Is the world agile?
A review of the IT knowledge base and debate on agile methodologies in the field of information systems development

Klaudia Knosalla
Candidate for M.Sc. In Analysis Design and Management of Information Systems
Information Systems and Innovation Group
Department of Management
London School of Economics

This paper starts with presenting the context of the development of agile methodologies and a general justification for the existence of methodologies. The following section will give an overview of the different types of literature available in IS research as well as identify the gaps that can be mainly stated in the lack of empirical studies and the academic rigour they were conducted with. The next part will present the major arguments of proponents and opponents of agile methodologies. Then a third perspective will be introduced that takes a more balanced view on competing methodologies. It proclaims more openness to a flexible use of methodologies which can be more generally related to a paradigm shift and new implications for the work of developers.

Keywords: agile methodologies, XP, literature review, traditional methodologies

Introduction

The question of how information systems are constructed, has been one of the major research objects in the IS field during the last decades (Avgerou, 2000). Actually, we can state that there has always been a kind of quest for the "best methodology" in systems development, in a way relating to the desire for having a rational, systematic way of designing a system (Parnas & Clements, 1986). As to why methodologies are needed at all, the answer can be found in the different functions they serve; in short, they represent work practices, making it easier to introduce new people to the process, defining responsibilities, convincing sponsors or providing a curriculum for training (Cockburn, 2007).

According to Avison & Fitzgerald (2003), the methodologies with the systems development lifecycle prevailing that have evolved in the early methodology era (70s/80s), have showed serious limitations. Agile software development methodologies have evolved as an answer to the so-called "heavyweight" methodologies, such as the lifecycle model which emphasizes a rationalized, engineering-based approach (Dybå & Dingsøyr, 2008) with the assumptions that problems can be fully specified and that there is an optimal and predictable solution to every problem. With regard to changing requirements and smaller systems this approach has turned out to be too inflexible (Nerur et al., 2005).

As to the assumptions about organizations, agile methodologies see them as fluid and adaptive as well as the environment in which it operates, therefore much emphasis is put on skilled employees who are capable of accommodating change during the development process (Avison & Fitzgerald, 2006). Rajlich (in Dybå & Dingsøyr, 2008) encounters a paradigm shift in software engineering: In short it can be put as a shift from methodologies offering all things one could do to agile methods offering generative rules that have to be followed as a minimum (Highsmith & Cockburn, 2001). Agile methodologies have generated a lot of interest not only in the practitioner's world (Dybå & Dingsøyr, 2008) but also in academia, where it is taken on by academic journals as well as by conferences (Abrahamsson et al., 2002). In the following parts, the current knowledge base on agile methodologies with a focus on Extreme Programming will be assessed as well as the major arguments and current state of the debate in the IS community will be presented.

The IS knowledge base

As it comes to the literature available on agile methodologies, one stream is highly descriptive and prescriptive as to textbooks explaining what extreme programming is and what actions need to be taken to use it, for example (Cockburn, 2007; Gordon & Gordon, 2004). Recently, a vast amount of new agile methodologies has emerged. Many researchers and practitioners are not aware of approaches available and their suitability for particular development projects (Abrahamsson et al., 2003).

Although lots of literature and debates have evolved on agile methodologies, academic research is still scarce, as most of the publications are written by practitioners and consultants (Abrahamsson et al., 2002). There are some reviews available on agile methodologies. The one by Abrahamsson et al. (2002) provides a comparative analysis of methods giving "anecdotal evidence" that agile methods are "effective and suitable for many situations and environments" (Dybå & Dingsøyr, 2008). But nevertheless, there are only few empirical studies supporting these claims. Another piece of work by Cohen et al. (2004, in Dybå & Dingsøyr, 2008) focuses on the history of agile development but also on the important factors for the selection of a method as to the number of people working on a project, the application domain, its criticality and innovativeness. The interesting point of this review is that the authors are convinced that agile methodologies are going to be consolidated and will rather have a "symbiotic relationship" with traditional methods and not necessarily "rule them out". Erickson et al. (2005, in Dybå & Dingsøyr, 2008) give an overview of the state of research on XP, the more well-established stream of research on agile methodologies. Dybå & Dingsøyr (2008) in their review of empirical studies of agile software development argue that there has been no systematized review of research, so that practitioners and researchers are mostly relying on practitioner books to get an overview. As their review method is concerned, they have set up a transparent set of inclusion and exclusion criteria, i.e. using journal articles from electronic databases and conference proceedings as well as indicators for quality as to rigour, credibility and relevance leading them to identify 36 out of 1996 studies as empirical ones. The studies included in their review showed that most of them were done on XP (25; 76%). Furthermore, 24 studies (73%) dealt with developer teams who were beginners in using agile methods. One of their major critiques is that the methods used in the studies were not described adequately.

A large part of the literature on agile methodologies consists of case studies describing practitioners' experiences (cf. Drobka et al., 2004; Smith & Pichler, 2005; Andersen, 1998). In the article by Williams & Kessler (2000), they refer to their "Pair Programming Questionnaire" showing that programmers were more confident in their solutions than if they were programming alone and that 96% showed a higher job satisfaction. Still, it remains unknown on what data and questions the results are based. In the case of Motorola (Drobka et al., 2004), the authors who had no experience with XP, report an increase in morale and engineer productivity as well as very good test coverage. One of the interesting aspects of this practitioners' experience is that in order to meet organizational requirements, there has been an attempt to identify a productivity formula to demonstrate tangible benefits to the senior management – in other cases the economic perspective on the development process is missing. Measuring economic benefits might remain an unresolved issue as the adoption of agile methodologies is often a result of former failure of traditional methods (e.g. Andersen, 1998) and therefore, comparison of traditional versus agile results is not likely to be feasible. Abrahamsson et al. (2002) also claim that already little is known about the return on investment into process technologies and even less is known about how much an organization will benefit from using agile methods. Although the initial experience reports are predominantly positive, "hard numbers" are very difficult to obtain.

Taking a more general view at the methods used for studying systems development methodologies, the part of evaluation represents a huge challenge in the IS field as an exhaustive, systematic and scientific study is not feasible (Wynekoop & Russo, 1997). The authors identify a "positivist box" of research methods and although a research paradigm shift in IS has established the legitimacy of interpretative methods, there is a strong bias to discuss successful practices as practitioners seem to be unwilling to share failures (ibid). Returning to the case of Motorola (Drobka et al., 2004), the authors provide some reflection on areas of difficulties as to scheduling for pair-programming or the difficulty of interfacing with teams using traditional processes. As far as the knowledge base is concerned, this aspect of different methodologies used for development projects within one organization and the effects it has on the organization's social context is rarely addressed.

Turning towards the debate on agile methodologies, it will become very evident that also partly due to the historically strong link between systems development and methods leading to problems in distinguishing method from systems development (Truex et al., 2000); the technical-rational view of engineering on information systems with strong lifecycle thinking is still very prevalent. Truex et al. (2000) more generally identify the mainstream "privileged text" in systems development methodologies as undertaking a technical-rational view. So let us now have a look on the arguments in the IS community.

The debate

Considering the fact that there is no agreement on what the concept of "agile" actually means (Abrahamsson et al., 2002), it is very interesting to have a look at the arguments that have evolved around it. In the debate we can identify three positions. There are the opposing camps of "traditionalists" and "agilists" and there are those who take a more balanced view on the use of methodologies in general (Nerur et al., 2005). The major problems or points of critique can be traced back to the tendency of organizations trying to create optimized and repeatable processes creating the biggest barrier for adopting agile methods as these organizations are implicitly looking for stability (ibid). Nerur et al. (2005) have undertaken an organizational and managerial perspective that is not found explicitly in other pieces of work. One important factor is the organizational culture, as job roles are redefined with the adoption of agile methods. For example, the project manager role shifts from a planner and controller to a facilitator who directs and coordinates. One requirement for agile methods is to have "premium people" who are competent and above-average (Boehm, 2002). This raises questions of how to attract them and how the "below-average" developers in the organization react. As there is a high reliance on teamwork, questions about performance measurement are raised, too (Nerur et al., 2005). Another point of critique is the absence of documentation, but as Fowler & Highsmith (2001) stated in the Agile Manifesto, agile methodologies embrace documentation but not to "senseless" extents as it does not represent the primary goal of the development effort. Turk et al. (2002) are thinking about situations where agile methodologies may generally not be applicable as to developing safety-critical software or large and complex software. They also address the difficulties of use in distributed development environments or subcontracting in outsourcing projects, as they are characterized by precise conditions as to deliverables.

Agile methodologies have four core values that define where the priorities are lying: individuals and interactions, working software, customer collaboration and responding to change (Fowler & Highsmith, 2001).

These are not meant to completely ignore other important aspects as opponents are attaching blame to agilists. Rakitin (in Boehm, 2002) has a "hacker interpretation" for this and claims that the values serve as an excuse for hackers to throw code together without paying attention to engineering discipline; Wiegers (in Avison & Fitzgerald, 2006) calls it "a licence to hack". This is probably one of most severe critiques.

The practices coming with agile methodologies are not new as some of the opponents are pointing out explicitly but the recognition of people as primary drivers is (Abrahamsson et al., 2002). Going through the literature by the community of agilists, it becomes evident that they are very self-reflective and less dogmatic than the traditionalists. They are very much aware of the limitations of agile methodologies, as Scott Ambler, one of the originators of agile modeling, expresses (in Boehm, 2002):

> "I would be leery of applying agile modeling to life-critical systems."

Beck (in Abrahamsson et al., 2002) also suggests not to try and "swallow it all at once" and Highsmith & Cockburn (2001) present twelve practices as generative rules stating that they are rather giving the "minimum" of what needs to be put into action rather than a vast amount of anything that could be done. It is interesting that the "traditionalists" presume a more prescriptive use of agile methodologies than is initially intended by its inventors.

One point that is relevant for the debate and the establishment of credibility to agile methodologies can be put under the heading of an "image problem": the 17 signers of the Agile Manifesto refer to themselves as "anarchists" (Fowler & Highsmith, 2001) and XP, the most successful methodology so far, is facing reluctance of managers to adopt "extreme" things (Boehm, 2002). Looking at the core of each side, the differences in argumentation can be assigned to differing views and assumptions about the organization and the problems it faces.

Last but not least, there is the third position, arguing that agile and traditional methods will become more symbiotic in the sense that people who work on a certain project are capable to select which parts of which methodology are most appropriate without regarding any of them as "the best" per se (Dybå & Dingsøyr, 2008). Having a look at the use of methodologies in practice, it becomes clear that there is much modification and omission as well as a kind of "mix and match" of seemingly contradicting methods or just a general limited use in practice (Fitzgerald, 1997; Truex et al., 2000; Introna & Whitley, 1997). Boehm (2002) also belongs to those arguing for a mix, claiming that hybrid approaches are feasible as agile and plan-driven methods form part of the spectrum from which developers can draw.

Some regard agile methodologies as a step towards "anti-methodology" (Avison & Fitzgerald, 2006). This statement leads us to the work of Introna & Whitley (1997) in which they are exploring the limits of method. They do not reject methodology in general but are postulating adjustments to the use of methodology. They argue to give up our thinking that there is one single methodology that will serve our needs, as well as to be aware of the importance of background understanding for the use, i.e. we have to be able to see methodologies in context - this is what they proclaim the new, "involved developer" should be able to do. They are also making us aware of the fact that methodology can be a "means to itself" in many cases, as to having a psychological effect when attaining a new client, for example. In addition, they provide the insight that one major, unquestioned belief prevalent for the use of method in information systems development is that methodology is a necessary and sufficient requirement for successful systems development, but they also point at the often-made, faulty assumption that methodologies are value-free. The Truex et al. (2000) paper raises also interesting questions as to whether we are able at all to think outside the "box" of methods, as we are prone to adopting a domineering concept. This also becomes evident in Abrahamsson et al. (2003) as their research was also aiming at finding out whether and where agile methodologies cover phases of the lifecycle model – so even with a new methodology evolving, the thinking about "thinking about systems development" in terms of a lifecycle makes its way through. They want to raise awareness that being too obsessed with method can result in us ignoring activities that do not fit within a methodical frame (Truex et al., 2000). As pointed out in Parnas & Clements (1986), one of the "marginalized texts" according to Truex et al. (2000), even if during the development phase there has been or is going to be no strict adherence to methods, there is still a "good" reason for pretending to have some in place, as guidance is seen as a major requirement for designers.

One thing that we can state is that a common base of understanding within the IS community is the necessity for planning – this still leaves room for interpretation as to the required extent and serves as major source for discussion between traditionalists and agilists. We can also notice that pure "a-methodical" systems development in the literal sense as Truex et al.'s work (2000) suggests at first glance is not imaginable in an organizational context.

To put the state of debate into a nutshell, this part has first presented the points of the two contrasting views of agilists and traditionalists on agile methods. Rather than treating methodologies as mutually exclusive, this paper has tried to show a shift of focus in research by introducing a third perspective with a flexible view on methodologies that is not asking the either-or-question but more the question of why not using both.

Limitations and further research

This paper did not give too many details on which agile methodologies are available and what practices are related to them. First, it has been assumed that the reader is already familiar with the concept and second, the focus chosen made it necessary to narrow down the level of detail in order to be able to present the types of literature available on the topic as well as arguments of the discussion on the use of methodologies in a broader sense. Taking a look at the literature available, a general lack of "academic rigour" can be recorded as well as a lack of a standardized or coordinated way of researching XP, as this is the agile methodology that has been most written about. Also a more interdisciplinary work would be desirable as Smith & Pichler (2005) note a gap between information systems and project management literature.

Conclusion

We have seen that information systems development research has strongly been coupled with the methods used for the construction of systems (Truex et al., 2000). Changes in the organizational contexts have led to changes in approaches to methodologies. We have also encountered a paradigm shift: Some literature promotes to be more open to a flexible use of methodologies. This has future implications for the work of developers. They will have to be much more involved with their work – apart from their technical abilities, they need to be familiar with a set of tools and methodologies available, as well as to judge which ones are the most appropriate ones for a particular development project. This gives way to a more pragmatic view, as also suggested by Hevner et al. in their design theory, where evaluation is based on whether the application works and whether the result is an improvement.

References

Avgerou, C. (2000). Information Systems: What sort of science is it?. Omega, 28, 567-579.

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

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: A comparative analysis. 25th International Conference on Software Engineering. (p. 244). IEEE.

Andersen, A. (1998). Chrysler goes to "Extremes". Distributed Computing, 1(10), 24-28.

Avison, D. & Fitzgerald, G. (2003). Where now for development methodologies?. Communications of the ACM, 46(1), 79-82.

Avison, D. & Fitzgerald, G. (2006). Information systems development: Methodologies, techniques and tools. (Vol. 4th ed.,). Berkshire: McGraw-Hill Education.

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

Cockburn, A. (2007). Agile software development: The cooperative game. (p. 199-206). Addison-Wesley.

Drobka, J., Noftz, D., & Raghu, R. (2004). Piloting XP on four mission-critical projects. IEEE Software, 21(6), 70-75.

Dyba, T. & Dingsoyr, T. (2008). Empirical studies of agile software development: A systematic review. Information and Software Technology, 50, 833-859.

Fitzgerald, B. (1997). The use of systems development methodologies in practice: A field study. Information Systems Journal, 7, 201-212.

Fowler, M. & Highsmith, J. (2001, Aug.). The Agile Manifesto. Software Development Magazine, Retrieved Nov. 20, 2008, from http://hristov.com/andrey/fht-stuttgart/The_Agile_Manifesto_SDMagazine.pdf

Gordon, S. & Gordon, J. (2004). Information systems: A management approach. In Agile programming (p. 321-322). Hoboken: John Wiley & Sons.

Hevner, A.R., March, S.T., Park, J. Ram, S. (2004) 'Design science in information systems research' MIS Quarterly, 28(1), 75-105.

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

Introna, L. D. & Whitley, E. A. (1997). Against method-ism: Exploring the limits of method. Information Technology & People, 10(1), 31-45.

Nerur, S., Mahapatra, R. K., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. Communications of the ACM, 48(5), 73-78.

Parnas, L. & Clements, P. C. (1986). A rational design process: How and why to fake it. IEEE Transactions on Software Engineering, 12(2), 251-257.

Smith, P. G. & Pichler, R. (2005). Agile risks / Agile rewards. Software Development, 13(4), 50-53.

Truex, D., Baskerville, R., & Travis, J. (2000). Amethodical systems development: The deferred meaning of systems development methods. Accounting, Management and Information Technologies, 10, 53-79.

Turk, D., France, R., & Rumpe, B. (2002). Limitations of agile software processes. Third International Conference on Extreme Programming and Flexible Processes in Software Engineering. (p. 43-46). Alghero:

Williams, L. & Kessler, R. (2000). All I really needed to know about pair-programming I learned in kindergarten. Communications of the ACM, 43(5), 108-114.

Wynekoop, J. & Russo, N. (1997). Studying system development methodologies: an examination of research methods. Information Systems Journal, 7, 47-65.