Journal of Information Systems Education, Vol. 16(4)

Agile Methodology Adoption Decisions: An Innovative Approach to Teaching and Learning

John McAvoy
David Sammon
Business Information Systems,
University College Cork,
Cork, Ireland
j.mcavoy@ucc.ie dsammon@afis.ucc.ie

ABSTRACT

This paper documents an innovative approach to teaching and learning in studying the area of agile software development methods. The observations and analysis of a Critical Adoption Factors workshop for agile methodologies are presented, where 'Phase One' of the workshop was undertaken in an academic context and 'Phase Two' was carried out in an industry context to validate the adoption assessment matrix developed in 'Phase One', and further refine classroom material. Overall, the innovative and unique contribution of this study is centred on the outputs of the Critical Adoption Factors workshops, in an effort to improve students understanding of the constituent parts of an agile methodology. Although this paper documents the initial stages of an ongoing research study, the early observations are encouraging for teaching and learning in the academic context.

Keywords: Active learning, teaching with cases, critical adoption factors, agile methodology

1. INTRODUCTION

Discussions on agile software development methodologies, in both academia and industry, have a tendency to develop into an argument between proponents of agile methods and proponents of more traditional process oriented methodologies. Bearing this in mind, the decision to adopt a particular software development methodology is a difficult one, and the decision to choose an agile method is no exception. In theory, as in practice, definitions and descriptions of the various agile methods are presented, yet the factors in the decision to adopt, or not adopt, an agile method are not addressed.

"*Agile, for a software organisation, is the ability to adapt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment*" (Kruchen, 2001, p 27). Within this research study, our pedagogical approach is influenced by this required adoption to change in today's software production environment. This contention is also supported by Lamnias (2002, p 25) who states that "*the modernization and rapid changes that have taken place within the field of production (new forces and techniques of production, new management techniques, etc.) demand a new complexity within the field of education.*" Therefore, taking on board the contention of Kruchen (2001), the purpose of this paper is to develop a model for student learning that improves a student's knowledge of the adoption of agile methods. It identifies the impact of using active learning techniques (including: Teaching with Cases; Group Problem Solving; and Brainstorming) in an effort to highlight the importance of certain adoption factors to the adoption of an agile method, and illustrate the usefulness of a decision support process (adoption assessment matrix) to determine the viability of an agile method for a specific software project. Therefore, Section 2 presents our understanding of the agile methodology, and Section 3 outlines the research approach, highlighting how these critical adoption factors are used to determine the viability of adopting an agile method from a pedagogical perspective. Finally, our initial observations are presented with conclusions drawn from this preliminary research study. Recommendations for future research plans are also considered.

2. AGILE SOFTWARE DEVELOPMENT METHODOLOGY

The agile software development method does not exist, it is instead a collection of methodologies with common core values, where examples of agile approaches include: Extreme Programming (XP); Crystal Methods; SCRUM[1]; Dynamic Systems Development Method (DSDM); Feature Driven Development (FDD); and Adaptive Software

[1] SCRUM is not an abbreviation, it is a rugby term used to depict a strategy meeting which gets an out of play ball, back into play.

Development (ASD) (Highsmith, 2001; Sutherland, 2001). Interestingly, Rainwater (2002) does not categorise extreme programming as an agile method, and is not alone in this view, as a major conference in Chicago, in 2002, was called the XP/Agile Universe, implying a differentiation between the two. However, Rainwater (2002) does agree that there are many similarities between XP and agile and shows how the founders of XP were among the founders of the agile movement. Furthermore, Stapleton (2003) proposes that DSDM and XP are complimentary approaches, where the deficiencies in each approach are countered by the benefits of the other. XP concentrates on the programming techniques, while DSDM concentrates on the complete lifecycle.

The basic principles of agile methodologies are qualified in Abrahamsson *et al.* (2002) as:

- Individuals are more important than processes and tools;
- Working software is more important than comprehensive documentation;
- Customer collaboration is more important than contract negotiation;
- Responding to change is more important than following a plan.

These principles are referred to as the '*Agile Manifesto*'. Therefore, agile methods are a response to the inability of traditional methods to embrace change in a turbulent business environment, that demands software to meet its needs quickly (Highsmith and Cockburn, 2001), and Rising and Janoff (2000, p3) describe it as the need to "*meet customer needs and turn this chaos to our advantage.*"

Glass (2001) describes the debate between proponents of traditional development approaches and proponents of the newer agile approaches. However, neither approach is correct in all circumstances, and the 'best-fit' needs to be determined for a given circumstance. As is the case in many aspects of the industrial reality, there are no silver bullets in software development (Brooks, 1987; Jeffries, 2001). For example, DSDM comes closest to determining the viability of an agile development method, and in a publication of the DSDM Consortium, guidelines for introducing DSDM are discussed, with a limited look at a feasibility study (DSDM, 1998). Within this publication, it is recommended that the organization has the right culture, but it is not determined how to evaluate or quantify this. Furthermore, the steps in the feasibility study that follow the recommendation, involve educating a key stakeholder, and producing a strategy and plan. Therefore, it is clear, based on these recommendations, that the choice has already been made to adopt DSDM, so it is not a decision whether to adopt an agile approach or not.

Williams *et al.* (2000) state that there are a variety of software projects and a variety of software processes to match the projects' goals and objectives. While further, Williams *et al.* (2000); Ramachandran and Shukla (2002); and Olson and Stimmel (2002) introduce pair programming, an agile technique, as bringing great benefit to software development. However, on examination of these articles, there is no discussion on what defines a project as being suitable for this agile technique, as opposed to any of the plethora of other software development techniques, processes, or methodologies. Therefore, the debate concerning agile methodologies has predominantly been based around whether it was a better choice than traditional development methods (De Marco and Boehm, 2002), rather than a debate on the appropriateness of an agile approach for a given company, team, or project.

2.1 Critical Adoption Factors in Adopting an Agile Methodology

Throughout the available literature on agile methodologies, critical success factors are discussed, yet the term is not used to determine the viability of adopting an agile approach. The agile manifesto, described in Abrahamsson *et al.* (2002), is a list of aspirations or ideals, and as such is not readily quantifiable as requirements. As a result, having conducted a preliminary literature review for the purpose of this research, and based on the industry experience of one of the researchers involved (in the adoption of agile methods in software projects), we present eleven factors, and their descriptions, that could be regarded as critical adoption factors in attempting to assess the suitability of a software project to the adoption of an agile methodology, as illustrated in Table 1.

On further examination of these critical adoption factors we propose a further classification of four groups: Project; Team; Customer; and Organization; into which each of these factors can be suitably placed, as illustrated in Table 2. The researchers believed that it would be easier to provide a detailed description of these four groups, as provided below, rather than each individual critical adoption factor, specifically taking into account the time constraints associated with a workshop format.

2.1.1 Project: The project grouping describes the relevance of critical adoption factors when considering the mechanics of the project being undertaken. Issues of importance that can impact on the appropriateness of adopting agile methods include: the duration of the project; increasing uncertainty and changing requirements within the project; and the criticality of the project. Highsmith, (2001, p 260) states that "*today's project manager must deliver concrete results in shorter time frames while being constantly bombarded with myriad changes and risk laden decisions*". This concern aligned with Kruchen's (2001) criteria of reacting quickly to market, impose a shorter time frame on agile projects. A core value of agile development is the acceptance of change, usually visible through changing requirements. Highsmith (2002) proposes that as the uncertainty associated with a project increases, the suitability of the agile approach will also increase. Kirkpatrick *et al.* (1992) describes the changing of requirements (requirements volatility) as one of the major sources of risk in a project. However, this directly contradicts the agile viewpoint, while stories (requirements definitions in the agile methodology) continue to be written throughout the project. This is regarded as one of the basic edicts of the agile approach (Beck and Fowler, 2001). Burns and Stalker (1961) found that firms could be categorized into two distinct management types, which were the best fit for their working environment. Mechanized systems are

	Critical Adoption Factors	Description	References
1	Duration of the project	The timeframe for the project should be short for an agile project	Highsmith (2001) Kruchen (2001)
2	Location of the customer	It is preferable that customers are available throughout the project, therefore an in-house customer would be ideal	Young (2003) Jeffries (2001)
3	Customer Involvement	Customer involvement is vital for the success of a project	Beck and Fowler (2001) Young (2003)
4	Acceptance of change (to requirements)	Agile methods are specifically aimed at projects subject to continual change	Highsmith (2002)
5	Team Size	Agile methodologies emphasise the importance of teams, recommending smaller team sizes	Boehm (2002) Rising and Janoff (2000)
6	Skill level of team	Highly skilled developers are required	Reifer (2002) Levine *et al.* (2002) Lindvall *et al.* (2002)
7	Organizational and reporting structure	An organic structure is required for an agile approach	Boehm (2002) Cockburn and Highsmith (2001) Dolan et al (2003)
8	Process	Processes, such as CMM or TL900, are regarded by many as stifling, even stopping, agile methodologies. At the very least, they can be in conflict	Martin (2003) Greening (2001)
9	Documentation requirements	While agile methodologies do not prohibit documentation, it should be kept to a minimum	Greening (2001) Olson and Stimmel (2002)
10	Layout of workspace	Open planned offices, with shared areas, are required to promote communication and team work	Poole and Huisman (2001) Kalita (2003)
11	Criticality of project	Some projects are critical to life, e.g. air traffic control systems, and they are not suitable candidates for agile approaches	Emery (2001)

Table 1: Critical Adoption Factors for an Agile Methodology

Group	Critical Adoption Factors
Project	(1) Duration of the project (4) Acceptance of change (11) Criticality of project
Team	(5) Team size (6) Skill level of team
Customer	(2) Location of customer (3) Customer involvement
Organization	(7) Organizational and reporting structure (8) Process (9) Documentation requirements (10) Layout of the workspace

Table 2: Grouping Classification of Critical Adoption Factors

appropriate for stable environments, while organic systems are pertinent to conditions of change and flux. Although this categorization was proposed years before the advent of agile methods, projects using agile methodologies show a close correlation with the definition of organic systems. Finally, agile techniques, such as ASD, are unsuitable for critical systems, such as air traffic control software. However, this is not a major concern as the majority of systems do not have this degree of criticality systems do not have this degree of criticality (Emery, 2001).

2.1.2 Team: The team grouping highlights the importance of the project team, in terms of team size and skill level of the team, for adopting agile methods. While the team size is important, it is not of vital importance. Boehm (2002) describes how agile development is optimal when used in small teams. Rising and Janoff (2000) quantify ten as the optimum number of developers, although not always compulsory, in an agile team. The skill level of the team is a continually stated requirement in research into agile methods. Martin (2003) further defines skill level by stating that a strong player does not necessarily have to be an expert programmer, though they must work well with others. Good communication skills and an ability to interact with others are of higher importance than expertise in a programming language. Furthermore, Reifer (2002) makes an interesting observation, which may have repercussions for agile research and implementation. Reifer (2002) surveyed thirty one agile projects across eight industry sectors and comments that the teams involved were made up of

motivated, experienced programmers, in cohesive teams. Furthermore, Levine *et al.* (2002) believe that agile projects must be staffed with skilled, experienced, software developers, while Lindvall *et al.* (2002) make a similar observation by discussing the percentage of an agile team that needs to be experienced or composed of 'good' people. As these descriptions do not apply to every developer and team in the software industry, there may be issues with agile if implemented in below average teams.

2.1.3 Customer: The customer grouping describes the customer, or customers, of the project, identifying location of the customer and customer involvement as critical adoption factors to consider in adopting an agile methodology. Beck and Fowler (2001) provide a flexible definition of customer, in that the customer is the person who makes the business decisions i.e. completion date, scope, etc. This person can be an internal product manager or an individual who will purchase the software. Young (2003) describes how a DSDM project in British Airways addressed the necessity for user involvement in DSDM projects. The project was in the area of e-commerce, so its user base was extensive and difficult to categorize. British Airways used sales and marketing staff to represent the customers. Furthermore, Jeffries (2001) specifically uses the term customer as opposed to customers, as there is a requirement for a single customer voice, which implies difficulties for projects with multiple customers in multiple regions. However, Reifer (2002) shows that the theory of customer participation is not reflected in real agile projects, in that customer surveys are used as a replacement, and attempts are made to get customer representatives on site as often as possible. In addition, Beck and Fowler (2001), while accepting the various definitions of customer, or customer stand-in, declare that a customer must be an integral part of the team. Without this integration into the team, the project will fail.

2.1.4 Organization: The organization grouping highlights the criticality of the organizational environment, which comprises the following: organizational and reporting structure; process; documentation requirements; and layout of workspace. Boehm (2002) states that the agile approach requires both responsive people and organizations. The relationship between managers and developers is one of collaboration rather than the traditional command and control structure (Cockburn and Highsmith, 2001). "*The inefficiency of rigid bureaucratic structures, with many hierarchical levels and watertight compartments, can no longer be tolerated in companies that must compete in turbulent environments*" (Dolan *et al.*, 2003, p 28). Also, Reichheld (2001), although discussing loyalty in the workforce, makes the pertinent point that in the complex work environment that exists, organizations need to be adaptable. To ensure this adaptability, managers must simplify the rules of decision making, adopting an organic system of management, as defined by Burns and Stalker (1961). Furthermore, Greening (2001) emphasizes the importance of a senior engineer in XP teams, therefore, ensuring that knowledge is spread through the team. Also, Rainwater (2002) describes management in extreme programming as being primarily carried out by the coach, the person with the ultimate responsibility for the project. These views are similar to the concept of a chief programmer addressed by Licker (1983) and Brooks (1995). It is clear that process, or the lack of process, plays a significant role in agile methods (Highsmith 2000). The agile concept and the importance of individuals over process highlights that "*a good process will not save the project from failure if the team doesn't have strong players*" (Martin, 2003, p 4). Therefore, the absence of rigid processes is embraced. "*Fast companies have a defining sense of purpose, supported by a few simple rules. They determine what is essential and ignore the rest*" (Gandossy, 2003, p 32). For example, Greening (2001) describes the difficulties involved in the implementation of XP, in a company that has a formal software development process.

A variation, or continuation, of the discussion on process is the documentation requirements of an agile project. Greening (2001) stresses that XP does not prohibit documentation, it merely stresses that documentation has a cost. Any documentation to be used must be evaluated to ensure that it has a benefit and that it is definitely necessary. The usefulness of documentation, as a method of controlling a project, is disputed by Johansen and Perkins (2002). Olson and Stimmel, (2002, p11) argue against XP by stating that documentation is required for the future engineers who will maintain the software product. "*Without good documentation, you know you will suffer undue loss if you lose staff*".

The layout of the workplace receives considerable attention in literature on agile methods. Poole and Huisman (2001) describe the efforts made in Iona Technologies to enhance team communications. The previous environment they describe resembles that of the Dilbert cartoon. The multiple bays of shoulder height cubicles were replaced by a common area, or group workspace. This was an attempt to increase awareness of the team, as opposed to individuals in their own tiny spaces. Kalita (2003) demonstrates that co-located teams, although not a formal principal of DSDM, should be used in DSDM projects to facilitate the necessity of communications.

3. RESEARCH APPROACH

In undertaking this research study we adopted a two phase approach to organising the Critical Adoption Factors workshop. The workshop format between 'Phase One' and 'Phase Two' was structured differently, due to that fact that the outputs of each phase were concerned with achieving different goals. In the first instance, phase one involved an examination, by students, of the critical adoption factors (Table 1), and the prioritising of these factors through assigning a generic weighting to each factor comprising the 'Adoption Assessment Matrix', presented in Table 4. While the primary goal of phase one was a pedagogical one, concerned with increasing students understanding of agile methodologies, and in what context these methodologies would provide benefit; the secondary goal was the creation of an adoption assessment matrix (decision support tool) which would be used in industry. Thereafter, phase two was

concerned with the practical application of the adoption assessment matrix. Two development groups from two companies used the adoption assessment matrix, as part of their investigation into the viability of an agile approach to software development. The primary goal of phase two was to factor these relevant industrial experiences and learning into the classroom material for future use, in affect the industrial reality would be used to reinforce or question the theories.

3.1 Phase One: Academic Context

Our pedagogical approach followed the cognitive school of thought, as described by Ul-Haq *et al.* (2003), which is learner-centered, involving problem solving and discussion. The workshop described in this paper uses multiple facets of active learning techniques[2], where "*students actively participate in the learning experience rather than sit as passive listeners*" (Lammers and Murray, 2002, p 55). Although these forms of learning need careful design, they are very effective in increasing knowledge retention, encouraging complex thinking about complex issues, and encouraging acceptance of different ways of learning on the part of the student. Active learning ultimately involves the student in the educational process and the benefits to the student include an increased ability to use the cognitive skills of objectivity, creative thinking, judgment, interpretation and problem solving (Seeler *et al.*, 1994), while more importantly students understand and display competent knowledge of the classroom material.

'Phase One' of this research study was conducting in a two hour 'workshop' style environment, with a class of twenty four students, divided into four groups. The sizing of groups aligns with Daniles *et al.* (2002) who argue that a group size of six is the optimum size for projects designed to deliver real world experiences. The project was group based, as opposed to individual, to demonstrate the complexity involved in decisions in industry, which predominantly involve group decisions and negotiations. Working in groups allowed argumentative reasoning, described by Love (2000, p 431) as "*various participants attempted to justify their own position or to persuade others to that opinion.*" As there was to be only one decision support process (adoption assessment matrix) designed per group in this workshop, consensus was required. This aligns somewhat with Richardson's (2003, p 1625) description of psychological constructivism where "*if the individuals within a group come to an agreement about the nature and warrant of a description of a phenomenon or its relationship to others, these meanings become formal knowledge.*" Each group was informed that the object of the exercise was to assess the viability of each of the two cases (Telecoms and Banking) in adopting an agile method for their software projects. Specifically, the lecturer highlighted to the groups that they should approach this exercise as if they were about to develop a generic decision support tool that would assist companies in the decision whether to adopt an agile methodology, namely Extreme Programming. At the beginning of this workshop, relevant lecture material was discussed (Section 2), around the area of agile methodologies and their application.

The use, and benefits, of workshops are described in Garcia and Moreno (2004); Krone *et al.* (20002); Hayes (2002); Daniels et al (2002); and Hilburn (1997). The benefits of workshops lie in their ability to represent the "real world" to the student. Several authors compare this to "toy" projects, which have little correlation with the realities in industry. Case studies can be used to bring theories to life (Krone *et al.*, 2002) and "*they make it easy to synchronise theory and practice*" (Garcia and Moreno, 2004). Our approach differs somewhat, while we agree with the benefits described, our approach ultimately causes the students to critically review the theory to ensure that it actually describes the industrial reality. Therefore, rather than simply using workshops to support theories, our approach ensures that the students critically question the theories with real world data.

3.2 Workshop Format

The format of the workshop centered around a number of steps being carried out by the student groups. Firstly, the groups prioritized each of the critical adoption factors (Table 1) based on their overall understanding of agile methodologies and assigned a generic weighting to each factor. This weighting was later plugged into the adoption assessment matrix, presented in Table 4. The student groups then focused on the specific cases under study, and in effect, the lecturer was representing the two teams, acting as the client. The lecturer's knowledge of the cases was based on insights gathered from research conducted on both the Telecoms and Banking development teams. The discourse was not regulated by the lecturer, in that once the requirements are known, the pacing of the workshop is the only control exercised. From the point of view of this research study, this is the opposite of traditional teaching roles, described by Morrison and Johnston (2003, p 151) as "*controlled dissemination of knowledge.*"

The two cases studied were deliberately chosen as they provide diverse views to the student on the possibility of utilizing an agile methodology, providing a more accurate depiction of the variations of practice in industry. One case was based on a software development team in the telecommunications sector that is considering the implementation of an agile methodology, and currently investigating the usefulness of agile techniques. The second case is from a software development team in the banking sector, which currently utilizes an agile methodology in its existing projects. However, neither company uses a formal decision process/system when determining the viability of agile methods for their projects. Also, both companies would be considered to be among the top ten companies in their sector.

At this point in the exercise, rankings were applied to each of the critical adoption factors in each of the cases. The rankings applied were based on the examination and interpretation, by each group, of each of the adoption factors, bringing into play their acquired knowledge and understanding of the issues of importance in adopting an

[2] Our understanding of active learning techniques is based on the work of Seller *et al.* (1994) and Bonwell and Eison (1991).

agile method, and applying it to the specific case. Each of the student groups rankings were further added into the adoption assessment matrix and multiplied by the weighting for each critical adoption factor, providing a result for each critical adoption factor for both the Telecoms and Banking cases. Finally, the totalled output was a confidence rating for each of the cases studied. These 'confidence ratings' represented the likelihood of success for Extreme Programming in the organization. A sample of the most representative adoption assessment matrix is illustrated in Table 4. (Note: each group determined the importance and criticality of the project within each case. Based on Section 2.1, the students determined that real-time projects were to be rejected automatically as unsuitable for agile methods.)

On completion of the workshop exercise, each student completed a survey to determine the usefulness of the teaching approach (active learning techniques). The survey was used to identify the benefits of the pedagogical approach for both the students and the lecturer. (The results are shown in Table 3.) It provided the students with an opportunity to reflect on what they had learned, as described in Thorpe (2000). It is accepted that this reflection is limited, from the lecturer's perspective, to the technical orientation of reflection, as described in Day (1999).

3.3 Initial Observations and Analysis of Feedback from Phase One

The usefulness of the exercise to the students is clear, from the 92% of students who responded positively, commenting that they now had a better understanding of Extreme Programming. The usefulness to the lecturer is less obvious from the survey. The ultimate goal for the lecturer was to use the adoption assessment matrix for each case with the object of increasing the student's knowledge and understanding of Extreme Programming.

Survey Question					
	Responses				# students
	Yes	Yes (%)	No	No (%)	24
Q1 Did you find the exercise useful?	22	92%	2	8%	
	Responses				
	Yes	Yes (%)	No	No (%)	
Q2 Do you have a better understanding of Extreme Programming after this exercise?	18	75%	6	25%	
	Responses				
Q3 If you had more time during the exercise, what would you have done?					
Determine Weightings for the CAFs	16	66.67%			
Examine additional Case Studies	5	20.83%			
Examine alternative Methodologies	3	12.50%			
	Responses				
Q4 What was the most difficult aspect of the exercise?					
Weightings for the CAFs	18	75%			
Determining the CAFs	6	25%			
	Ranking				
Q5 What aspect of the exercise was most beneficial?					
Determining the CAFs	1				
Working in Groups	2				
Weighting the CAFs	3				
Using Case Studies	4				
	Responses				
Q6 Did working in groups help you?	24	100%			

Table 3: Student Survey Responses

However, 25% of students felt that their understanding of Extreme Programming did not increase as much as their knowledge of all agile methodologies, in general. This is not a negative impact, but it does highlight that future changes are required to enable a concentration on Extreme Programming. The choice of case studies may have directed the students away from Extreme Programming, as 20.83% of students stated that other case studies would be useful, but they did not specify if it would assist in concentrating on Extreme Programming.

Interestingly, 66.67% of the students responded that, if they had more time, they would like to spend it on weighting the critical adoption factors. This is interesting as 75% found this to be the most difficult aspect, and it was ranked only third as providing the most benefit. It is possible that the short time-frame (two hours) for the workshop exercise had an impact on this result.

Overall, the most beneficial aspects of the workshop exercise were, in order of benefit:

- Determining the CAFs
- Working in groups
- Weighting the CAFs
- Using Case Studies

The choice of group work over individual work appears to have had a positive outcome. All students agreed that, despite the difficulties associated with group consensus, working in groups assisted in the completion of the adoption assessment matrix, and in interpreting the case studies.

The output of the adoption assessment matrix, the 'confidence ratings', provided the lecturer with a further indication of how the students were applying their knowledge and understanding from the workshop. Lecturers are often accused of providing little or no feedback to their students (Mutch, 2003). Although in this instance the feedback provided to the student was limited to a comparison between their final confidence ratings and the lecturer's subjective confidence ratings, it was felt that this was still a useful indicator for the students as to the standard of their work. In each group, for both cases, the confidence rating was approximately 10% higher than the lecturer's 'subjective rating', based on knowledge gained from past research conducted in these sites. This implies that areas of the classroom material need to be examined to determine if aspects of the decision are omitting factors that cause an over-optimistic interpretation and ranking from the student groups. Therefore, it is possible that negative aspects of Extreme Programming need to be further emphasised in the classroom material. The groupings of the critical adoption factors (project, team, customer, organization), presented in Table 2, also provide further feedback on students interpretations and further highlights areas that require further emphasis in the classroom material. Based on the analysis of the workshop exercise, and the outputs produced from the adoption assessment matrix, the project grouping (duration of the project; acceptance of change; criticality of project) along with customer (location of customer; customer involvement) received the highest rankings in terms of

Critical Adoption Factor	Weighting	CASE A Rank Telecoms	CASE A Result Telecoms	CASE B Rank Banking	CASE B Result Banking
Duration of project					
1 = more than 5 years 5 = less than 6 months	4	4	16	4	16
Location of customer					
1 = many customers in many countries 5 = In-house	4	1	4	5	20
Customer involvement					
1 =will have no interaction 5 = willing to fully interact	4	2	8	4	16
Acceptance of changes (to requirements)					
1 = Rigid 5 =Flexible	4	5	20	3	12
Team Size					
1 = more than 20 5 = 3	3	5	15	3	9
Skill of team					
1 = inexperienced 5 = very experienced	3	4	12	4	12
Organization and reporting structure					
1 = many reporting layers 5 = flat structure	2	4	8	3	6
Process					
1 = 5 or more standards to follow 5 = No standards to follow	2	1	2	2	4
Documentation requirements					
1 = A lot of documentation required 5 = Very little documentation required	1	1	1	2	2
Layout of workspace					
1 = Individual cubes, people Isolated 5 = Open plan, no walls	1	3	3	3	3
	Confidence Ratings		**64**		**71**

Confidence Rating Calculation = (SUM(case results)/(SUM(weighting)*5))*100 Telecoms Example: 64 = (89 / (28 * 5)) * 100

Table 4: Adoption Assessment Matrix

overall impact on the viability to adopt an agile method. However, the organization grouping (organizational and reporting structure; process; documentation requirements; layout of the workspace) received the lowest rankings, implying that students perceive these critical adoption factors to be of little importance to the viability of the adoption of an agile method. However, if we refer back to the literature presented in Section 2.1, we can observe the criticality attributed to the organizational environment and its constituent parts. Therefore, it needs to be determined if this is due to a lack of emphasis in the lecturers classroom material of the perceived importance of the organization grouping.

Up to this point, we have presented the use of a workshop to reinforce theory, as per Krone *et al.* (2002) and Garcia and Moreno (2004). Phase two of this workshop examines how the industrial reality differs from the students under-standing of the agile theories, as presented in phase 1.

3.4 Phase Two: Industry Context

Within 'Phase Two', two software development companies took part in separate workshops. These workshops were somewhat similar in design to phase one, in that an overview of agile methods was presented to company participants, and both companies were extremely different in order to provide a diverse view of the possibilities in adopting an agile methodology. However, in phase two, the workshop

participants were using the adoption assessment matrix (developed from phase one) as an actual decision support tool, in that they were using their practical development experiences to assess the viability of adopting an agile method for their specific software development projects. Developers and managers completed the adoption assessment matrix independently, so as to provide a broader perspective from each company, but also to allow comparisons to further refine the weightings of the individual critical adoption factors, and of the groupings of factors. The adoption assessment matrices completed by the companies were analysed to determine if there were any critical adoption factors that need to be considered for inclusion or warrant greater emphasis in the presentation of future agile methods classroom material.

3.5 Initial Observations and Analysis of Feedback from Phase Two

The thirteen adoption assessment matrices completed by the two companies were analysed in order to assess their perception of the viability of the adoption of an agile method, with the view to improving the lecturers' depth of knowledge of the application of the adoption assessment matrix. This is turn would be used to provide a more relevant and accurate description of the agile critical adoption factors to the students. A summary of the completed matrices for Company A and Company B is presented in Table 5 and Table 6, respectively.

The most striking observation, from the analysis was that managers were more positive about the benefit of adopting an agile methodology than the developers themselves. The average confidence ratings for managers against developers were 60% compared to 55% (for Company A) and 81% compared to 73% (for Company B). This seems unusual as we expected developers to appreciate the agile methodologies more than managers, due to the fact that agile methodologies propose a movement away from rigid management to one of increasing trust in, and empowerment of, developers. It is worth noting that classroom material used, and the literature it was based on, proposes that developers would be more amenable to the agile concepts than managers. However, our observation, as illustrated above, highlights the existence of a contradictory conclusion between the theory literature used in phase one and the companies participating in phase two. Therefore, this provides an opportunity for management as described in Koontz *et al.* (1980); Lucas (1982); and Cole (1988) would imply an approach that, while not anti-agile, could be considered as presenting problems for the use of agile methods. Therefore, it is interesting to find managers being more optimistic and positive about an agile approach than developers (who some consider agile to be aimed at). One possible reason is that agile is regarded by developers as just another set of processes aimed at forcing a team to comply with policy, as suggested by Birmingham (2000). The impact of this, from the viewpoint of the adoption assessment matrix

Role / CAF	Dev	Dev	Dev	Dev	Dev	Manager		Average Developer	Manager
Rating	**59**	**58**	**57**	**51**	**52**	**59**		**55**	**60**
Project	28	32	28	24	28	32		28	32
Team	27	24	24	24	24	21		25	21
Customer	12	12	12	12	12	12		12	12
Organization	15	13	16	12	9	18		13	18

CAF = Critical Adoption Factors; Dev = Developer

Table 5: Summary of Adoption Assessment Matrix Outputs in Company A

Role / CAF	Dev	Dev	Dev	Dev	Dev	Manager	Manager		Average Developer	Average Manager
Rating	**78**	**69**	**76**	**71**	**74**	**79**	**83**		**73**	**81**
Project	36	28	28	28	28	28	36		30	32
Team	21	18	27	24	24	24	27		23	26
Customer	32	24	28	28	28	32	28		28	30
Organization	20	26	24	19	23	27	25		22	26

CAF = Critical Adoption Factors; Dev = Developer

Table 6: Summary of Adoption Assessment Matrix Outputs in Company B

is that the role of the individual completing the matrix needs to be taken into consideration (e.g. accepting that managers may be overoptimistic and/or developers may be overly pessimistic). Therefore, this observation needs to be introduced into classroom material. It is noted though, that the difference in confidence ratings between managers and developers is not that large.

Analyzing the groupings of critical adoption factors provides additional feedback which will be of future benefit in the classroom. The greatest difference between developers and managers arises from their views on the rankings applied to the organizational group of adoption factors. The organizational factors concern the organizational environment in which projects take place, and in both companies managers are more positive than developers. As a result, this may be another interesting area for classroom discussion. This finding may be explained by the fact that organizational factors encompass areas of control: process, documentation, etc. These areas are predominately management requirements, and are quite often resented by developers. It is in the area of control, and the level of control, that management and development often differ, and "*the inescapable interface between managers and their employees is the control process*" (Storey, 1983, p 83). Again, this is an area that will have an impact on the adoption assessment matrix. Therefore, the level of control in a company will need to be further examined in the classroom, to better determine why developers specifically give a lower weighting.

The area where developers were more positive than managers was that of the skill of the team. It is probably obvious that developers would have a higher opinion of their own ability than managers, but, from the point of view of the student studying the adoption assessment matrix, the role of the individual completing the survey has an impact. The overall average confidence rating for both companies shows a large difference. The confidence rating is an indication of the suitability of an agile methodology for this particular development team, for a particular project. Company A had an average confidence rating of 56% while Company B had a confidence rating of 76%. Knowledge of the two companies explains an aspect of this difference, but not completely. The main factor in this difference is with the organizational group of critical adoption factors. The matrices from Company A show that there is a higher degree of process required, and this was verified through discussions with the team. Company B is a considerably smaller organization than Company A, so there is a lower level of process. The adoption assessment matrix highlights this fact, so the importance of the organizational factors in the matrix is evident in the industrial context. Therefore, this highlights the difference between the students' perception and the reality of an industrial setting, with regard to the importance of the organizational factors. Furthermore, from a pedagogical perspective, this further highlights the need to introduce the importance of the organizational group of factors to the students.

4. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Overall, the unusual alignment and unique approach of introducing a decision support process (adoption assessment matrix) to determine the viability of adopting agile methods, coupled with the use of cases, brought positive benefits and insights to both students and the lecturer. However, based on our initial observations and an analysis of feedback from the students involved in the workshop exercise, a number of improvements can be introduced for future work in this research study.

From a pedagogical perspective, this workshop has some deficiencies which could be addressed in future planned workshops. The two major deficiencies, which need to be addressed in future work, are listed below.

- The factors, which determine individual and group's abilities to acquire knowledge, were not considered in detail. These factors include power, social, and economical issues - described as social constructivism by Richardson (2003). Swiercz and Ross (2003) describe how case studies tend to have a concentration on the rational domain – structure, strategies, technology, etc. This is to the detriment of discussions of the human and political aspects of an organization or system under review.
- Time factors prevented the actual creation of a decision support tool by the groups. Potentially, the use of computer based learning, where the core of the decision support tool was already created and the students merely "fed in" their data, could have provided benefit. Even the ability to see a working version of their solution could have had a positive effect on learning. The positive motivational effects of computer use in learning, as described in Becker (2000), may have allowed this activity to continue outside of classroom hours.

At this stage in the research, it is comforting to note that the goals of our initial study have been fulfilled, in that the students understanding of the area of agile methodology seems to have improved considerably and the lecturer has been provided with extremely useful feedback on potential areas of revision and improvement in classroom material to further enhance students understanding and to enable them to critically review the theory. Therefore, as a learning tool, the workshop exercise could be altered (allocate a longer time-frame for completion, introduce additional cases) to provide even more benefit to the students. Furthermore, as a decision support tool (adoption assessment matrix), there is potential for this model to be applied in industry, as it is unique in attempting to support the decision, in assessing the viability, to adopt an agile methodology, while this knowledge gained from industry can be re-introduced into the classroom-based workshop. Therefore, the next phase of this research study involves the researchers taking the adoption assessment matrix into other companies to further refine its comprising critical adoption factors, their weightings, and the overall design of the decision support

tool (the inclusion or exclusion of certain critical adoption factors). The researchers believe that with further refinement, the adoption assessment matrix has the potential to be utilised in organizations considering the adoption of an agile methodology.

Overall, the analysis of the matrices completed by the two development companies shows the benefit of the adoption assessment matrix in the learning process. The matrix developed in phase one helped students to gain a better understanding of agile software development. The feedback received from industry, in phase two, will be used to further refine the teaching material used. Probably the greatest benefit from the industrial feedback is the new areas for discussion in the classroom setting. Students will be asked to explain why some industrial findings are different than those proposed in classroom material and the literature. This will ensure that the students will be creative in their thought process, as the literature they would normally consult for answers is being called into question.

5. REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. (2002) "Agile software development methods. Review and analysis." VTT Publications. Finland.

Barros, M., Werner, C. and Travassos, G. (2004) "Supporting risk in software project management." Journal of Systems and Software. 70(1-2). pp 21-35.

Beck, K. and Fowler, M. (2001) "Planning extreme programming." Addison Wesley. NJ

Becker, H. (2000) "Pedagogical motivations for student computer use that lead to student engagement." Educational Technology. 40. pp 5-17.

Birmingham, D. (2002) "Software development on a leash." Springer-Verlag. NY

Boehm, B. (2002) "Get ready for agile methods, with care" IEEE Computer. 35(1). pp 64-69.

Bonwell, C.C. and Eison, J.A. (1991) "Active Learning: Creating Excitement in the Classroom". George Washington University, Washington DC

Brooks, F. (1987) "No silver bullets: Essence and accidents software engineering." IEEE Computer. 20(4). pp 10-19.

Brooks, F. (1995) "The mythical man month: Essays on Software Engineering." Addison-Welsey Publishing.

Burns, T. and Stalker, G. (1961) The Management of Innovation, Tavistock Publications, London, England.

Carr, D. (2003) "Philosophy and the meaning of education." Theory and Research in Education. 1(2). pp 195-212.

Cockburn, A. and Highsmith, J. (2001) "Agile software development: The people factor." IEEE Computer. 34(11). pp 131-133.

Cole, G. (1988) "Management Theory and Practice Second Edition". DP Publications. Hants, England.

Daniels, M., Faulkner, X. and Newman, I. (2002) "Open ended group projects, motivating students, and preparing them for the real world." Proceedings of the 15th Conference on Software Education. pp 115-126.

Day, C. (1999) "Professional Development and Reflective Practice: purposes, processes and partnerships." Pedagogy, Culture and Society. 7(2). pp 221-233.

DeMarco, T. and Boehm, B. (2002) "The agile methods fray." IEEE Computing. 35(6). pp 90-92.

Dolan, S., Garcia, S. and Auerbach, A. (2003) "Understanding and managing chaos in organisations." International Journal of Management. 20(1). pp 23-35.

DSDM (1998) "Guidelines for the introducing DSDM into an organisation." DSDM Consortium. Kent. England.

Emery, J. (2001) "Adaptive software development: An experience report." in Constantine, L. (ed) "Beyond chaos: The expert edge in managing software development". pp 273-280. Addison-Wesley. NJ, USA.

Gandossy, R. (2003) "The need for speed." Journal of Business Strategy. 24(1). pp 29-33.

Garcia, F. and Moreno, M. (2004) "Software modelling techniques for a first course in Software engineering: A workshop based approach." IEEE Transactions on Education, 47(2). pp 180-187.

Glass, R. (2001) "Agile versus traditional: Make love not war." Cutter IT Journal. 14(12). pp 12-18.

Greening, J. (2001) "Launching extreme programming at a process-intensive company." IEEE Software. 18(6). pp 27-33.

Hayes, J. (2002) "Energizing software engineering education through real-world projects for experimental studies." Proceedings of the 15th Conference on Software Engineering Education. pp 192-206.

Highsmith, J. (2001) "Opening Statement". Cutter IT Journal. 14(12). pp 2-4.

Highsmith, J., Cockburn, A. (2001) "Agile software development: The business of innovation." IEEE Computer. 34(9). pp 120-122.

Hilburn, T. (1997) "Software engineering education: A modest proposal." IEEE Sofware. 14(6). pp 44-48.

Jeffries, R. (2001) "Card magic for managers: low-tech techniques for design and decisions." in Constantine, L. (ed) "Beyond chaos: The expert edge in managing software development". pp 27-32. Addison-Wesley. NJ

Johansen, K. and Perkins, A. (2002) "Establishing an agile testing team." Extreme Programming and Agile methods – XP/Agile Universe Chicago 2002". p 52-59.

Kalita, T. (2003) "DSDM in process improvement" in Stapleton, J. (ed) "DSDM Business focused development. Second edition." pp 175-191. Pearson Education. NY, USA

Kirkpatrick, R., Walker, J. and Firth, R. (1992) "Software development risk management: an SEI appraisal". SEI Technical Review. Software Engineering Institute. PA

Koontz, H., O'Donnell, C. and Weihrich, H. (1980) "Management Seventh Edition". McGraw-Hill. Tokyo, Japan.

Krone, J., Juedes, D. and Sitharam, M. (2002) "When theory meets practice: Enriching the CS curriculum through industrial case studies." Proceedings of the 15th Con-ference on Software Education and Training. pp 207-214.

Krutchen, P. (2001) "Agility with the RUP." Cutter IT Journal. 14(12). pp 27-33.

Lammers, W. and Murray, J. (2002) "A profile of teaching techniques used in the university classroom." Active Learning in Higher Education. 3(1). pp 54-67.

Lamnias, C. (2002) "The Contemporary Pedagogic Device: functional impositions and limitations." Pedagogy, Culture and Society. 10(1). pp 21-38.

Levine, L., Baskerville, R., Link, J., Pries-Heje, J., Ramesh,B. and Slaughter, S. (2002) "Discovery Colloquim: Quality software development at internet speed". Technical Report CMU/SEI-2002-TR-020. Software Engineering Institute. PA, USA.

Licker, P. (1983) "The Japanese Approach: A Better Way To Manage Programmers". Communications of the ACM. 26(9). pp 631-636.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L. and Zelkowitz, M. (2002) "Experical findings in agile methods." Extreme Programming and Agile methods – XP/Agile Universe Chicago 2002". pp 197-207.

Love, K. (2000) "The regulation of argumentative reasoning in pedagogic discourse". Discourse Studies. 2(4). pp 420-451.

Lucas, H. (1982) "Companion to Management Studies". Heinemann Ltd. London, UK.

Martin, R. (2003) "Agile software development. Principles, patterns, and practices." Prentice Hall. NJ

Morrison, A. and Johnston, B. (2003) "Personal creativity fr entrepeneurship." Active Learning in Higher Education. 4(2). pp 145-158.

Mutch, A. (2003) "Exploring the practice of feedback to students." Active Learning in Higher Education. 4(1). pp 24-38.

Olson, D. and Stimmel, C. (2002) "The manager pool. Patterns for radical leadership." Addison-Wesley. NJ

Poole, C. and Huisman, J. (2001) "Using extreme programming in a maintenance environment" IEEE Software. 18(6). pp 42-49.

Rainwater, J. (2002) "Herding cats: A primer for programmers who lead programmers". Springer-Werlag. NY

Ramachandran, V. and Shukla, A. (2002) "Circle of life, spiral of death: are XP teams following the essential practices." Extreme Programming and Agile methods – XP/Agile Universe Chicago 2002". pp 166-173.

Reichheld, F. (2001) "Lead for Loyalty". Harvard Business Review. 79(7). pp 76-84.

Reifer, D. (2002) "How to get the most out of Extreme Programming/Agile methods." Extreme Programming and Agile methods – XP/Agile Universe Chicago 2002". pp 185-196.

Richardson, V. (2003) "Constructivist pedagogy". Teachers College Record. 105(9). pp 1623-1640.

Rising, L. and Janoff, S. (2000) "The scrum software development process for small teams." IEEE Software. 17(4). pp 26-32.

Seller, D.C., Turnwald, G.H. and Bull, K.S. (1994) "From Teaching to Learning: Part III. Lectures and Approaches to Active Learning", Journal of Veterinary Medical Education, 21(1). pp 1-9.

Stapleton, J. (2003) "DSDM Business focused development. Second Edition". Pearson Education. London, UK.

Storey, J. (1983) "Managerial Prerogative and the Question of Control". Routledge and Kegan Paul. London, UK.

Sutherland, J. (2001) "Agile can scale: Inventing and Reinventing SCRUM in five companies." Cutter IT Journal. 14(12). pp 5-11.

Swiercz, P. and Ross, K. (2003) "Rational, human, political, and symbolic test in Harvard business school cases: A study of structure and content". Journal of Management Education. 27(4). pp 407-430.

Thorpe, M. "Encouraging students to reflect as part of the assignment process." Active Learning in Higher Education. 1(1). pp 79-92.

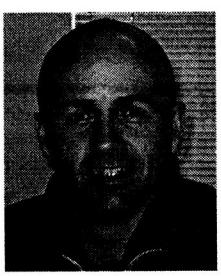
Ul-Haq, R., Stiles, J. and Pond, K. (2003) "Learning expectations and learning process design." Active Learning in Higher Education. 4(2). pp 168-180.

Williams, L., Kessler, R., Cunningham, W., Jeffries, R. (2000) "Strengthening the case for pair-programming." IEEE Software. 17(4). pp 19-25.

Young, G. (2003) "Implementing DSDM in eBA" in Stapleton, J. (ed) "DSDM Business focused development. Second Edition." pp 97-103. Pearson Education. London, UK.

AUTHOR BIOGRAPHIES

John McAvoy is a researcher/lecturer in Business Information Systems at University College Cork, Ireland. John holds a BE in Industrial Engineering from University College Galway and an MSc in Computer Science from University College Cork. Prior to joining BIS, John held a variety of roles in Information Systems, ranging from systems administration to managing software development teams. His primary research interest is in the area of software development methodologies, primarily the role played by small teams of software developers. Other research interests include IT education and operating systems performance optimization.



David Sammon is a researcher/lecturer in Business Information Systems at University College Cork, Ireland. David holds a BA in Economics and an MSc in Management Information Systems from University College Cork. His current research interests focus on decision making and organizational experiences around Data Warehousing, Enterprise Resource Planning systems implementation, and Application Service Provision adoption. He has presented his research at major

international conferences in Europe and in the United States and published in top ranking international journals including: *Information Systems Journal* and *DATABASE*. David is a co-author and co-editor of a book from Idea Group Publishing titled: *The Enterprise Resource Planning Era: Lessons Learned and Issues for the Future.*

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096