Association for Information Systems
AIS Electronic Library (AISeL)

ECIS 2006 Proceedings

European Conference on Information Systems (ECIS)

2006

# Towards the development of a simple tool to assist in agile methodology adoption decisions: agile adoption matrix

John McAvoy
*University College Cork*, j.mcavoy@ucc.ie

Ian Owens
*Cranfield University*, i.owens@cranfield.ac.uk

David Sammon
*University College Cork*, dsammon@afis.ucc.ie

# TOWARDS THE DEVELOPMENT OF A SIMPLE TOOL TO ASSIST IN AGILE METHODOLOGY ADOPTION DECISIONS: AGILE ADOPTION MATRIX.

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

Ian Owens, Department of Information Systems, Cranfield University, Defence Academy of the UK, Shrivenham, Swindon SN6 8LA, UK, i.owens@cranfield.ac.uk

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

## Abstract

*This paper describes the development of a decision support tool, an adoption assessment matrix based on critical adoption factors, that addresses a need in industry; namely, to improve the overall understanding of the constituent parts of agile systems development methodologies. It highlights the importance of critical adoption factors to the adoption of an agile method and illustrates the usefulness of a decision support process to determine the viability of an agile method for a specific software project. The paper describes the results of a series of workshops (two carried out in commercial software developement companies, and one with personnel from the British Minsitry of Defence) where the adoption assessment matrix was used to assess the suitability of agile methods in software development projects. A major benefit of the tool is that it guides discussion, concentrating the debate on the critical factors, applied to the individual project. These discussions proved to be as valuable as the output of the tool itself. The results of these workshops show that an argument can be made for the use of and benefit of such a decision support process in industry, in supporting the decision to adopt an agile methodology.*

*Keywords: Agile Methodology, Adoption Decision, Critical Adoption Factors.*

# 1 INTRODUCTION

Discussions on agile software development methodologies 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. There are no procedures available to assist software developers choose the best development method for a given situation (Abrahamsson & Salo & Ronkainen & Warsta 2002). In the literature, both academic and practitioner, definitions and descriptions of various agile methods are presented, but the factors that should be considered in the decision to adopt, or not adopt, an agile method are not addressed. This is not an uncommon trend in other areas of IS, for example: ERP adoption decisions, or the decision to adopt an e-business model (Sammon & Adam 2004, Hayes & Finnegan 2003), and heightens the need for studying the importance of the intelligence phase of decision making. Our literature review has demonstrated that there are no procedures or tools to assist in this task. Indeed, agile methodologies try to avoid the excessive use of tools (Beck & Fowler 2001). However, one agile methodology, Dynamic Systems Development Method (DSDM) does recommend the use of appropriate tools during the development process (Stapleton 2003, Jones 1998). Coesmans (2003, p.119) agrees stating that "working from the mindset of DSDM (the principles) and using the right tools from the toolbox, gets many difficult projects successfully done." In each case, although the tools are used to effect during the project, none are used to assess the viability of the methodology, before project initiation. Although one study by Boehm and Turner (2004) provides a graphical method of determining the suitability of adopting an agile methodology, for our purposes, it falls short in increasing the understanding of agile methods, by those considering adoption.

The next sections describe what we mean by agile methods and we provide a number of critical adoption factors that can be considered when deciding whether to adopt an agile methodology. These critical adoption factors are then used to develop an adoption assessment matrix that can be used by software teams to determine the viability of adopting agile methods. We then describe two workshops where the adoption assessment matrix was used by software developers in a commercial setting, and one workshop held with development staff from the British Ministry of Defence. Our results are presented with conclusions drawn from the workshops. In the final section we present recommendations for future research.

# 2 AGILE SOFTWARE DEVELOPMENT METHODOLOGY

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 & Cockburn 2001, Kruchen 2001). The basic underlying principles of agile methodologies are:

- 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 (Abrahamsson *et al.* 2002).

These principles are referred to as the 'Agile Manifesto'. There is no single agile development methodology. Some examples of agile approaches and methodologies that share many of these core values include: Extreme Programming (XP); Crystal Methods; SCRUM; Dynamic Systems Development Method (DSDM); Feature Driven Development (FDD); and Adaptive Software Development (ASD) (Highsmith 2001, Sutherland 2001).

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. "There are no silver bullets in software development, and there probably never will be" (Jeffries 2001).

Of all the agile methods, DSDM (DSDM 1998) comes closest to recommending an approach for determining whether a project is suitable for using an agile development method. DSDM recommends that the organisation must have the right culture for using agile approaches, but it is not specific about how one should go about measuring or evaluating this culture. Furthermore, the steps in the feasibility study that follow this recommendation involve educating a key stakeholder, and producing a strategy and plan. Based on these recommendations, it appears that the choice has already been made to adopt DSDM, so it is not a decision whether to adopt an agile approach or not.

The debate concerning agile methodologies has predominantly been based around whether it was a better choice than tradition development methods (De Marco & 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 the Decision to Adopt an Agile Methodology

Throughout the available literature on agile methodology, factors important to the success of an agile project are discussed, yet these factors are not specifically used to determine the viability of adopting an agile approach. The agile manifesto, described in Abrahamsson *et al.* (2000), 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 ten 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.

| | Critical Adoption Factors | Group | Description | References |
|---|---|---|---|---|
| 1 | Duration of the project | Project | The timeframe for the project should be short for an agile project | Highsmith (2001)<br>Kruchen (2001) |
| 2 | Location of the customer | Customer | It is preferable that customers are available throughout the project, therefore an in-house customer would be ideal | Young (2003)<br>Jeffries (2001) |
| 3 | Customer Involvement | Customer | Customer involvement is vital for the success of a project | Beck and Fowler (2001)<br>Young (2003) |
| 4 | Acceptance of change (to requirements) | Project | Agile methods are specifically aimed at projects subject to continual change | Highsmith (2001) |
| 5 | Team Size | Team | Agile methodologies emphasise the importance of teams, recommending smaller team sizes | Boehm (2002)<br>Rising and Janoff (2000) |
| 6 | Skill level of team | Team | Highly skilled developers are required | Reifer (2002)<br>Levine *et al.* (2002)<br>Lindvall *et al.* (2002) |
| 7 | Organisational and reporting structure | Organisation | An organic structure is required for an agile approach | Boehm (2002)<br>Cockburn and Highsmith (2001)<br>Dolan *et al.* (2003) |
| 8 | Process | Organisation | 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)<br>Greening (2001) |
| 9 | Documentation requirements | Organisation | While agile methodologies do not prohibit documentation, it should be kept to a minimum | Greening (2001)<br>Olson and Stimmel (2002) |
| 10 | Layout of workspace | Organisation | Open planned offices, with shared areas, are required to promote communication and team work | Poole and Huisman (2001)<br>Kalita (2003) |

*Table 1: Critical Adoption Factors for an Agile Methodology*

One overriding assumption made throughout this research is that ‘life-critical projects’ are not suited to an agile approach. Furthermore, the researchers acknowledge that there maybe differences of opinion as to what should constitute the ten critical adoption factors, but, are also aware that these factors could change through further research. These critical adoption factors are used to provide a simple structure to assist companies in the agile method adoption decision making process, through the use of an ‘Adoption Assessment Matrix’ (Table 2). These factors are not ordered or grouped, as it is not intended to lead the individual completing the matrix. However, each factor does relate back to the values of the agile manifesto, for example, the process factor relates to the importance of individuals over processes.

| Critical Adoption Factor | Weighting | Rank Case | Result Case |
|---|---|---|---|
| **Duration of the project**<br>1 = more than 5 years<br>5 = less than 6 months | 4 | | |
| **Location of the customer**<br>1 = many customers in many countries<br>5 = in house | 4 | | |
| **Customer involvement**<br>1 = will have no interaction<br>5 = willing to interact | 4 | | |
| **Acceptance of change (to requirements)**<br>1 = rigid<br>5 = flexible | 4 | | |
| **Team size**<br>1 = more than 20<br>5 = up to 3 | 3 | | |
| **Skill of team**<br>1 = inexperienced<br>5 = very experienced | 3 | | |
| **Organisational and reporting structure**<br>1 = many reporting layers<br>5 = flat structure | 2 | | |
| **Process**<br>1 = 5 or more standards to follow<br>5 = no standards to follow | 2 | | |
| **Documentation requirements**<br>1 = a lot of documentation required<br>5 = very little documentation required | 1 | | |
| **Layout of Workspace**<br>1 = individual cubes, people isolated<br>5 = open plan, no walls | 1 | | |
| | | | |
| **Confidence Rating** | **28** | | |

**Note on Workings of Matrix:**

**[1]** Rankings (between 1 and 5) are applied to each of the critical adoption factors. These rankings are based on a workshop participants assessment of the factors applicability to a particular project.

**[2]** Each ranking is multiplied by the weighting for each critical adoption factor, providing a result for each critical adoption factor.

**[3]** Finally, the totalled output, referred to as the ’confidence rating’ is calculated as follows:

**Confidence Rating Calculation** = (SUM(Result Case)/(28*5))*100

*Table 2: Adoption Assessment Matrix*

On further examination of these critical adoption factors, and as highlighted in Table 1, we propose a further classification of four groups: Project; Customer; Team; Organisation; into which each of these factors can be suitably placed.

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.

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 describes how agile development is optimal when used in small teams (Boehm 2002). Martin (2003) 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.

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. In fact, Turk, France and Rumpe (2005) describe customer involvement as one of the core assumptions for XP. Beck and Fowler are willing to compromise on the definition of customer. The customer is the person who makes the business decisions i.e. completion date, scope, etc (Beck & Fowler 2001). This person can be an internal product manager or an individual who will purchase the software.

The organisation grouping highlights the criticality of the organisational environment, which comprises the following: organisational and reporting structure; process; documentation requirements; and layout of workspace. Boehm states that agile requires both responsive people and organisations (Boehm, 2002). The relationship between managers and developers is one of collaboration rather than the traditional command and control structure (Cockburn & Highsmith 2001).

Our research has shown that process, or the lack of process, plays a significant role in agile methods (Highsmith & Cockburn 2001). In discussing the agile concept and the importance of individuals over process, Martin (2003, p.4) highlighted that "a good process will not save the project from failure if the team doesn't have strong players". Therefore, the absence of rigid processes is embraced. A variation, or continuation, of the discussion on process is the documentation requirements of an agile project. It has been stressed 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 (Greening 2001). This is based on the assumption that tacit knowledge is more valuable than explicit knowledge (Turk *et al.* 2005)

In conclusion, we propose that these four groups (Project; Customer; Team; Organisation) represent the critical elements of any software project in terms of its suitability to the adoption of agile methods, namely, the project itself, the team assigned to the project, the client for whom the project is being delivered, and the organisational setting in which the project is undertaken.

# 3 DATA COLLECTION AND RESULTS

The Adoption Assessment Matrix was initially tested in a series of academic workshops, conducted over a one year period, with final year undergraduate and postgraduate Information Systems students. This phase allowed the researchers to test and operationalize our research instrument, the adoption assessment matrix, and validate the Critical Adoption Factors identified from the literature. Following this testing phase, the Adoption Assessment Matrix, was used in agile methods workshops organised by the researchers in two software development companies (Company A Telecommunications and Company B Engineering) and with personnel from the British Army, Navy and Air Force (tri-service). The two companies were selected because they operate in different industrial sectors: Company A Telecommunications and Company B Engineering. The tri-service workshop was held to see whether developers from very process oriented organisations would be confident to adopt agile methods. It was hoped that the differences between the groups selected to participate in the workshops would provide a diverse view of the possibilities in adopting an agile methodology in practice. In the workshops for the two commercial software development companies the participants were using the adoption assessment matrix 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. It is worth noting that a major benefit of the tool is that it guides discussion, concentrating the debate on the critical factors, once applied to the individual project. These discussions proved to be as valuable as the output of the tool itself. Developers and managers completed the adoption assessment matrix independently, so as to provide a broader perspective from each company of those comprising the project team, and to present possibilities to the researchers to further refine the individual critical adoption factors, and of the groupings of factors. It is important to note that the derived confidence rating is not compared to a threshold, used to recommend or reject the adoption of an agile approach. However, the confidence rating is used to facilitate debate, which in turn leads to a better analysis of suitability; for example, the matrix as completed by managers and developers of a project team highlights differences in the confidence ratings, which encourages debate.

In total thirteen adoption assessment matrices were completed by workshop participants from the two software development companies, and eight adoption assessment matrices were completed by the tri-service personnel. These were analysed in order to assess the participants' perception of the viability of the adoption of an agile method, with the view of also improving the adoption assessment matrix for future use. A summary of the completed matrices for Company A and Company B and the tri-service case is presented in Table 3.

| | Company A | | Company B | | Tri-Service |
|---|---|---|---|---|---|
| **Critical Adoption Factors** | **Average Developer** | **Average Manager** | **Average Developer** | **Average Manager** | **Average TS** |
| Duration of the project | 15 | 16 | 15 | 16 | 17 |
| Location of the customer | 4 | 4 | 13 | 14 | 14 |
| Customer involvement | 8 | 8 | 15 | 16 | 19 |
| Acceptance of change (to requirements) | 13 | 16 | 15 | 16 | 10 |
| Team size | 13 | 12 | 12 | 15 | 12 |
| Skill of team | 11 | 9 | 11 | 11 | 11 |
| Organisational and reporting structure | 6 | 6 | 8 | 10 | 8 |
| Process | 4 | 6 | 9 | 9 | 5 |
| Documentation requirements | 3 | 4 | 4 | 5 | 8 |
| Layout of Workspace | 2 | 2 | 2 | 3 | 11 |
| **Rating** | **55** | **60** | **73** | **81** | **66** |
| Project | 28 | 32 | 30 | 32 | 27 |
| Team | 25 | 21 | 23 | 26 | 20 |
| Customer | 12 | 12 | 28 | 30 | 31 |
| Organisation | 13 | 18 | 22 | 26 | 28 |

*Table 3: Summary of the three workshops*

The researchers used the tri-services case as an instrumental case (Stake 2000) to provide insight into certain issues around the decision to adopt an agile method. While the workshop format was identical to that conducted in the two commercial software development companies, the case was of secondary interest, playing a supportive role; the purpose of which was to facilitate our understanding of issues around: the criticality of a project, the documentation required for a project, the processes to follow in completing a project, and the timeframe for completion of a project; and the impact of these issues on the decision to adopt an agile method. The researchers' initial expectation of the tri-services case was one of unsuitability to adopt an agile method, due to the highly process-oriented nature of the British Ministry of Defence.

### 3.1 Discussion

The most striking observation, from the analysis of the two commercial software development companies 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. For example, the Fayol and Brech definitions of management as described in Koontz, O'Donnell and Weihrich (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. Fayol defined management as forecasting, planning, organizing, command, co-ordinate, and control, while Brech narrows the definition to a process of planning and regulating operations towards an agreed objective. 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 (2002). The impact of this, from the viewpoint of the adoption assessment matrix 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). It is noted though, that the difference in confidence ratings between managers and developers is not that large.

Analysing the groupings of adoption factors provides additional feedback. The greatest difference between developers and managers arises from their views on the rankings applied to the organisational group of adoption factors. This was the greatest difference in both software development companies. The organisational factors concern the organisational environment in which projects take place. Managers are more positive, in both companies regarding this area. This may be explained by the fact that organisational 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 of the adoption assessment matrix. The level of control in a company will need to be further examined within the structure of the matrix, 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 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 organisational 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 organisation than Company A, so there is a lower level of process. The adoption assessment matrix does highlight this fact, so the importance of the organisational factors in the matrix is shown. Furthermore, the difference in the customer grouping reflects the nature of the individual projects undertaken; where a project in Company A serves multiple customers, as opposed to a project serving a single customer in Company B. This explains the perception of customer involvement in a project, within Company A, which is identified in the lower rating attributed to the customer factors.

Overall, the analysis of the matrices completed by the two software development companies highlights the benefit of the use of the adoption assessment matrix. Although, the output of the tri-services workshop was expected to reveal rather different results from the commercial cases due to the process-intensive nature of the organisation, some interesting, if not divergent, issues were discussed. The respondents overall average confidence rating was 66% which was higher than the confidence ratings returned by Company A, and somewhat lower than the ratings returned by Company B. Again, the usefulness of the actual discussions around the ratings of the Critical Adoption Factors facilitated the participants to consider stages of a project, as opposed to the entire lifecycle of the project. The participants explained that they were reasonably confident that agile methods would work for the development stage of projects, but qualified this by stating that the initial feasibility, planning and analysis stages would favour process-oriented techniques. For example, projects within the Ministry of Defence go through a chain of command, however, it was also highlighted that some of these projects are often fast-tracked for completion in a shorter time-frame, and as a result, it was identified that agile methods would be suitable in such a situation. The participants also confirmed our overriding assumption that life-critical projects are not suited to an agile approach.

# 4 RECOMMENDATIONS FOR FUTURE RESEARCH

Although the agile methodology recommends limited use of tools, the purpose of this adoption assessment matrix is to support decision makers in assessing the suitability of adopting an agile approach to software development. Therefore, we consider the use of this decision support tool to be a precursor to the selection and use of an agile methodology. The use of the adoption assessment matrix moves beyond the debate on agile versus traditional methodologies and concentrates on specific cases of methodology adoption, rather than on agile being the universal solution. The two companies involved in this research study using the adoption assessment matrix believed that the matrix made them think differently about their decision, and provided a solid foundation for that decision. It is not proposed that the adoption assessment matrix provides the ultimate answer in making that decision, but its use does provide immediate benefits.

Therefore, the next phase of this research study involves the researchers taking the decision support tool (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 further successfully utilised in companies considering the adoption of an agile methodology.

# References

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

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

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

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

Boehm, B. and Turner, R. (2004) Balancing agility and discipline. Pearson Education, MA, USA.

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

Coesmans, P. (2003) “DSDM in a non-IT project” in Stapleton, J. (ed) “DSDM Business focused development. Second Edition”. pp 113-119. Pearson Education. London, UK.

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

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. Elsevier Science. Great Britain.

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

Glass, R. (2001) “Agile versus traditional: Make love not war.” Cutter IT Journal. 14(12). pp 12-18. Cutter Information Corp. MA, USA.

Greening, J. (2001) “Launching extreme programming at a process-intensive company.” IEEE Software. 18(6). pp 27-33. IEEE Computer Society. CA, USA.

Hayes, J. and Finnegan, P. (2003) “Assessing the potential of e-business models. Towards a framework for assisting decision-makers”, European Journal of Operational Research.

Highsmith, J. (2001) “Opening Statement”. Cutter IT Journal. 14(12). pp 2-4. Cutter Information Corp. MA, USA.

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

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, USA.

Jones, C. (1998) “Software Project Management in the 21st Century”. American Programmer. 11(2.) pp 24-29. Cutter Information Corp. MA, USA.

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

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

Kruchen, P. (2001) “Agility with the RUP.” Cutter IT Journal. 14(12). pp 27-33. Cutter Information Corp. MA, USA.

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.

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. Springer-Verlag. NY, USA.

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

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

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

Poole, C. and Huisman, J. (2001) “Using extreme programming in a maintenance environment” IEEE Software. 18(6). pp 42-49. IEEE Computer Society. CA, USA.

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. Springer-Verlag. NY, USA

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

Sammon, D. and Adam, F. (2004) Towards a Model of ERP Software Selection - Widening the Debate, in Adam and Sammon (2004) The Enterprise Resource Planning Decade: Lessons Learned and Issues for the Future, Idea Publishing Group, Hershey, PS, February.

Stake, R. (2000) “Case Studies” in Denzin, N. and Lincoln, Y.S. (eds.) “The Handbook of Qualitative Research. Second Sdition.” pp 453-454. Sage Publications. CA, USA

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

Storey, J. (1983) “Managerial Prerogative and the Question of Control”. Routledge & 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. Cutter Information Corp. MA, USA.

Turk, D., France, R. and Rumpe, B. (2005) “Assumptions Underlying Agile Software-Development Processes” Journal of Database Management, 16(4). pp 62-87

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