A New Teaching Paradigm in Information Systems Education:

An Investigation and Report on the Origins, Significance, and Efficacy of the Agile Development Movement

Samuel S. Conn[1]
School for Professional Studies Graduate Program
Regis University
3333 Regis Boulevard
Denver, Colorado 80021

Abstract

Systems and software development life cycles are fundamental to systems analysis and design coursework in Information Systems education. There is today a movement away from the traditional, linear development life cycle models toward the newer "Agile" development life cycle models. This paradigm shift affects the way we are approaching how we teach the topic of development life cycle models. For the Information Systems educator it is important to know about the origins, significance, and current efficacy of the Agile development movement, and to understand the paradigm shift occurring in industry today. This is an investigation and research into three principal areas related to the Agile development movement and how it is affecting Information Systems education. The first area is how the movement was organized, facilitated, and sponsored, with a focus on how the seventeen charter members of the movement came together. The members, known as the Agile Alliance, met at the Lodge in Snowbird, Utah in February 2001 to find common ground in their various practices of adaptive and iterative software and system development methodologies. The second area of investigation and report concerns how the Agile Manifesto, the main deliverable of the Snowbird meeting, was discussed, written, approved and endorsed by all seventeen founding members of the movement in such a short period of time. The third area of investigation and report concerns how the Snowbird meeting and the Agile movement are impacting the software and systems development industry and Information Systems education. A summary and analysis of the findings and the impact of the paradigm shift's effect on Information Systems education are a part of this report. The reliability and validity of the investigation and research was greatly enhanced by the direct interaction and supply of information from five of the seventeen founding members of the Agile movement. This research and report also provides exclusive detail about the Snowbird meeting and the reflections of the participants in their continued attempt to further advance the Agile development methodology movement.

Keywords: agile methodology, SDLC, IS education, development life cycle

1. AGILE METHODOLOGY

Introduction

Development life cycle methodologies affect both software and system development projects. This critical concept finds application in software and application development companies, and at the enterprise level when designing and implementing new information systems. The role of the Systems Development Life Cycle (SDLC) model in software and systems

[1] sconn@regis.edu

development has generally been defined along structured lines since the linear, traditional methodologies achieved acceptance. There are three major traditional models that have been used extensively. They represent an era of what has been termed "heavy" methodology development. The term "heavy" was used because of the characteristic requirement for heavy documentation, heavy process, and long, sequential phases that defined the development life cycle models. Winston Royce, Barry Boehm, Daniel D. McCracken, Michael A. Jackson, and G.R. Gladden were the main contributors to these three major traditional development methodologies known as the Waterfall Model, the Prototyping Model, and the Spiral Model. The Waterfall Model was the first structured approach to systems development. It was developed by Winston Royce and introduced in 1970 in his publication "Managing the Development of Large Software Systems" (Royce, 1970). The Waterfall Model provides abstraction of the critical activities in the software development process, and lists them in a sequential order based on their relative dependencies. Cynical treatment of the model suggests that real software development projects do not use the model literally, since the model is applied to itself in a recursive manner. This yields a somewhat chaotic flow of real time activity. Additionally, the sequential stages are too lengthy to provide for the valuable asset of fast time to market in the competitive world of software and system development. Despite this treatment, the Waterfall Model is actually correct in that each step provides the most accurate platform for the next one. As a general concept, the Waterfall Model makes perfect sense, but in the real world of software development, the model looses efficacy. In August of 1986, Barry Boehm first introduced the Spiral Model of development in the ACM SIGSOFT Software Engineering Notes. Boehm (1986) later published the Spiral Model in the IEEE journal, fueling the increasing popularity of his model. He describes the model as innovative because it permits combinations of the conventional phases, and is incremental and iterative in construct. It shifts the focus from developmental phases to risk, and calls for clear evaluations of whether or not a project should be terminated at any point. The point of the model is that the form of a development cannot be determined to any level of precision in advance of the development activities. Boehm (1986) looks at the model as a process model generator, where a given set of conditions allows the spiral to produce detailed development. The Prototyping Model has an interesting development history with the influence of G.R. Gladden, who in his work on the Principle of Limited Reduction, is actually opposed to development methodologies in general. Gladden (1982) published an article titled "Stop the Life-cycle, I Want to Get Off" that reported that 35% of delivered software is not used because of the gap between it and the user's concept of the system. Daniel D. McCracken and Michael A. Jackson took to heart the report from Gladden and decided that if the user could be more involved in the development process, then higher efficacy could be achieved. So the Prototyping Model, which later became known as *Rapid Prototyping*, became a widely used development methodology. In this model, the amount of requirements documentation that is required is dependent upon the specifics of the problem. This model generally was used for development in two areas: user interfaces and heavily transaction-oriented functions such as database operations. This iterative building process allowed for user response to the development as the development occurred, through the use of "mockups" to simulate the final system to the user, make changes accordingly, and refine the end product to more closely align with the end user's requirements as the development occurred.

Problem statement to be investigated

The advent of the Agile methodology movement displaced the convention of the day, and has generated several new methodologies as a result. As the need for faster time to market, shorter development cycles, lower development cost, and the ability to move and change quickly have become a critical element in the organization's ability to compete in the marketplace; the traditional "heavy" methodologies have proven themselves to be a liability rather than an asset. This is because, characteristically, they have traits that do not lend themselves to facilitating the newer business needs, as previously stated. So as smaller companies moved into the competitive arena, and larger companies have streamlined to become more effective, the Agile movement has begun to take hold and prove itself viable. Alternative views to the Agile movement can argue that it is just old concepts repackaged, sloppy management of development, or just a west coast trend in development methodologies. The problem addressed in this report is with respect to the validity of the concept and efficacy of the methodologies. There is limited research available, since the Agile movement is relatively new, so a major goal of the research was to contact the principals of the Agile movement, and ask direct questions of them relative to the problem statement. Initially, all seventeen of the men who formed the Agile Alliance were contacted via email, and asked to participate in a survey. Of the seventeen requests, nine members accepted the offer to participate in the survey, one member declined, and seven members did not reply to the request. Actual replies were received from five. This research will rely, in part, on the survey responses from Jim Highsmith, Alistair Cockburn, Kent Beck, Ron Jeffries, and Steve Mellor. The research goal was achieved through the review of literature and direct communications with the principals of the Agile Alliance.

Goal(s) of the research and report

The origin of the project idea came from the author's interest in how the Agile Alliance formed and the impact that it has had on Information Systems education. The opportunity to research and investigate the genesis of a new Information Technology movement, with access to the principals, formed the basis of a strong and relevant area of research. Beyond the principles and concepts of the Agile development methods, the real story of the Agile movement is the collaboration and agreement of the seventeen thought leaders who established and implemented the movement.

The goals of the research are two fold: first, research on the story behind the assembly of these particular men to create the Agile Alliance; and second, an investigation of the impact the movement is having on the industry and Information Systems education. Having made contact with the members of the Agile Alliance to secure their participation in the research, the ability to adequately research the group's history and collaboration was materialized.

The relevance of the research goals is that an emerging methodology that fundamentally changes the way software and systems development has been viewed for many years is highly significant to the Information Systems development process. The study of systems development includes software development methodology, so the major changes that the Agile methodologies are suggesting are being seen in the implementation of various vendor products, IT conferences and training, and IT projects in industry.

Relevance of the Report to Information Systems Education

This report is relevant because the occurrence of this significant event is still a subject of conversation at conferences and in trade journals, and is greatly impacting both the way software development projects are being addressed and taught, and the technologies that are being developed to employ the methods. In some ways, the "jury is still out" on the proliferation, acceptance, and success of the Agile methodologies. This current area of evolution calls for additional research and informed assumptions to be made about whether the Agile Manifesto is about an alternative to traditional methodologies, or a fundamental shift away from traditional methodologies.

The signatories of the Agile Manifesto are apparently working hard to get out the message of the "Agile revolution", but it is also important to understand who is listening to them, and what impact is it having now and in the future on Information Systems development. In the study of the Information Systems development process, any new thinking, new methods, and new concepts are directly related to the topic of study.

Barriers and issues to the research

The charter members of the Agile Alliance met in somewhat of a secret nature to begin the deliberation of Agile methods and to develop and implement a strategy for moving forward with what is now a revolutionary movement in the Information Systems industry. Naturally, it would be hard to come by intimate research of the meeting, unless those in attendance agree to share commentary on the proceedings. The nature of the barriers and issues to the project centers on the ability to find adequate research and information about the members, the meeting, or the industry results of using Agile methodologies.

One of the main barriers to the research will be the availability of personal and professional information on all signatories of the Agile Manifesto, insight into their relationships prior to February 2001, and the professional histories that lead them to their beliefs and support of a new software development process. Initial research indicates that there has not been an abundance of supporting material published, and that personal contact and interviews will be required to advance the necessary research for the project.

Another barrier will be the proceedings of the meeting. It has historical significance, yet there is a limited amount of data available on the meeting agenda, topics of discussion, meeting format, and process for consensus. Again, it will require some primary research to acquire data related to, and analysis of how the Agile Manifesto was discussed, agreed to, and written.

Research questions investigated

The research focuses on investigation of three main questions. The first question concerns the meeting at Snowbird, Utah. The main questions associated with the meeting are related to how the meeting date and location were decided, how the participants were selected for invitation, who organized and conducted the meeting, how the agenda for the meeting was established, and what the purpose of the meeting was. This information is important because the meeting at Snowbird was a pivotal instance in the history of the Agile movement, since prior to the meeting, the movement was really evolving in disparate parts, and not in a unified, "one-voice" manner. The second question concerns the Agile Manifesto and how it was discussed, agreed upon, and written. As it turns out, this was the main deliverable of the meeting, and it has endured as the definitive statement for the Agile movement. Three days seems like a very short period of time in which to produce a document of this significance, so the investigation into how the document was actually produced is important to the understanding of how solid and cohesive the thinking is between the seventeen men who represented the various Agile disciplines at the Snowbird meeting. The third question concerns how the Agile movement, via the Agile Manifesto, is impacting the industry and Information Systems education. The intent of the parties was clearly to get together for the purpose of creating one voice within the industry for the practice of adaptive, light development methods.

The central problem being addressed is the impact of the Agile development movement on Information Systems educators and curricula. Conventionally, the application and system development life cycles taught in Information Systems curricula has followed the traditional, linear System Development Life Cycle models. The research presented here indicates that industry is adopting new development methods at an increasing rate. This has direct impact on Information Systems education.

Limitations of the study

The limitations to the research were primarily associated with access to the principals of the Agile Alliance. It was uncertain whether of not any of the participants of the Snowbird meeting would have time to respond to the questionnaire sent to them. If none of them had replied, then the research would have been severely limited to only published information about the meeting. Although not all of the participants of the Snowbird meeting were able to reply, significant responses were received from five of the seventeen members, including three "key" members: Jim Highsmith, Alistair Cockburn, and Kent Beck. The other major limitation to the research project is the lack of material published about the meeting, and about the impact of the meeting in industry. Since the meeting was relatively recent, and the Agile movement is just becoming known, published on, and adopted in academia, there is not a lot of available information about details of how the movement has evolved.

2. REVIEW OF THE LITERATURE AND RESEARCH

Historical overview of the theory and research literature

An alternative to the traditional methodologies has made its introduction and is being touted by many as the next generation of development methodologies and tools. They were initially referred to as "lightweight" methodologies, but that name gave way to the label "Agile" methodologies. There are fundamental differences in not only strategy, but also philosophy, between the two categories of methodologies. As smaller development companies tried to implement traditional methodologies and yet compete against larger companies in time to market, the traditional methodologies were found to be too slow and cumbersome, or "heavy" as the pseudonym implies. Initially, many companies tried to amend the heavy methodologies to more closely resemble what they knew they needed to do to be competitive in time to market. But it became more and more apparent that what was needed was a paradigm shift in the way development was taking place. The newer methodologies began to define themselves, and advocates and champions in the market place began to speak out. At the opportune time, thought leaders from the software development world emerged and collaborated to formally announce and define the Agile methodologies. This collaboration was in the form of a meeting where seventeen men, all thought leaders in the industry, gathered to compose the Agile Manifesto. From that point forward, the growth of agile methodologies has been the popular topic of conferences, articles, and books, and has shown evidence of growth, in tandem with the Unified Modeling Language, in standards based tools for software development.

The research literature specific to the topic

The seventeen men, proclaimed "organizational anarchists" (Fowler and Highsmith, 2001), who compose the Agile Alliance are driving the agile methodology movement. Their backgrounds are relevant to the research in that, as it turns out, they have the right combination of background, experience, and industry stature to implement a paradigm shift in the software development industry. The members of the Agile Alliance are profiled as follows:

Mike Beedle is the founder and Chief Executive Officer of a consulting company called e-Architects. The company works exclusively in application development using Internet technologies and distributed objects. As an early adopter, he principally uses the Scrum and XP methodologies where he works with companies to develop large-scale reusable architectures. Beedle publishes routinely in the area of agile methodologies and has co-authored a book with his colleague, Ken Schwaber, on the subject of Scrum methodology. He has more than twenty years of industry experience in the software development industry. *Arie van Bennekum* initially worked with Rapid Application Development technologies, but switched to the DSDM in 1997. He believes in delivering results to customers, and consequently is a believer in the agile methodologies out of this philosophy. One of his strong beliefs is in the group process. He enjoys facilitating and coaching, and is a certified trainer and consultant in DSDM. Arie lives in Europe and represented the European DSDM contingency at the meeting. *Alistair Cockburn* has exhaustively interviewed project teams. Working on project teams and conducting numerous interviews has led him to the conclusion that methodologies should be light (agile), sufficient enough to accomplish the goal, and self-evolving in nature. In the 1990s, Cockburn conducted research and work that led to the formation of the Crystal family of agile methodologies. He is co-authoring a book with Jim Highsmith about agile software development with numerous examples of success stories using the agile methodologies. *Ward Cunningham* founded Cunningham & Cunningham, a software development consulting organization. His background also includes hardware engineering and computer research, where he experimented with early practices in object-oriented programming. He has developed various design methods for software engineering teams, and has been instrumental in advancing the concept of Extreme Programming. *Martin Fowler* works for an application development and consulting company called Thoughtworks. He has over ten years of experience with object-oriented technology and programming, and OO methods for Information System design. Philosophically, he believes that processes should fit people, and not the reverse. *Jim Highsmith* was the principal architect for the Adaptive Software Development (ASD) methodology, and continues to author books, speak, and consult on Agile Methods. He has co-authored the article on the Agile Manifesto with Martin Fowler. Working with another colleague, Alistair Cockburn, Highsmith is combining ASD with the Crystal methodologies. He current project is a text on agile methods, due to be published in 2002. *Andrew Hunt* is the co-author of a best selling text on programming. He writes, speaks, and consults in the area of programming techniques and development methods, and has wide industry experience in telecommunications, banking, finance, medical, and Internet verticals. Hunt focuses on best practices in software development and is self-described as pragmatic in his approach. He is active in major organizations such as ACM and IEEE. *Ron Jeffries* is a leading consultant with a company called Object Mentor. He co-authored a book on Extreme Programming, and was the first coach for the XP methodology. He is expert in knowledge in the XP area, and frequently speaks at XP conferences and Internet groups. *Jon Kern* spent his early days in C++ programming and thinking about how to apply systems engineering and OO principles in the development process. He has been heavily influenced by the work of Peter Coad, who advocates "frequent, tangible, working results". He has co-authored books on Java design with Coad and Jeff De Luca. He continues to improve his software development beliefs with a combination of XP and FDD, and applying his method of incremental, iterative development. *Brian Marick* is a consultant specializing in software testing and programming. He has been involved for some time in a testing style that emphasizes exploration, decreased reliance on documentation, an increased acceptance of change, and the idea that a development project has an on-going conversation related to quality issues. Brian is interested in what the concept of "Agile Testing" might be, and is continuing to explore how it might fit into the overall concept of Agile Methodologies. *Robert C. Martin* has a long history of software development going back to 1970. He founded a company called Object Mentor that offers expertise in XP and agile processes. As an author, he is well published and is recognized as a leading expert in the field of the Agile Method's XP development concepts. *Ken Schwaber*, as president of Advanced Development Methods, is known as an experienced software developer and manager, and has also worked as an industry consultant. His are of expertise within the Agile Methodologies is Scrum, where over the past five years he has formalized the Scrum process. He has co-authored a book titled "Scrum, Agile Software Development" with a colleague, Mike Beedle. *Jeff Sutherland* works as the CTO for a startup company involved in wireless medical applications. With previous experience as the VP of Engineering or CTO in nine software development companies, he has a wealth of knowledge and experience in the development field. He is one of the original inventors of the Scrum methodology and continues his advancement of the concept in his current development efforts. *Dave Thomas* is a people centric philosopher when it comes to development teams. He believes that people, not process, are the cause of a successful development project. He is a published author and thought leader in the area of team methodology, and continues to promote the Agile Methodologies through industry speaking engagements and consulting. Even though he was not at the Snowbird meeting, he contributed to the drafting the twelve principles of the Agile Manifesto.

3. THE MAJOR AGILE ECOSYSTEMS

The following are the major ecosystems in the Agile development movement. *Scrum* is a term captured from the sport of Rugby. Ken Schwaber and Jeff Sutherland initially developed it, with contributions later on from Mike Beedle. As a project management framework, it organizes development into 30-day cycles called "Sprints". In this time frame, a certain set of features is scheduled as deliverables. One of the main practices in Scrum is the daily meeting of the development team for 15 minutes. The purpose of the short meeting is to ensure coordination and integration. This model has been in play for almost ten years now, and has been proven to successfully deliver software products on time. *Dynamic Systems Development Method*, or DSDM, was first used in the United Kingdom around the mid 1990s. Rapid Application Development (RAD) had a large part to play in the development of DSDM practices. One of the things that DSDM practitioners in Europe proclaim is that DSDM is fully supported with documentation and training materials. DSDM has nine principles that guide users through the development life cycle. The *Crystal* methods are best described as a "people-centric" method of development. Derived from a study of what actually works in development, rather than what people say should work in development, this method is all about the people aspects of development. Each member of the Crystal team is assigned work that is suitable for his or her talents, personality, and work abilities. Jeff De Luca and Peter Coad advanced *Feature-Driven Development*, or FDD. It is a minimalist approach with respect to process. As a five-step process that builds a project object model with associated feature list that drives the development process. Each step in the process is brief and concisely documented. The FDD process has a couple of key roles played by the chief architect and chief programmer. FDD has been used in large projects of more than 50 developers. *Lean Development* is the most strategy intensive of the ASDE family. Derived by Bob Charette (2002), it is based on the concepts of the Japanese Automotive industry. Characteristically, it calls for lean production resources, and management of risk by viewing it as an opportunity. The model has been used successfully in several large-scale telecommunications projects in Europe. *Extreme Programming*, developed by Kent Beck, Ward Cunningham, and Ron Jeffries, espouses the following values: courage, simplicity, feedback, and community. Technical excellence is achieved through refactoring and what they call "test-first" development. Extreme Programming, or XP, has developed a system of dynamic practices, and has been proven in many large-scale projects (Beck, 2000). It has generated the most interest of the ASDE family to date. *Adaptive Software Development*, developed by Jim Highsmith, focuses on how organizations embrace change. The view here is that change is a positive force, and if managed correctly and not avoided, it can result in positive results for the organization. It has standard practices associated with iterative development, feature-based planning, close customer involvement and collaboration with management. The *Rational Unified Process (RUP)* is specified and implemented by the Rational Software Company (recently purchased by IBM) and utilizes a suite of products to design, document, and implement an agile development methodology in concert with the Unified Modeling Language (UML). Grady Booch, one of the pioneers in UML and Agile methods is the Chief Scientist for the Rational Software Company.

Development of a significant event

In February of 2001, these seventeen men met in Utah to discuss the future of agile software development methodologies. The men comprised representation from all the various "light" methodologies: Scrum, Crystal, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Feature-Driven Development (FDD), Lean Development (LD) and, of course, Extreme Programming (XP). They went skiing, they relaxed, they talked, they ate, and after three days they wrote what is now known as the *Agile Manifesto*. This "agile alliance" made history, as the Agile Manifesto is now the definitive document for the statement of agreement and advancement of light software development methodologies. These men are the leaders on the new frontier of light methodologies, and the future of Information Systems development. Since that meeting, there has been much research and debate over the viability and impact of Agile methodologies. Its history is still being written.

The meeting was held from February 11 to 13, 2001 at a resort called the Lodge at Snowbird. This location in the Wasatch Mountains of Utah was selected because of its central location, easy access, and facilities for recreation. The precursor to this meeting was a meeting in the spring of 2000 in Oregon. Kent Beck organized the meeting, principally as a meeting to discuss Extreme Programming (an XP leadership meeting), but also allowing other people who were sympathetic to the cause of Extreme Programming to attend. These "sympathizers" were supporters of other "light" methodologies. At this point in time the use of the term "light" in reference to the methodologies was still in play, but soon after, the term "agile" was substituted as the designator for this category of methodologies. Alistair Cockburn was instrumental in beginning to use the word "agile" as opposed to "light". Following the Oregon meeting, Robert Martin suggested via a memo that went out in September of 2000 a meeting in Chicago to gather all of the "light" methodology champions in one room to discuss evolution of the movement on a larger scale. This was right after the OOPSLA meeting in October. The meeting was to be held in the January to February of 2001 timeframe. Chicago in the winter was not deemed to be the most attractive location. Two alternative suggestions were made, one in favor of Snowbird, Utah and the other in favor of Anguilla, in the Caribbean. It is unclear who made those suggestions. Although the Caribbean was warm and had fun things to do, it was decided that the logistics for that location did not work, and the selection of Snowbird, Utah, with the opportunity to ski and a central, easy access location, was made with the urging of Alistair Cockburn and Jim Highsmith. So with this first group decision over, plans were made to go to the Lodge at Snowbird in Utah from February 11 to the 13th. Alistair Cockburn made the hotel arrangements and did the majority of the organizing.

Most of the members of the meeting knew each other from OOPSLA, Kent Beck's meeting, the patterns movement, Smalltalk groups, or from other XP activities. Ari Van Bennekum was a new face to everyone. He had made the trip from Europe to represent the international DSDM perspective. There was no formal, pre-set agenda for the meeting. On the first day, a rudimentary agenda was put together. The main event seemed to be to try to define and agree on some common goals, but the idea of writing a "manifesto" began to evolve. As previously mentioned, the reference to "light" methodologies was troublesome to many in the group, so coming up with an appropriate name was high on the agenda. Prior to the meeting, everyone had posted on the Wiki a list of his preferred topics. Then at the meeting, after introductions, everyone used 3x5 index cards to write topics for discussion, and they were compiled in the center of the room. Someone in the group took on the task of taping them to the wall in some proposed order of discussion. Some of the key items of focus were 1) definition of what each member stood for; 2) finding a better name than "light"; and 3) finding the synergy in the group. Throughout the meeting the wall was used to tape up new agenda items that came up in discussion. Several of the members had been advocating adaptive/light methods for years, so they saw this as an opportunity to strengthen their voice by banding together. Some saw it as an opportunity for good discussion with their peers, or just an opportunity to find out more about what people were up to. The group considered themselves to be knowledgeable leaders in their field, innovative thinkers, and the right group to accomplish the goals at hand. There was some thought that the group did not represent enough diversity, or that the meeting should have included two other industry leaders: Grady Booch (Chief Scientist for the Rational Software Corporation), Tom DeMarco and Dave Thomas. But the consensus among respondents to the survey was that the group shared a core philosophy, even with apparent differences in detail. By way of differences, there were comments that Steve Mellor seemed apart from the group, or had a background and perspective that was very different. The new name and core values were agreed upon very quickly in the meeting, but the twelve principles took much longer to reach agreement. After that, agreement started to break down, but the core values and subsequent twelve principles were clearly articulated and agreed upon. Most respondents to the questionnaire felt that the Agile movement is definitely impacting the software development industry. Some of the viewpoints of the group are that mind share in the industry is increasing, publication on the topic is increasing, and industry interest and adoption is increasing. Alistair Cockburn's view of the how the Agile movement is impacting industry is that it is offering a "counterforce" to the SEI and CMM that didn't exist prior. Jim Highsmith considers that the Agile movement is in the early adopter stage at this point. There is recent evidence from the Giga Group to suggest that industry adoption of Agile methods will increase over the next 18 months. Some thought was given to the role that colleges and universities might play in advancing the Agile movement by teaching the concepts in their information technology and software engineering programs.

The respondents to the survey indicated that they felt that finding common ground has strengthened the movement, and naming and clarifying what are essentially a distinction between predictive and adaptive approaches, and an exploration of all the various ways that Agile methods may be practiced have been lessons learned since the meeting at Snowbird. Most of the respondents felt that there was no professional risk associated with being involved with the meeting. There was a sense that the future of the Agile movement will depend, in part, on successfully moving past the early adopter to early majority stage by providing tools, support services, and seeing some success stories. The Agile Manifesto was the main deliverable of the Snowbird meeting, and the process of developing and drafting the document is rather lack luster. Apparently, it was a collaborative process where the four value statements (core philosophy) were developed at the Snowbird meeting, but the twelve principles were subsequently written via back and forth emails, over the following three months. Jim Highsmith played a lead role in starting the process by making notes on the board. Martin Fowler began to help wordsmith, and coming in from a break, the rest of the group began to join in the dialogue. So the Manifesto was written in an ad-hoc manner over about an hour's time. Some group members moved in and out of the room, while others stayed for the entire time. When the draft was complete, the group stared at it, and discussed it line by line. This process continued until everyone expressed comfort and said they could support the document as written.

The outcome of the Snowbird meeting was the signing of the Agile Manifesto, or formally, The Manifesto for Agile Software Development. This symbolic document was the result of discussion and collaboration on how they could uncover better ways to develop software and help others understand how to do it as well. Each participant relayed stories from their own experiences about how the "heavy" methodologies had been inadequate or ineffective in the past, and how the need for rapid business change and response calls for new ways of thinking and doing development activities. The participants focused on the values that they had in common. Each participant agreed that there was value in processes and tools, comprehensive documentation, contract negotiation, and following a plan. But more importantly, what they valued was individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. The operative word is "more", meaning that even though they valued processes and tools, comprehensive documentation, contract negotiation, and following a plan; what they valued *more* was individuals and interactions, working software, customer collaboration, and responding to change. This approach worked well in that it formed the basis for detailed discussions and consensus about how to support the various methodologies represented beneath one umbrella. This is a very different perspective from what sometimes gets associated with "Agilists"; that is, the notion that they are anti-methodology. The Agile movement is not anti-methodology at all, but one that wants to achieve balance in the methods (Highsmith, 2002). Finding a balance and being able to facilitate a "change" mentality in the organization seems to be the underlying foundation of the Agile movement. So the Agile Manifesto was drafted based on four underlying principles (Alliance, 2001).

The contribution this study makes to the field of Information Systems Education

This study contributes to the body of knowledge in the field of system and software development by offering insight into what is essentially a paradigm shift in development methodologies, a study of the motivations behind the movement, and conclusions about the future direction and efficacy of the Agile movement. By investigating questions related to the

Snowbird meeting, its participants, and the development and impact of the meeting and its main deliverable, the Agile Manifesto, and providing a comprehensive report, the foundation of the movement is captured in a snapshot that will allow future researchers to better understand the catalysts for such movements, and the genesis activities with such movements.

4. METHODOLOGY

Tools to be used for this report

The principal research tool to be used is a questionnaire. The questionnaire was composed of 15 open ended questions about the formation of the Agile Alliance, the proceedings of the three day meeting, and the opinions of the thought leaders about the current and future impact of the Agile Methodology movement in the software development industry.

Results of data analysis

The most relevant data attained was from the respondents to the survey. These included Jim Highsmith, Ron Jeffries, Kent Beck, Steve Mellor, and Alistair Cockburn. The chronological analysis of the data revealed the following facts about the meeting in Snowbird, the formation of the Agile Alliance group, the writing of the Agile Manifesto, and the group's thoughts on the future of the Agile movement.

Kent Beck originally organized a meeting in Oregon for leaders in the XP field and invited others who had an interest in adaptive methodologies to attend. This was a "pre-cursor" to the Snowbird meeting.

Subsequent to that meeting and the October OOPSLA meeting, Bob Martin sent out an email suggesting a meeting of all interested parties in adaptive methodologies. Alistair Cockburn had planned to organize a similar meeting in Snowbird in February, so he threw support to Bob Martin's meeting idea.
Togethersoft, the U.S. DSDM Consortium, Alistair Cockburn's company Humans and Technology, and Jim Highsmith sponsored the Snowbird meeting.

Alistair Cockburn and Jim Highsmith did the majority of logistics work for the meeting. Most of the participants in the meeting knew each other previously from professional conferences, activities, Smalltalk groups, and the XP community. Ari van Bennekum, representing the European DSDM community, was an unknown participant.
The agenda for the meeting was developed the first day. There was no agenda going into the meeting. The agenda was developed by using index cards to state topics of discussion, and then by organizing them on a wall. Preliminary agenda topics were posted to the Wiki prior to the meeting to "float" topic ideas. High on the list of important items were finding a new, acceptable name for the movement, and developing a core statement of values for the group.

Participation in the meeting was principally out of professional interest, or to create a stronger voice in the software industry to advocate the adaptive/light methodologies.
The perceived strengths of the group were that they were very knowledgeable and experienced industry leaders, they had a commitment to improving software development, and the full spectrum of "agile" methodologies was represented at the meeting.

The perceived weaknesses of the group were that they mostly represented small, two-person companies, they were only represented by a white male population, and that Tom DeMarco, Dave Thomas, and Grady Booch should have been at the meeting. Everyone mostly had the same core philosophies or values, and that top down agreement came quickly with respect to the name and the core values of the group. The range of thought about how the meeting is impacting industry today is quite broad. Some view a definite impact, cited by increasing publications, industry buzz, and corporate adoptions. Some view it as minimal, and more industry driven. The future direction of the Agile movement is somewhat diverse in prediction. Some feel it will be an uncertain, an up-hill battle for corporate adoption, while others cite research from the Giga Group predicting very near-term positive outcomes for the Agile movement. Some notation was made about the potential boost to the Agile movement from having it adopted in college and university curricula.

Major lessons learned since the meeting include the impact of naming and defining the movement, clarifying the difference between predictive and adaptive approaches to software development, and the publication of the various ways in which agile methodologies can be implemented. Most participants felt no professional risk by associating with the meeting and signing the Agile Manifesto. The success of the Agile movement will be dependent on it moving from the early adopter stage to the early majority stage, more adoption of the practices, changing the conversation among people who converse on the subject, and by educating corporate management about the alternative to traditional methods. The Agile Manifesto was developed and drafted in a very collaborative, ad-hoc manner. The session was informal, and Jim Highsmith and Martin Fowler helped facilitate the discussion. The four value statements were developed at Snowbird, while the twelve principles were developed and published in the subsequent three months to the meeting. The naming and core values were described as easy, while the development of the twelve principles was described as horrible and painful. XP is considered to be the best known and most popular of the methodologies. The popularity of XP gave momentum to the meeting. The Agile movement focuses on a balance between planning and executing. There is a noticeable level of pessimism about the future of the Agile movement among some respondents to the survey. There are still many questions that are unanswered about the Agile methodologies that can only be answered on down the road.

What brought these men together at this specific time and place?

Most of the participants of the meeting have been working in the industry for many years, and have been promoting the adaptive or light methodologies for years. Over the years, they had seen each other at conferences, and most of them were well published in their respective areas. The XP methodology has been gaining in notoriety for some time, both within industry, and within the research and investigative communities. Publishing on the subject of XP has increased over the past decade as well, both by advocates for the Agile movement and critics of the Agile movement. In a January 2002 article in Software Development by Dr. Barry Boehm (2002), he says, "Real-world examples argue for and against agile methods." XP is being used in large, well-known companies such as Symantec, highlighted in a January 2002 article by Alexandra Weber Morales (2002). Arthur English (2002) points out that XP is "Known by the acronym XP well before Microsoft began using if for Office XP and Windows XP, the development methodology has caused as much of a stir as Java, .Net, XML and Web Services." Kent Beck's work in developing XP in 1996 at the Chrysler Comprehensive Compensation project has really been the catalyst for the Agile movement. So with the momentum of the XP methodology, the upstart development of similar "non plan oriented" methodologies, the leadership and organizational abilities of Jim Highsmith and Alistair Cockburn, and the professional acquaintance of industry leaders and innovators in adaptive methodologies, the concept of the Snowbird meeting was a natural progression from Kent Beck's meeting in Oregon. Jim Highsmith met several of the members at Kent Beck's meeting, Ron Jeffries knew most of the men from being active in the XP/agile community for several years, and Kent Beck knew most of the men from OOPSLA and XP activities. He had not met Jeff Sutherland or Ari van Bennekum. Steve Mellor knew Bob Martin and Martin Fowler, and some of the men (Kent Beck, Alistair Cockburn, etc.) by name and reputation only. And Alistair Cockburn knew most of the men through conferences, the patterns movement, and Smalltalk. Ken Schwaber and Jeff Sutherland did not know the other men, Mike Beedle was generally known through the patterns movement, and John Kern was generally new to everyone. So the data indicates that the answer to the research question of what brought these men together at this specific time and place, is answered by four principle things:

There was momentum to move forward from the Oregon meeting sponsored by Kent Beck, and attended by many of the Agile Alliance members.

There was strong leadership and organizational skills applied by Jim Highsmith, Alistair Cockburn, and Kent Beck.

Using their industry contacts, this group sought to invite additional members to have full representation from all of the agile methodologies.

Snowbird, Utah was a central location with good facilities for meeting, skiing, and relaxing.

How was the Agile Manifesto discussed, agreed upon, and written?

The Agile Manifesto is based on four core values. The Agile Alliance agreed that they value:

Individual and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

Agreement on these core values came very quickly and naturally to the men at the Snowbird meeting. These are consistent with the values touted by the XP followers, and other adaptive methodology advocates. So there was really no new territory to discuss, just time needed to formulate the best way to articulate the values. The Agile Manifesto was drafted at the meeting in the form of the four core values, but the twelve defining principles of the Manifesto came later. At the end of the meeting, the principle outcome was that everyone agreed to the core values, and would make the effort to continue the work to define the principles of the Agile movement. The Manifesto has been referred to as their "rallying cry" (Boehm, 2002). This connotation seems harsh, in that the manner in which the Manifesto was developed and written, and ultimately presented, does not depict irrationality or inciting behavior, but rather a codification of existing thoughts on the practice of adaptive, or agile, methodologies. So with that in hand, the members of the Agile Alliance worked for the next three months, via the passing of emails, to develop and properly articulate the guiding principles behind the Agile movement. The principles are twelve statements that further support the four core values. The Manifesto was written in a completely ad-hoc manner during the meeting in Snowbird. The twelve principles were begun in a similar fashion at the meeting, but time did not allow for substantive completion of the work. There was disagreement about the meanings of the words being used to define daily practices. So the follow-up work that was done via email was slow and much less effective because people were less engaged, were busy again with their occupations, and generally had less energy to devote. The final product was posted to the web, where additional signatories to the Manifesto continue to grow daily. So is the Manifesto terminal in its current form? Yes, it is, in the sense that it is not viewed as a precursor to more consolidation of practice among the Agile methodologies. When asked whether of not the Manifesto was a precursor to a "Unified Lightweight Methodology", the Alliance responds "Absolutely not! While the group believes that a set of common purposes and principles will benefit the users of agile methodologies, we are equally adamant that variety and diversity of practices are necessary (Fowler and Highsmith, 2001).

How are the Agile methodologies impacting industry and Information Systems education?

The Agile movement is having a clear impact on industry and education: people are talking about it. Discussion now includes the Agile methodologies, their validity, use, and value in developing software. Large organizations, such as the Rational Software Corporation, are adopting Agile processes in their suite of products. Information Technology research organizations, such as the Cutter Consortium, the Giga Group, and Gartner are publishing articles. Articles in Software Development, CIO, Computerworld, and IEEE have been published, and major industry conferences, such as OOPSLA, and the SEI's SEPG all have Agile methodology presentations, panels and tutorials. The Agile methodologies are principally changing industry and education by creating conversation about change and new ways of doing things. Evidence of impact is apparent in the fact the long term system development industry leaders such Barry Boehm (2002) feel compelled to comment about the Agile methods. And the Agile methods are getting a lot of mindshare. Not just small development shops, but larger corporate Information Systems development organizations are finding the need to shorten development cycles, and one of the best ways to do it is by adopting Agile methods. Project managers sometimes find it tough to make the transition from traditional methodologies, like the Waterfall model, to the iterative life cycle models (Krutchen, 2000). The Rational Software Corporation (now a part of IBM) with its RUP may very well be the next major Agile method to gain prominence, right behind XP (Smith, 2001). The majority of Information Systems curricula teach object-oriented development methods in the front end development of database system applications. The backend, however, remains generally relational in design. This creates an "impedance mismatch" between the front end and back end of systems development. The implementation of the SQL3 standard offers today's DBMS engines the opportunity to work with "objects" in the database, and the movement toward object-relational design is supported by the Agile methodology framework associated with the UML and such industry leading Integrated Development Environments (IDEs) and development suites as the Rational Unified Process.

Despite the impact the Agile movement is having on Information Systems development in both industry and education, it should be noted that there are still many unanswered questions. The long-term effects of common ownership of code are still in question, how less documentation affects the maintainability of code is still in question, and the impact of culture on the Agile methods are still yet unanswered. So the main impact on industry and education by the Agile methods, at this point, is generated from the conversation that has been stimulated by the advancement of the awareness of the Agile methods by the Agile Alliance, and the object-relational development being done using the Agile methodologies. The drafting of the Agile Manifesto was a first step in articulating a vision and a position for an Information Systems development concept that is essentially a paradigm shift for the industry and for education in systems analysis and design.

5. CONCLUSIONS AND IMPLICATIONS FOR INFORMATION SYSTEMS EDUCATION

Conclusions

This research leads to several conclusions regarding how seventeen men met to articulate an industry statement for the new, Agile methods of Information Systems development, how the Agile Manifesto came about as a document, and how the meeting in Snowbird is impacting the industry and Information Systems education. In the area of how these particular men came together, it is a combination of factors that caused it. The history and development of Extreme Programming had a lot to do with the initial momentum of the Agile movement. Kent Beck, along with Ward Cunningham and Ron Jeffries, originated the concept of Extreme Programming, and perfected it and documented it over the past ten years. The Extreme Programming movement has gained a following, and has a history. Associated with the Extreme Programming movement are the industry related articles, publications, conferences, and events, that have helped to advance the cause of a heretofore unnamed family of "non-traditional" development methodologies. As the inherent weaknesses of the traditional, plan-based methodologies caused problems for organizations competing in an environment of rapid change, and with necessitated quick time to market in order to remain competitive, the strengths of adaptive, or "light" methodologies began to gain even more attention. Through various software development conferences and events, such as OOPSLA, most of these men had made acquaintance, and had gained a healthy professional respect for one another. Kent Beck had sponsored an XP Leadership meeting in Oregon in 2000 where the idea of having a general meeting of all interested parties in the newer, lighter methodologies was discussed. However, it was Bob Martin who sent out the initial email in September of 2000, that really started things moving in the direction of having a general meeting for the newer, lighter methodologies. Martin's idea was to have the meeting in Chicago, and concurrently Alistair Cockburn was thinking about having a similar meeting in Utah. So the debate over location ensued, and rather than have a Chicago meeting in the winter where it was very cold, or fly all the way to the Caribbean where it was out of the way, the choice of Utah was selected because it was convenient, provided good accommodations and recreation, and Alistair Cockburn was in the area and could help co-ordinate travel and lodging planning. Jim Highsmith was instrumental in the process of organizing the Snowbird meeting, along with Alistair Cockburn, and to ensure that equal representation among the various "practices" of the newer, lighter methodologies was at the meeting, invitations went out to other industry leaders from the Scrum, FDD, DSDM, and LD practices. Of course, Alistair Cockburn represented the Crystal methods point of view, and Jim Highsmith represented the ASD point of view.

So on February 11th everyone met at the Snowbird Lodge in the Wasatch Mountains in Utah, and after introductions, history began. From what was a developing agenda, one of the first orders of business was to find a name that adequately represented the movement. Alistair Cockburn was chief among those who thought that the use of the word "light" to describe the methodologies was a misnomer, and did not adequately represent the movement. So the suggestion of "agile" was adopted, and the collective family of adaptive, lighter methodologies now had a name. So the Agile Alliance is formed, and the discussion over the next couple of days centered on what everyone thought their practices stood for, and what common ground was there for articulating the Agile values, and ultimately the underlying principles that separated their beliefs about software development from those in the traditional camp. A strong theme emerges in how the values are articulated. It is made abundantly clear that the value of processes, tools, comprehensive documentation, contract negotiation, and plans are accepted and noted by the movement, but what is valued *more* is individuals, interactions, working software, customer collaboration, and responding to change. A lot of the movement's core values are explained by the search for finding balance and by embracing change and using it as a positive force.

The Agile Manifesto was written in an ad-hoc manner with Jim Highsmith (largely) facilitating and Martin Fowler helping as wordsmith. Once a final draft was achieved, everyone looked at it and made comments to fine tune the four core value statements, and talk about how they might relate to underlying principles. Given the manner in which the agenda was constructed and the meeting was conducted, it is an amazing piece of work product to emerge from a three-day meeting of men who had a relatively fragmented relationship through industry conferences and meetings. Their ability to find common ground in these four core values is a testimony to the strength of the agile concept. The formulation period for the movement has actually been over many years, but the organization and "voice" of the movement came into formality in February of 2001.

Another conclusion is that some of the members of the meeting do not seem to believe that the Agile movement will move to a position of national or global prominence in the world of software development. Some of the members expressed a "wait and see" attitude about the promulgation of the Agile movement. And some members see only a bright future of advancement and acceptance for the Agile movement. There is ample evidence that the movement is being challenged, accepted, studied, and discussed. In a field of fast changing practices, technologies, and concepts, the advancements in the Agile movement are a partial success. There are many questions still in search of answers that can only be provided with time and experience.

This study also concludes that Information Systems educators must respond to the emergence of the Agile methodologies in the Information Systems curricula. Layered on top of the design distinctions between object-oriented design and relational design is the change in development methodology. This study further concludes that the Agile methodologies have been successful in major projects since the early 1990s, and not only small to mid-cap companies, but large-cap companies are adopting Agile methods in their software and systems engineering organizations. The Agile development movement has massive momentum in industry and will undoubtedly be a major force in the future shaping of Information Systems organizations. A final conclusion of the study is that relatively few

Information Systems education programs are teaching Agile methodologies, or are looking at how to integrate the topic into existing program courses. Over the next few years this paradigm shift will complete at the industry level, and will become the standard for Information System development.

Implications for Information Systems Education

The Agile development methodologies are real, they are advancing in concept, and are increasing in adoption. Educators in Information Systems curricula must find innovative ways to teach students that there are two worlds when it comes to development methodology. The traditional, linear development life cycle models that have long been associated with projects involving back-end relational databases are now challenged by the emerging development methodologies with completely different foundations, philosophies, and concepts. Industry currently deals with the problem of different development methods being used to develop front-end applications and the back-end database. The disparate worlds are coming together, and at the center of the collision is Agile development methodologies.

Information Systems education must constantly change to reflect advances in technology, architectures, standards, and methodologies. The "impedance mismatch" between front-end application development and back-end database system design is a prime example of how Information Systems education must change how it has conventionally taught. Object –oriented technology now must work in concert with relational technology, and advances in database management systems lead to object-relational design. A major implication for Information Systems education is that the "systems" view is imperative as modeling tools and design methodologies integrate to develop both the front-end and back-end as a unified view. The Agile methodologies have impacted front-end design and system engineering methods by providing the capability to eliminate the impedance mismatch between two disparate design methodologies (object-oriented and relational). The Rational Unified Process is a good example of the use of the next generation of modeling tools that employ Agile methods. Based on the Unified Modeling Language, this suite of products utilizes object-relational modeling of a database system and application while working within the Agile methodology framework. Information System educators must teach students how to work in two worlds: the conventional relational design world utilizing traditional SDLC methods, and the object-oriented world utilizing Agile development methods.

Information Systems educators must be cognizant of the acceptance and adoption of the Agile methodologies. They represent a fundamental paradigm shift in how systems analysis and design is taught. In order to prepare Information Systems program students for the industry changes resulting from the Agile methodologies, Information Systems educators will need to understand the origins, significance and efficacy of the Agile development movement and revise programs of instruction accordingly. Further study of the Agile movement should be in the direction of resolving issues with cultural differences adopting Agile methods. Do Agile methodologies universally apply among different cultures? The impact of Agile methodologies on the ownership and maintainability of code needs to be studied. The major implication of this report for Information Systems education is that more questions exist and future study is required to fully investigate and evaluate the future impact of the Agile movement on Information Systems curricula. The new teaching paradigm for Information Systems in the area of systems analysis and design with respect to the Agile development movement is probably best expressed by Margaret Mead (1960) when she said "Never doubt that a small, thoughtful group of people can change the world. Indeed, it is the only thing that ever has."

6. REFERENCES

Alliance, A. (2001). Principles behind the agile manifesto. Retrieved July 3, 2002, http://agilemanifesto.org/principles.html

Beck, K. (2002). Extreme programming explained: Embrace change. Boston: Addison-Wesley.

Boehm, B. (1986). A spiral model of software development and enhancement. IEEE Computer, 21(5), 61-72.

Boehm, B. (2002). Get ready for agile methods, with care. IEEE Computer, 35, 64-69.

Charette, R. (2002). Agile methodologies: The great debate. Arlington, MA: Cutter Consortium.

English, A. (2002). Extreme programming. Strategies, 19.

Flowler, M. and J. Highsmith (2002). The agile manifesto. Software Development.

Gladden, G.R. (1982). Stop the life-cycle, I want to get off. ACM SIGSOFT Software Engineering Notes, 7(2), 35-39.

Highsmith, J. (2002). Agile software development ecosystems. Boston, MA: Addison-Wesley.

Krutchen, P. (2000). From waterfall to iterative lifecycle –A tough transition for project managers. Rational Software Corporation.

Mead, M. (1960). The golden age of american anthropology. George Braziller Publishers.

Morales, A.W. (2002). Embracing change. Software Development.

Royce, W.W. (1970). Managing the development of large software systems. Proceedings of IEEE WESON (August).

Smith, J. (2001). A comparison of RUP and XP. Rational Software Corporation.