies or even outside an agile software development environment (Williams & Kessler, 2003). Using a single agile software development methodology has advantages as the practices are designed to work well with each other (Constantine, 2002); however, it is common for software development shops to "simply pick and choose the parts that seem to work for them and toss out the bits they don't like or don't believe" (Constantine, 2002, p. 2). According to Williams and Kessler (2003), the ability to custom pick agile practices may help IT managers in rigorous development systems, especially when those systems are dictated by the organization, find agile solutions to knowledge transfer needs.

A key aspect to agile software development methodologies is the underlying social contact that is necessary for an agile team to function (Kahkonen, 2004). Agile practices require constant feedback to the individual from the team, especially concerning awareness of team activity and commitment toward team goals (Whitworth & Biddle, 2007). Whitworth and Biddle (2007) state that agile teams produce a high level of social support for individuals; further, they determine that team members are more successful at fulfilling requests for knowledge in agile environments.

Barriers to knowledge transfer are more easily transcended in an environment in which (a) there is genuine concern and trust among team members, and (b) members share a common interest and feel obliged to help one another (Disterer, 2001). As noted by Crawford, Castro, and Monfroy (2006), agile practices tend to create an atmosphere of trust, which is key to overcoming social barriers to knowledge transfer; this trust is developed through (a) collective software and system ownership, (b) frequent open team discussions, (c) cooperative development and problem solving, and (d) strong interactions with the customer. Within an environment of trust, team members voluntarily share knowledge without prompting from the organization (Crawford, Castro, & Monfroy, 2006).

What is Knowledge and How is it Related to Knowledge Transfer?

Nonaka (1994) describes two primary forms of knowledge (a) explicit (or concrete), referring to knowledge that can be systematically communicated, and (b) tacit (or intangible), referring to knowledge based in specific context, action, or experience. These two forms of knowledge comprise the basis of intellectual capital, that is, the elusive value in a company that does not appear on a balance sheet (Rus & Lindvall, 2002). Intellectual capital is further subdivided into three forms, each of which requires distinct handling to ensure it remains within

the company: (a) human capital (tacit knowledge and skills), which requires informal knowledge sharing; (b) structural capital (culture, process and procedures), which is preserved by job rotation and via *communities of practice;* and (c) relational capital (influences through external knowledge), which is maintained through cooperative interactions between internal and external entities (Kong & Thomson, 2008). Intellectual capital is at risk when fragile legacy systems are dependent on a solitary or legacy subject matter expert (SME). To reduce the risks, a company needs to establish a successful knowledge transfer system.

Knowledge transfer occurs through a wide variety of systems; for example, in *Working Knowledge,* Davenport and Prusak (2000) describe knowledge markets where buyers, that is, people looking for information, and sellers, that is, people with specialized knowledge, go through knowledge brokers, managers and corporate librarians, to facilitate the exchange of ideas. Some knowledge transfer systems are informal and include job rotation, staff mentoring and training, or basic two-way interactions (Alavi & Leidner, 1999). Others embrace more defined practices such as facilitated workshops (Kahkonen, 2004) or company-based *community of practice* (Wasko & Faraj, 2000). De Long and Davenport (2003), suggest that it is difficult to determine the types of knowledge that require transfer. They propose a process of asking five questions: (a) how long is this knowledge valuable, (b) is the knowledge easy to store explicitly or can it only be experienced or shared, (c) is the subject matter expert likely to leave the company soon, (d) how willing is the subject matter expert to share their knowledge, and (e) what costs will the company need to cover to ensure knowledge is transferred? Once the five questions are addressed, an IT manager is better able to determine the level of risk that a legacy system SME creates within the company.

A *community of practice* refers to a group of people "bound together by shared expertise and passion for a joint enterprise" (Wenger & Snyder, 2000, p. 139). Pawlowski and Robey (2004) believe that members of a *community of practice* tend to develop a mutually held world view where beliefs, values, definitions, assumptions, and practices reflect a shared body of knowledge. The membership is bound together by common goals regardless of an individual member's role within the company (Pawlowski & Robey, 2004). Disterer (2001) believes that a *community of practice* helps overcome individual and social barriers that interfere with knowledge transfer. Alavi and Leidner (2001) point out that a *community of practice* within an organization does not only serve as a means to support tacit knowledge transfer, it is also a vital element for knowledge creation by maintaining a collective knowledge base, a common language related to the knowledge shared, and a forum where ideas can be openly discussed and freely challenged.

What are the Risks Associated with a Subject Matter Expert or a Legacy System?

Legacy systems, an inherit part of the IT infrastructure, are difficult to maintain and risky to replace (Visaggio, 2001). Once code is written or a system is set in place, it begins a slow decline into obscurity (Visaggio, 2001). These artifacts of the IT domain frequently fall into the hands of a few technology subject matter experts (SMEs) who are required to maintain a critical understanding of the operations necessary to care for these aging systems (Massingham, 2008).

Leaving valuable legacy system knowledge in the hands of a limited number of experts creates risk for an IT manager to consider. Risks include:

- The risk associated with cost, in that system experts may have higher quality standards of work, but they are more costly than cross-trained counterparts (Marentette, Johnson, & Mills, 2009);

- The risk associated to an accident, illness, or job attrition is magnified in the case of solitary knowledge experts whose absence may leave the enterprise in a potentially unstable state (Kong & Thompson, 2008, p.358); and
- The risk associated with knowledge when it is isolated to a single person. New knowledge in an organizational setting, whether in the form of new ideas or new products, is a synthesis of knowledge from different sources (Carlile & Rebentisch, 2003).

Massingham (2008) points out that a company creates value through intellectual capital; he examines risk in four categories: (a) loss of relational capital in which losing a SME creates less risk to the exchange of information when the role that the SME holds is formal and well defined, (b) loss of structural capital in which a SME leaving the company reduces the company's ability to spread knowledge through organizational learning, (c) loss of human capital which is the tacit or experiential knowledge held by individuals that is no longer transferable, and (d) loss of social capital in which the knowledge maintained in the company's knowledge network is disrupted by the loss of a SME. Further, Massingham (2008) examines levels of risk within each category, and demonstrates that risks may be mitigated by increasing the number of people involved (human capital risk), or establishing higher degrees of formality to a role, thus limiting specialized knowledge (relational capital risk).

Attempts to capture knowledge in explicit forms such as documentation also create risks. Traditional (also called Tayloristic) software development methodologies strongly rely on a heavy documentation model through systems "to ensure all possible requirements, design, development, and management issues are addressed and captured" (Crawford, Castro, & Monfroy, 2006, p. 311). Crawford, Castro, and Monfroy (2006) further point out that

heavyweight methods that rely on extensive documentation have a smaller chance of being properly and accurately maintained than agile alternatives (2006). Traditional documentation models are inherently risky in that explicit knowledge artifacts require significant time to create and are often produced by a single author as opposed to a network of knowledgeable people (Crawford, Castro, & Monfroy, 2006; Kahkonen, 2004). An IT manager should eschew using explicit knowledge devices as their only form of knowledge transfer as tacit knowledge (intangible, experiential) frequently cannot be transferred without a shared experience (Nonaka, 1994).

What are the Barriers that Inhibit Knowledge Transfer?

In order to reduce risks related to the loss of intellectual capital, an IT manager needs to understand the barriers that prevent the free flow of knowledge among software developers. Several categories of barriers may arise when attempting to extract systems knowledge from the SME (Connelly, Zweig, Webster & Trougakos, 2010; Ford & Staples, 2008; Pawlowski & Robey, 2004; Visaggio, 2001). Disterer (2001) shows that barriers to knowledge transfer can be categorized as either individual or social. There are four different barriers that may inhibit knowledge transfer from the perspective of the individual: (a) the possible loss of power an individual feels as the holder of exclusive knowledge and the respect that having that knowledge creates, (b) the fear an individual holds when revealing the knowledge that others may discover that the knowledge is neither rare or exclusive, (c) the uncertainty an individual holds as to the value of knowledge, and (d) the expectation of quid pro quo when an individual sees no reason to share without reciprocation from the exchange (Disterer, 2001). There are five social barriers to knowledge transfer including (a) the need to understand specialized language associated with the knowledge base, (b) linguistic difficulties that arise when communication transpires between

different spoken languages, (c) the desire of an individual to avoid conflict that may arise when knowledge is counter to established norms, (d) bureaucratic or hierarchical barriers when knowledge is restricted or controlled, and (e) incompatible or incoherent paradigms from the company stemming from strategies, the corporate mission and goals, vision, and intent (Disterer, 2001).

Another barrier to knowledge transfer is the use of codified explicit knowledge, (i.e., documentation) to mitigate the risk; this is inadequate as developers often do not maintain documents sufficiently, nor do they record the real layout of the system or decisions made at the time of change (Visaggio, 2001, p. 285). Other reported barriers affecting knowledge transfer include (a) knowledge hiding (Connelly, Zweig, Webster & Trougakos, 2010); (b) knowledge hinting, or knowledge hoarding (Ford & Staples, 2008); and (c) behaviors that may inhibit or slow knowledge transfer from the SME to others.

To counteract barriers to knowledge transfer, an IT manager should use one or more of the following measures: (a) manage in ways that provide a nurturing environment with clear communications, (b) create a work environment where trust and concern exist between members of the network, (c) provide both financial and recognition incentives, and (d) foster and support informal *communities of practice* (Disterer, 2001). Leadership is used to overcome barriers in two ways: (a) by example, when leaders embrace knowledge sharing through actions that show commitment to transferring knowledge; and (b) by providing time for workers to network with other workers (Disterer, 2001). Disterer (2001) states that a manager can create a setting that counteracts barriers to knowledge transfer by creating a work environment of concern and trust that supports “interest for different viewpoints and experiences, access to help, lenience in judgement, courage to voice opinions, to allow experiments and to take risks” (p. 4). The use of

incentives provides a means to encourage cross-training, mentoring, tutoring, and other forms of knowledge transfer; incentives may be financial means or through recognition such as trophies and awards (Disterer, 2001). Lastly, barriers to knowledge transfer can be overcome by the establishment of *communities of practice* within the company; these *communities of practice* require time for members of the community to gather, and management needs to provide tools required by the group, whether these are software applications or whiteboards (Disterer, 2001).

Tying *Communities of Practice* and Agile Software Development Methods

Wenger and Snyder (2000) provide a concise definition of a *community of practice* that describes a group of passionate people who share expertise. Agile team culture tends to create such a group through a "complex system of values, principles, and practices" and a strong "commitment to collective goals and cohesiveness" (Whitworth & Biddle, 2007, p. 8, sect. 4.2). According to Williams and Kessler (2003), the agile practice of *pair programming* serves as an effective approach to share insights and ideas that creates a *community of practice* when used with the technique of *pair rotation*. For larger organizations such as Nokia, Kahkonen (2004) reports that divisions using different agile software development methods independently created *facilitated workshops*, practices designed to create a *community of practice* with a goal of transferring knowledge between distributed teams. Another agile practice, the daily stand-up meeting used in Scrum, may create the same passion among team members as a *community of practice* by engendering a high level of enthusiasm and backing of the participants (Whitworth & Biddle, 2007). The daily stand-up meeting also shows that the community is not just limited to developers but may include product owners, stakeholders, or managers (Abrahamsson, Salo, Ronkainen, & Warsta, 2002), which is a larger and more diverse pool of engaged participants.

The fifth principle behind the Agile Manifesto (Beck et al, 2001) shows similarities to the knowledge transfer system *community of practice*. The first half of the statement in the fifth principle of the Agile Manifesto is "build projects around motivated individuals" (Beck et al, 2001, *Principles behind the Agile Manifesto*) and strongly resembles Wegner and Snyder's (2000) definition of a *community of practice* as a group that is "bound together by shared expertise and passion for a joint enterprise" (p. 139). The second half of the statement of the fifth principle of the Agile Manifesto, conveys that developers have "the environment and support they need, and trust them to get the job done" (Beck et al., 2001, *Principles behind the Agile Manifesto*); this statement similarly mirrors Disterer's (2001) *community of practice* measure to counter barriers to knowledge transfer, which shows that a company must provide the management support, a place to meet and exchange information (physically and virtually), and the means to allow for *communities of practice* to function within the corporation.

Removing Barriers to Knowledge Management through Agile Software Development Methods

This annotated bibliography examines how selected agile practices can help reduce risks and overcome barriers to knowledge transfer within an IT department. In particular, focus is on social mechanisms within agile software development methods enable knowledge transfer from an individual, team, or organizational perspective. Frequently, an agile practice will transcend over two perspectives, such as when *pair programming* uses regular rotations to create a larger pool for the shared knowledge, or during daily stand-up meetings where the team creates an atmosphere of openness and trust while also incorporating participants from outside their group.

Pair programming is the agile practice in which two developers collaborate to design, code, debug, and test software (Williams & Kessler, 2003). One developer, the driver, acts as the

primary person at the keyboard and generally writes the code, draws the design, etc. The second developer, the navigator, acts as an observer looking for defects, suggesting strategies, thinking of long term implications of the design (Williams & Kessler, 2003). The practice of *pair programming* can be beneficial to facilitate knowledge transfer if the pairs are the right type. Williams and Kessler (2003) note that pairing experts shows worth if the average or novice programmer asks the right questions; the non-expert can also provide expertise learned outside the pairing, when applicable. The expert, whether acting in the primary role (the driver) or the secondary role (the navigator), is able to convey crucial system tricks and tools; the non-expert provides fresh insights and can draw knowledge from experts by asking well-considered questions (Williams & Kessler, 2003). By switching roles between driver and navigator, knowledge transfer is able to diffuse knowledge quickly among the pair and, if using pair rotation, among the entire team (Williams & Kessler, 2003).

Daily stand-up meetings are short by design, typically limited to 15 minutes, in which each member of the team conveys three pieces of information: (a) what they worked on since the last meeting, (b) what they intend to work on next, and (C) anything blocking their progress to getting work done (Schwaber, 2009). Team members may offer help to individuals during this meeting by discussing ideas for upcoming work or by helping to remove barriers. *Daily stand-up meetings* are a key part of the collective ownership of systems and lead to team trust and a sense of individual autonomy; this environment creates a setting that supports efficient knowledge generation and sharing (Crawford, Castro, & Monfroy, 2006). Within the Scrum-driven community of developers, system knowledge is shared by continuous communications, constant collaboration, strong social support, and personal accountability (Whitworth & Biddle, 2007). Other Scrum practices such as the *sprint retrospective* create an atmosphere in which continuous

improvement and supported learning (individual, team, organizational) are part of the team process on a frequent, recurring basis (Dingsøyr, Bjørnson, & Shull, 2009).

Agile software development practices address aspects of communication that create a team environment of openness, trust, and support (Crawford, Castro, & Monfroy, 2006). This environment mirrors the knowledge transfer system known as *community of practice* (Wenger & Snyder, 2000). At the individual level, a pair of programmers working together on a system (a) share insights and (b) expand their learned knowledge through a cycle of regular partner rotations (Williams & Kessler, 2003). At the team level, knowledge sharing occurs through regular gatherings that provide social support and respect. The team environment includes an expectation of reasonable accountability that helps to break down barriers to knowledge transfer (Whitworth & Biddle, 2007) and creates a spirit of élan that can be viewed as a *community of practice*. At an organizational level, agile groups interact outside the team by including product owners and stakeholders into the team's meetings, or more formally through facilitated workshops (communities across multiple teams) as a means to disperse knowledge and expertise that each team holds for the company at large (Kahkonen, 2004).

When a company has a SME dedicated to working with a legacy system, the risk associated with the siloed subject matter expert (Kong & Thompson, 2008; Marentette, Johnson, & Mills, 2009) adds to the risks and costs to replace a legacy software system (Visaggio, 2001). By adopting one or more agile practices, an IT manager can address knowledge transfer barriers and reduce the risks presented by isolated knowledge through use of transfer systems such as *communities of practice* (Kahkonen, 2004; Wegner & Snyder, 2000). The results will help preserve the corporation's hidden asset, *intellectual capital*. In keeping with the spirit of agile software development, an IT manager can pick and choose an agile practice that fits their

operational need and organizational culture without adopting an entire methodology (Constantine, 2002).

References

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile software development methods : Review and analysis*. Espoo [Finland]: VTT. doi: 10.1.1.161.5931

Alavi, M., & Leidner, D. (1999). *Knowledge management systems: Emerging views and practices from the field.* Paper presented at the Thirty-second Annual Hawaii International Conference on System Sciences, Hawaii. doi: 10.1109/HICSS.1999.772754

Alavi, M., & Leidner, D. E. (2001). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly, 25*(1), 107-136.

Ambler, S. (2008). When it gets cultural: Data management and agile development. *IT Professional, 10*(6), 11-14. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4747648&isnumber=4747641&tag=1

Argote, L., & Ingram, P. (2000). Knowledge transfer: A basis for competitive advantage in firms. *Organizational Behavior and Human Decision Processes, 82*(1), 150-169. doi: 10.1006/obhd.2000.2893

Arisholm, E., Gallis, H., Dyba, T., & Sjoberg, D. I. K. (2007). Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering , 33*(2), 65-86.

Bari, M. A., & Ahamad, S. (2011). Managing knowledge in development of agile software. *International Journal of Advanced Computer Science and Applications, 2*(4), 72-76.

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change*. Boston, MA: Addison-Wesley.

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). Principles behind the agile manifesto. Retrieved November 6, 2011, from http://agilemanifesto.org/principles.html

Begel, A., & Nagappan, N. (2007, 20-21 Sept. 2007). *Usage and perceptions of agile software development in an industrial context: An exploratory study.* Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, 2007. Retrieved from http://research.microsoft.com/en-us/um/redmond/groups/hip/papers/agiledevatms.pdf

Bell, C. (2009, May 19). *UO Libraries: Critical evaluation of information sources.* Retrieved from http://libweb.uoregon.edu/guides/findarticles/credibility.html

Benedicenti, L., & Paranjape, R. (2001). *Using extreme programming for knowledge transfer.* Paper presented at the 2nd International Conference on eXtreme Programming and Agile Processes in Software Engineering. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.2113&rep=rep1&type=pdf

Bjørnson, F. O., & Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology, 50*(11), 1055-1068. doi: 10.1016/j.infsof.2008.03.006

Boehm, B. W. (1991). Software risk management: Principles and practices, *8,* 32-41. Retrieved from http://doi.ieeecomputersociety.org/10.1109/52.62930

Bontis, N. (1999). Managing organisational knowledge by diagnosing intellectual capital: framing and advancing the state of the field. *International Journal of Technology Management, 18*(5), 433. Retrieved from http://www.business.mcmaster.ca/mktg/nbontis/ic/publications/ijtmbontis.pdf

Brown, J. S., & Duguid, P. (2001). Knowledge and organization: A social-practice perspective. *Organization Science, 12*(2), 198-213. Retrieved from http://www.jstor.org/stable/3086055

Brusco, M. J., & Johns, T. R. (1998). Staffing a multiskilled workforce with varying levels of productivity: An analysis of cross-training policies. *Decision Sciences, 29*(2), 499-515. doi: 10.1111/j.1540-5915.1998.tb01586.x

Busch, C., De Maret, P., Flynn, T., Kellum, R., Le, S., Meyers, B., Saunders, M., White, R. (2005). *Content analysis*. Writing@CSU. Colorado State University. Retrieved December 10, 2011 from http://writing.colostate.edu/guides/research/content/printformat.cfm?printformat=yes

Carlile, P. R., & Rebentisch, E. S. (2003). Into the black box: The knowledge transformation cycle. *Management Science, 49*(9), 1180-1195. Retrieved from http://www.jstor.org/stable/4134034

Chau, T., Maurer, F., & Melnik, G. (2003, 9-11 June 2003). *Knowledge sharing: Agile methods vs. Tayloristic methods.* Paper presented at the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. doi: 10.1109/ENABL.2003.1231427

Cockburn, A., & Williams, L. (2000). *The costs and benefits of pair programming.* Paper presented at the eXtreme Programming and Flexible Processes in Software Engineering -- XP2000, Cagliari, Sardinia, Italy. doi: 10.1.1.26.9064

Coman, I. D., Sillitti, A., & Succi, G. (2008). Investigating the usefulness of pair-programming in a mature agile team. In P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L.

Morgan & X. Wang (Eds.), *Agile processes in software engineering and extreme programming* (Vol. 9, pp. 127-136). doi: 10.1007/978-3-540-68255-4_13

Connelly, C. E., Zweig, D., Webster, J., & Trougakos, J. P. (2010). Knowledge hiding in organizations. *Journal of Organizational Behavior*. doi: 10.1002/job.737

Constantine, L. L. (2002). Process agility and software usability: Toward lightweight usage-centered design. *Information Age, 8*(2). Retrieved from http://121.52.210.238/jiaohu/pdf/agiledesign.pdf

Crawford, B., Castro, C., & Monfroy, E. (2006). Knowledge management in different software development approaches. In T. Yakhno & E. Neuhold (Eds.), *Advances in Information Systems* (Vol. 4243, pp. 304-313): Springer Berlin / Heidelberg. doi: 10.1007/11890393_32

Cromity, J., & de Stricker, U. (2011). Silo persistence: It's not the technology, it's the culture! *New Review of Information Networking, 16*(2), 167-184. doi: 10.1080/13614576.2011.619924

Davenport, T. H., & Prusak, L. (2000). *Working knowledge: How organizations manage what they know*: Harvard Business Press.

De Long, D. W., & Davenport, T. (2003). Better practices for retaining organizational knowledge: Lessons from the leading edge. *Employment Relations Today, 30*(3), 51-63. doi: 10.1002/ert.10098

Demarest, M. (1997). Understanding knowledge management. *Long Range Planning, 30*(3), 374-384. doi: 10.1016/s0024-6301(97)90250-8

Dingsøyr, T., Bjørnson, F.O., & Shull, F. (2009). What do we know about knowledge management? Practical implications for software engineering. *IEEE Software, 26,* 100-103. doi: 10.1109/MS.2009.82

Disterer, G. (2001). *Individual and social barriers to knowledge transfer.* Paper presented at the 34th Annual Hawaii International Conference on System Sciences, Hawaii. doi: 10.1109/HICSS.2001.927138

Eckel, N. T. (2010). Collaborating with subject matter experts. *T+D, 64*(3), 76-77. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=48407706&site=ehost-live&scope=site

Fægri, T. E., Dybå, T., & Dingsøyr, T. (2010). Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. *Information and Software Technology, 52*(10), 1118-1132. doi: 10.1016/j.infsof.2010.06.002

Ford, D. P., & Staples, D. S. (2008). What is knowledge sharing from the informer's perspective? *International Journal of Knowledge Management, 4*(4), 1.

Fronza, I., Sillitti, A., & Succi, G. (2009, 15-16 Oct. 2009). *An interpretation of the results of the analysis of pair programming during novice's integration in a team.* Paper presented at the Empirical Software Engineering and Measurement, 2009. doi: 10.1109/ESEM.2009.5315998

Gabriel, S. (2000). The process of knowledge transfer: A diachronic analysis of stickiness. *Organizational Behavior and Human Decision Processes, 82*(1), 9-27. doi: 10.1006/obhd.2000.2884

García, J., Amescua, A., Sánchez, M.-I., & Bermón, L. (2011). Design guidelines for software processes knowledge repository development. *Information and Software Technology, 53*(8), 834-850. doi: 10.1016/j.infsof.2011.03.002

Goodwin, S. (2009). *Knowledge management sharing in medium-to-large organization: Constraints, enablers and alignment.* PHD, University of Bath, Bath. Retrieved from http://opus.bath.ac.uk/18225/

Granier, T. B. (2007, July 27, 2007). *Cross-training: A case study*. Retrieved November 6, 2011, from http://www.sans.edu/research/leadership-laboratory/article/granier-mgt421

Gupta, B., Lakshmi, S. I., & Aronson, J. E. (2000). Knowledge management: Practices and challenges. *Industrial Management & Data Systems, 100*(1), 17-21. doi: 10.1108/02635570010273018

Hannay, J. E., Arisholm, E., Engvik, H., & Sjoberg, D. I. K. (2010). Effects of personality on pair programming. *Software Engineering, IEEE Transactions on, 36*(1), 61-80. doi: 10.1109/TSE.2009.41

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer, 34*(9), 120-127. doi: 10.1109/2.947100

Holz, H., & Maurer, F. (2003). Knowledge management support for distributed agile software processes advances in learning software organizations. In S. Henninger & F. Maurer (Eds.), (Vol. 2640, pp. 60-80): Springer Berlin / Heidelberg. doi: 10.1007/978-3-540-40052-3_7

InformIT (n.d.). *Authors*. Various author bibliography data retrieved from http://www.informit.com/authors/index.aspx

Jennex, M. E. (2009). *Assessing knowledge loss risk.* Paper presented at the AMCIS 2009. Retrieved from http://aisel.aisnet.org/amcis2009/446

Kahkonen, T. (2004, 22-26 June 2004). *Agile methods for large organizations - building communities of practice.* Paper presented at the Agile Development Conference, 2004. doi: 10.1109/ADEVC.2004.4

Kong, E., & Thomson, S. B. (2008). An intellectual capital perspective of human resource strategies and practices. *Knowledge Management Research & Practice, 7*(4), 356-364. doi: 10.1057/kmrp.2009.27

Lee, M. F., & Mehlenbacher, B. (2000). Technical writer/subject-matter expert interaction: The writer's perspective, the organizational challenge. *Technical Communication, 47*(4), 544-552.

Leedy, P. D., & Ormrod, J. E. (2001). *Practical research: Planning and design* (7 ed.): Prentice-Hall, Inc.

Lui, K. M., & Chan, K. C. C. (2006). Pair programming productivity: Novice-novice vs. Expert-expert. *International Journal of Human-Computer Studies, 64*(9), 915-925. doi: 10.1016/j.ijhcs.2006.04.010

Marentette, K. A., Johnson, A. W., & Mills, L. (2009). A measure of cross-training benefit versus job skill specialization. *Computers & Industrial Engineering, 57*(3), 937-940. doi: 10.1016/j.cie.2009.03.010

Maruping, L. M., Zhang, X., & Venkatesh, V. (2009). Role of collective ownership and coding standards in coordinating expertise in software project teams. *European Journal of Information Systems, 18*(4), 355-371. doi: 10.1057/ejis.2009.24

Massingham, P. (2008). Measuring the impact of knowledge loss: more than ripples on a pond? *Management Learning, 39*(5), 541-560. doi: 10.1177/1350507608096040

Microsoft Research. (n.d.). *People.* Retrieved from http://research.microsoft.com/en-us/people/default.aspx

Moe, N. B., & Dingsøyr, T. (2008). Scrum and team effectiveness: Theory and practice agile processes in software engineering and extreme programming. In P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan & X. Wang (Eds.), (Vol. 9, pp. 11-20): Springer Berlin Heidelberg. doi: 10.1007/978-3-540-68255-4_2

Mohan, K., Ramesh, B., & Sugumaran, V. (2010). Integrating software product line engineering and agile development. *Software, IEEE, 27*(3), 48-55. doi: 10.1109/MS.2010.31

Nerur, S., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM, 50*(3), 79. doi: 10.1145/1226736.1226739

Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science, 5*(1), 14-37. Retrieved from http://www.jstor.org/stable/2635068

O'Dell, C. S., Grayson, C. J., & Essaides, N. (1998). *If only we knew what we know: The transfer of internal knowledge and best practice*. New York: Free Press.

Parise, S., Cross, R., & Davenport, T. H. (2005). It's not what but who you know: How organizational network analysis can help address knowledge loss crises. *Lost Knowledge Roundtable, The Network Roundtable at the University of Virginia*. Retrieved from http://www.robcross.org/pdf/roundtable/lost_knowledge.pdf

Pawlowski, S. D., & Robey, D. (2004). Bridging user organizations: Knowledge brokering and the work of information technology professionals. *MIS Quarterly, 28*(4), 645-672. Retrieved from http://www.jstor.org/stable/25148658

Poff, M. A. (2003). *Pair programming to facilitate the training of newly-hired programmers.* Masters, Florida Institute of Technology. Retrieved from http://www.cs.fit.edu/media/TechnicalReports/cs-2003-08.pdf

Rus, I., & Lindvall, M (2002). Guest editors' introduction: Knowledge management in software engineering, *19*, 26-38. doi: 10.1109/MS.2002.1003450

Salo, O. A. P. (2008). Agile methods in European embedded software development organisations: A survey on the actual use and usefulness of extreme programming and scrum. *IET Software*, *2*(1), 58-64. doi: 10.1049/iet-sen:20070038

Scacchi, W. (2002). *Process models in software engineering*: John Wiley & Sons, Inc. doi: 10.1002/0471028959.sof250

Schwaber, K. (2009). *Agile project management with scrum.* [Adobe Digital Editions version]. Retrieved from http://multco.eblib.com/patron/FullRecord.aspx?p=488725

Subject Matter Expert. (n.d.). In *iSixSigma Glossary*. Retrieved from http://www.isixsigma.com/index.php?option=com_glossary&id=332

U.S. Small Business Administration. *Summary of size standards by industry*. (2011) Retrieved 11/21/2011, 2011, from http://www.sba.gov/content/summary-size-standards-industry

University of North Carolina. (n.d.). *Writing center: Literature reviews*. Retrieved from University of North Carolina at Chapel Hill, Writing Center Web site: http://www.unc.edu/depts/wcweb/handouts/literature_review.html

Vanhanen, J., Lassenius, C., & Mantyla, M. (2007). *Issues and tactics when adopting pair programming: A longitudinal case study.* Paper presented at the International Conference of Software Engineering Advances. doi: 10.1109/ICSEA.2007.48

VTT. (n.d.). *Publications.* Retrieved from VTT business site: http://www.vtt.fi/publications/?lang=en

Visaggio, G. (2001). Ageing of a data-intensive legacy system: Symptoms and remedies. *Journal of Software Maintenance and Evolution: Research and Practice, 13*(5), 281-308. doi: 10.1002/smr.234

Volpe, C., Cannon-Bowers, J., Salas, E. (1996). The impact of cross-training on team functioning: An empirical investigation. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 38*, 87-100. Retrieved from http://go.galegroup.com/ps/i.do?id=GALE%7CA18474088&v=2.1&u=s8492775&it=r&p=CDB&sw=w

Von Hippel, E. (1994). "Sticky information" and the locus of problem solving: Implications for innovation. *Management Science*, 429-439. Retrieved from http://www.jstor.org/stable/pdfplus/2632751.pdf

Wasko, M., & Faraj, S. (2000). "It is what one does": Why people participate and help others in electronic communities of practice. *The Journal of Strategic Information Systems, 9*(2-3), 155-173. doi: 10.1016/s0963-8687(00)00045-7

Wenger, E. C., & Snyder, W. M. (2000). Communities of practice: The organizational frontier. *Harvard Business Review, 78*(1), 139-145. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=2628915&site=ehost-live&scope=site

Whitworth, E., & Biddle, R. (2007, August 13, 2007-August 17, 2007). *The social nature of agile teams.* Paper presented at Agile 2007, Washington DC. doi: 10.1109/AGILE.2007.60

Williams, L., & Kessler, R. R. (2003). *Pair programming illuminated*: Addison-Wesley Professional.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *Software, IEEE, 17*(4), 19-25. doi: 10.1109/52.854064

Wolfe, C. T. (2008). Knowledge sharing: The effects of incentives, environment, and person. *Journal of Information Systems, 22*(2), 53-76. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=mth&AN=34946011&site=ehost-live&scope=site