Agility in information systems development: characterisation, motivation and conceptualisation

Werner Keplinger
Candidate for M.Sc. in Analysis Design and Management of Information Systems
Information Systems and Innovation Group
Department of Management
London School of Economics

Agility in information systems development has recently received a considerable amount of attention. It is seen as a radical response to the difficulties organisations face when following a rigid traditional software development approach in the context of constantly evolving requirements in an increasingly competitive business environment. Although practiced in industry for years, agility is still an elusive concept and researchers have only just started to conceptualise it. This paper reviews literature on how software engineering accounts for agility by delineating agile and traditional methodologies. Motivations for agile methodologies, given by information systems literature are reviewed and attempts of conceptualising agility are presented. The theoretical underpinnings used by researchers to conceptualise agility are found to be very diverse and range from military, manufacturing, complex adaptive systems theory to information technology innovations and organisational learning. Finally a novel way of investigating agility is briefly presented by linking it to the new socio-technical view on systems development.

Introduction

The highly competitive business environment organisations operate in is characterised by constant change. In order to manoeuvre through or even take advantage of the turbulences created by internal and external change factors, organisations are increasingly confronted with the need to respond and adapt in a quick and resourceful fashion – they need to be *agile* (Mathiassen & Pries-Heje, 2006). Information technology (IT) has a crucial, and at the same time ambivalent role with regards to agility: it can be an enabler and a disabler (van Oosterhout *et al.*, 2006). While a malleable and simple IT infrastructure may support an organisation in its attempts to reconfigure its capabilities and business processes in order to take advantage of expected and unexpected turbulences, a rigid and complex IT infrastructure may hinder those efforts. Organisations developing software or software-supported products face additional challenges, as not only their IT infrastructure, but also their software development processes need to be agile (Mathiassen & Pries-Heje, 2006).

As a reaction to numerous problems with traditional information systems development (ISD) approaches, new methodologies incorporating agility have emerged and recently received a considerable amount of attention within the fields of software engineering and information systems (IS). A catalyst was the publication of the Agile Manifesto (Beck *et al.*, 2001), which lays down the main values of agility from a practitioners' point of view. The first part of this literature review introduces what agility in ISD is. Then follows a discussion of the main motivations for agility as identified in the literature. In the third part different attempts of conceptualising agility are put forward. Finally a conclusion is presented, which reflects on the reviewed literature and proposes directions for further development.

What is agility in ISD

New and rather unorthodox ISD approaches emerged at the end of the 1990s and indicated the beginning of a new era, a '*post-methodology era*' (Avison & Fitzgerald, 2003), or better a post-traditional-methodology era. Those new approaches were informally referred to as lightweight approaches to distinguish them from the traditional, well-established, heavyweight engineering methodologies they sought to challenge (Fowler, 2006). Although the lighter methods, among which the most prominent were and still are Extreme Programming (Beck, 1999) and Scrum (Schwaber & Beedle, 2001), had different emphases and priorities, they shared a set of common principles, which were expressed by their proponents, a group of independent practitioners, in the coining of the term agile for their methods and in the subsequent publication of the Agile Manifesto:

> *"Individuals and interactions over processes and tools*
> *Working software over comprehensive documentation*
> *Customer collaboration over contract negotiation*
> *Responding to change over following a plan"*
> (Beck, 1999)

Although the Agile Manifesto gives evidence of when a methodology may call itself agile and when it may not, it does not actually define the term. In fact, there does not seem to be a generally accepted definition of agility in literature (Conboy & Fitzgerald, 2004). However, as agile methodologies are a countermovement to traditional methodologies, reviewing how software engineering literature delineates both approaches in terms of process, product and people, presents an insightful picture of agility and yields an implicit definition.

Process

A process, in the context of an ISD methodology, defines a strict sequence of phases and a specific set of conditions that need to be met before moving on to the next phase (Boehm, 1988). Traditional methodologies are fully specified and steered by a statistical process control model (Humphrey, 1988). Different aspects of the process are measured and used as input to stabilize the process and make it repeatable, so that it delivers the same result when executed repeatedly, as with industrial processes. In contrast to traditional approaches agile processes are adaptive and follow an empirical process control model (Schwaber & Beedle, 2001). The loosely defined agile process is steered towards a desired end by constantly assessing and redirecting the process, without aiming for repeatability, as the environment is assumed to be volatile.

People

Traditional methodologies define roles for project members, for example analyst, developer or tester (Fowler, 2005). Those roles and their associated fully specified tasks drive the project forward, from one phase to the next. A good process

is seen as the critical success factor (Humphrey, 1988). Agile methodologies are different; they do not regard the process, but the people, who work as a team to achieve a desired outcome, as crucial for success or failure (Cockburn & Highsmith, 2001). A premium is therefore placed on the excellence and superior capabilities of the project members to thrive in the uncertainties of an adaptive process. However, as Boehm (2002) cynically points out almost half of all software developers are below average. Another peculiarity of agile methodologies is the strong collaboration with customers. While traditional methodologies limit direct contact to and collaboration with customers to the early upfront specification of the requirements, agile methodologies aim to have a customer as a permanent member of the team (Baskerville *et al.*, 2006, Beck, 1999, Schwaber & Beedle, 2001). It is difficult however, to convince a client organisation to give away an employee with high domain knowledge and enough authority to push decisions as Nerur *et al.* (2005) point out.

Product

Traditional methodologies aim at delivering at once at the end of a project a fully-fledged product, which hopefully meets the set out specification. Agile methodologies are different; they frequently deliver working software to the customers, who see immediate benefits and are given the opportunity to reprioritise their requirements and to propose new ideas for further development (Beck, 1999, Schwaber & Beedle, 2001). The product is delivered quickly in a base version and then growing in an organic way (Truex *et al.*, 1999).

Delineating agile and traditional methodologies using literature by the creators of agile methodologies (Beck, 1999, Beck & Andres, 2005, Cockburn, 2002, Cockburn & Highsmith, 2001, Fowler, 2005, Schwaber & Beedle, 2001) conveys a somewhat uncritical view towards the new approaches and their advantages. A more balanced view is given by Boehm (2002), who argues that both traditional and agile methodologies have their '*home grounds*', in which they perform best. Nerur *et al.* (2005) also take a more unbiased stance towards agile methodologies and point at challenges and problems ISD organisations face when they adopt the agile approach.

Motivations for agility

Literature by creators of agile methodologies often very broadly identifies change as the key motivation to be agile (Beck, 1999, Cockburn & Highsmith, 2001). The IS literature on system development is richer and identifies various push factors for agility, which are all directly or indirectly related to a mismatch of traditional methodologies with modern organisations.

Traditional rational software design processes, as Parnas & Clements (1986) argue, can hardly be strictly followed. They are too idealized and do not reflect the inherent uncertainties of our world and the limitations of human mental capacity. However, it can be tried to follow a rational process as closely as possible and where workarounds or deviations from the ideal process are needed, the required intermediate results can be faked. This argument is supported and enhanced by Nandhakumar & Avison (1999) who contend that traditional rational ISD methodologies are merely a '*necessary fiction*' to simulate being in control of a world full of uncertainties, in which improvisations and workarounds are an ongoing practice and exist tacitly within the rational methodology. These workarounds and improvisations can be seen as '*amethodical*' (Truex *et al.*, 2000) agile practices which have always existed within traditional methodologies. They have just been distilled and moved into the foreground by agile methodologies as Pries-Heje *et al.*(2004) indicate.

Truex *et al.*(1999) see the fast moving business climate driven by globalisation and technological advances as factors undermining the stability of an organisation. Organisations are in constant motion, they are '*emergent*'. This transformation of the nature of an organisation has implications on IS and redefines the way how IS need to be developed. It is argued that systems must evolve with their organisational environment and as a result systems must be under constant development. Users are provided with systems as soon as possible and then the systems grow organically, while virtually being in perpetual development and maintenance. Such a development model can hardly be realised with conventional traditional ISD methodologies as long-term planning and meticulous up-front specification are not possible. Agile methodologies however, fit such a development model as they regard organisations as complex adaptive systems with emergent properties (Highsmith & Cockburn, 2001). The '*emergent organisation*' can thus be seen as a push factor for agility in ISD.

Technological characteristics have changed in the last decades, from monolithic mainframe computers to the distributed nature of the internet. Agility in ISD is often associated with the rise of the internet (Aoyama, 1998, Baskerville *et al.*, 2003, Baskerville *et al.*, 2006). The effects of the internet on ISD are visible in two dimensions - on the one hand as a business driver in the form of a reduced time-to-market, on the other hand as a technological driver, which has brought about a novel technological platform and provided developers with tools to quickly develop applications. In fact Baskerville *et al.* (Baskerville *et al.*, 2003) note that ISD for the internet, has led to a strong dependence of developers on tools, which is somewhat in contrast to the Agile Manifesto, which plays down the importance of tools.

Higher-order agility, such as '*enterprise agility*'(Overby *et al.*, 2006), i.e. '*the capability of quickly sensing and responding to change*', can also be identified in the literature as a driver for agility on a lower level, i.e. the IS production level. Overby *et al.* (2006) argue that the responding capability can be improved by rapid and cost-efficient systems development. Agile methodologies match these systems development characteristics better than traditional approaches and can be seen as an enabler for enterprise agility.

Conceptualisation of agility

Although chief information officers strongly feel they need agility, their notions of agility differ and they have difficulty pinning down the term (Schrage, 2004). A reason for this can be seen in the missing theoretical underpinning of agility in ISD as Conboy & Fitzgerald (2004) point out. Theory in ISD has historically had problems keeping up with best practice in industry (Fitzgerald, 2000). In the case of agility this seems to be validated once again. Slowly, however theory tries to catch up with practice as scholars embark on conceptualising agility.

Conboy & Fitzgerald (2004) try to broadly conceptualise agility by drawing from manufacturing and systems thinking. By combining principles of flexible and lean manufacturing they

propose a definition of agility:

> *"…the continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment".*

The underlying aim of agility in ISD is identifying and coping with change, according to Conboy & Fitzgerald (2004). They propose an agility framework for ISD organisations, which identifies four main activities in regards to handling change: '*creation*', '*proaction*', '*reaction*' and '*learning*'. Accordingly, an agile ISD organisation is supposed to handle change as early as possible by actively creating change and proactively eliciting change. Being surprised by change and having to react to change is to be avoided. The learning activity aims at improving the change handling capabilities over time. A similar avenue is taken by van Oosterhout *et al.* (2006), who enhance the concept of flexibility to define agility as having the capacity to quickly respond to familiar and unfamiliar changes. The definition of agility given in Oosterhout *et al.* (2006) is however at enterprise-level and not as in Conboy & Fitzgerald (2004) at process-level. An agile IT-infrastructure facilitated by agile ISD is seen by van Oosterhout *et al.* (2006) as an enabler for higher-level agility. Fowler (2005) calls for prudence in terms of drawing analogies between manufacturing and ISD as Conboy & Fitzgerald (2004) and van Oosterhout *et al.* (2006) do, since manufacturing and its underlying scientific management paradigm have historically brought many problems into the ISD domain, such as a strict separation of design, implementation and test for instance.

A different and rather unorthodox approach to conceptualise agility in ISD is taken by Adolph (2006). He draws on military literature, in which a fighter pilot is regarded as agile and as likely winner of an air combat, if she runs an '*observation, orientation, decision and action (OODA) – loop*' faster than her opponent. Accordingly, development teams that can orient themselves quickly and make fast decisions in times of uncertainty, will be able to take better action and will be more likely to survive in the fast-paced business world. A corporate culture facilitating agility is presented in terms of the German Blitzkrieg, where the principles of '*Einheit* (unity or trust)', '*Fingerspitzengefühl* (skill or expertise)', '*Austragstaktik* (intent)' and '*Schwerpunkt* (vision)', are claimed to help an agile team to thrive in an uncertain and unpredictable environment.

Highsmith & Cockburn (2001) argue that agility in ISD is based on the worldview that organisations are complex adaptive system. That thought is taken up by Meso & Jain (2006), who describe the concept of agility by mapping agile practices to complex adaptive systems theory principles. Especially phenomena like emergent requirements, growing systems or self-organising teams find a strong theoretical foundation in the rich and well-developed literature on complex adaptive systems theory.

Lyytinen & Rose (2006) make a case that agility in ISD can be conceptualised as having the dynamic capability to balance the organisational learning concepts of exploration and exploitation in order to match the level of innovation and speed required within the ISD process. Speed and innovation are seen as conflicting goals. During exploration ISD organisations sense and adopt innovations from organisations deploying their products and from organisations delivering base technologies. During exploitation ISD organisations strive to speed up their delivery process by incorporating and adapting the acquired innovations. As agility is often associated with speed, Lyytinen & Rose (2006) argue that agile ISD methodologies (Beck, 1999, Schwaber & Beedle, 2001) may improve the exploitation capabilities of an ISD organisation, but are of little help to drive exploration. A contrasting view is provided by Vinekar *et. al.* (2006), who regard agile methodologies as appropriate for supporting the exploration ambitions of an ISD organisation, which take place in a volatile environment. Traditional methodologies are associated with exploitation activities, which are assumed to be executed in a stable environment. As an organisations' environment can have both, fast changing parts and relatively more stable parts, they propose an '*ambidextrous*' ISD organisation, which has agile development units as well traditional development units.

Conclusion and reflections

Early research in and descriptions of agility in ISD have predominantly come from the discipline of software engineering and from practitioners, who saw the short comings of the traditional highly formalised methods in a changed and highly accelerated business world. The first publications about agility in ISD were a novel combination of best practices based on simple values (Beck, 1999, Beck *et al.*, 2001, Cockburn & Highsmith, 2001, Schwaber & Beedle, 2001). Many of those best practices and values are rooted in the social sciences and are not of a technical nature. The movement from traditional formalised ISD to agile ISD can therefore be seen as an interpretivist one (Casey & Brugha, 2005). Recently IS research has taken up agility in ISD and made attempts to ground it in theory. The range of theoretical links and underpinnings is very diverse and includes military (Adolph, 2006), complex adaptive system theory (Meso & Jain, 2006), flexible and lean manufacturing (Conboy & Fitzgerald, 2004) and organisational learning and IT innovation (Lyytinen & Rose, 2006). IS research on agility in systems development has just started and the concept is still elusive. Further efforts are needed to enhance existing IS research topics to accommodate agility in systems development or to link agility in systems development to enduring IS themes, such as the new socio-technical approach to systems development for instance.

The author sees strong similarities between the agile approach to systems development and the new socio-technical view on systems development. In agile ISD, system developers and users find themselves within a continuous feedback loop, try to make sense of the changing organisational environment and do not believe in objective pre-specifiable requirements. Users participate in the agile design process as much as developers. They are encouraged to feed back new requirements and ideas while the system is incrementally growing. Systems in agile ISD are in an ongoing state of design, as the traditionally separated activates of analysis, design and implementation are practically merged and executed in very short cycles (Beck, 1999). The new socio-technical view on systems development similarly stresses that systems are to be designed and shaped '*in-use*' (Lin & Cornford, 2000) and are to co-evolve with ever changing work practices (Berg, 1999). Also Ciborra's (2002) notion of '*hospitality*' can be found in the agile approach to ISD. Users and devel-

opers align themselves with the growing systems. They decide in the context of the current situation with no meticulous long-term plans and formalisations imposed on them and they have the '*negative capability*' (Ciborra & Lanzara, 1994, quoted in Ciborra 2002) to thrive in uncertainty. It is thus very surprising that scholars have not yet drawn from socio-technical literature to explain and conceptualise agility in ISD.

References

Adolph, S. (2006) "What Lessons Can the Agile Community Learn from a Maverick Fighter Pilot?" in Proceedings of Agile Conference 2005, Minneapolis, Minnesota, USA, pp. 94-99, IEEE.

Aoyama, M. (1998) "Web-Based Agile Software Development", IEEE Software, 15 (6), pp. 56-65.

Avison, D. E. and G. Fitzgerald (2003) "Where Now for Development Methodologies?" Communications of the ACM, 46 (1), pp. 79-81.

Baskerville, R., L. Levine, J. Pries-Heje and S. Slaugher (2003) "Is Internet-Speed Software Development Different?" IEEE Software, 20 (6).

Baskerville, R., B. Ramesh, L. Levine and J. Pries-Heje (2006) "High-Speed Software Development Practices: What Works, What Doesn't", IT Pro, July/August.

Beck, K. (1999) "Embracing Change with Extreme Programming", Computer, 32 (10).

Beck, K. and C. Andres (2005) Extreme Programming Explained, Pearson Education Inc., Upper Saddle River.

Beck, K., et al. (2001) Manifesto for Agile Software Development Last accessed: 25.11.2006 Last updated: Address: http://www.agilemanifesto.org/.

Berg, M. (1999) "Patient Care Information Systems and Health Care Work: A Sociotechnical Approach", International Journal of Medical Informatics, 55 pp. 87-101.

Boehm, B. (1988) "A Spiral Model of Software Development and Enhancement", Computer, 21 (5), pp. 61-72.

Boehm, B. (2002) "Get Ready for Agile Methods, with Care", Computer, 35 (1), pp. 64-69.

Casey, D. and C. M. Brugha (2005) in Action in Language, Organisations and Information System ALOIS*2005 Limerick, Ireland.

Ciborra, C. (2002) "Xenia - Chapter 6" in Labryrinths of Information, (Ciborra, C. ed.) OUP, Oxford.

Ciborra, C. and G. F. Lanzara (1994) "Formative Contexts and Information Technology: Understanding the Dynamics of Innovation in Organisations", Accounting, Management and Information Technology, 4 (2), pp. 61-86.

Cockburn, A. (2002) "Agile Software Development Joins the 'Would-Be Crowd'", Cutter IT Journal, 12 (1), pp. 6-12.

Cockburn, A. and J. Highsmith (2001) "Agile Software Development: The People Factor", Computer, November 2001.

Conboy, K. and B. Fitzgerald (2004) "Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines". in Proceeding of ACM Workshop on Interdisciplinary Software Engineering Research (WISER), Newport Beach, CA, USA,

Fitzgerald, B. (2000) "Systems Development Methodologies: The Problem of Tenses", Information Technology and People, 13 (3), pp. 174-185.

Fowler, M. (2005) The New Methodology Last accessed: 25.11.2006 Last updated: 13.12.2005 Address: http://martinfowler.com/articles/newMethodology.html.

Fowler, M. (2006) The Agile Manifesto: Where It Came from and Where It May Go Last accessed: 25.11.2006 Last updated: July 2006 Address: http://martinfowler.com/articles/agileStory.html.

Highsmith, J. and A. Cockburn (2001) "Agile Software Development: The Business of Innovation", Computer, 34 (9), pp. 120-122.

Humphrey, W. S. (1988) "Characterizing the Software Process: A Maturity Framework", IEEE Software, 5 (2), pp. 73-79.

Lin, A. and T. Cornford (2000) "Sociotechnical Perspectives on Emergence Phenomena" in The New Sociotech: Graffiti on the Long Wall, (Cookes, E., D. Willis and R. Lloyd-Jones eds) Springer, Godalming.

Lyytinen, K. and G. M. Rose (2006) "Information System Development Agility as Organizational Learning", European Journal of Information Systems, 15 (2), pp. 183-200.

Mathiassen, L. and J. Pries-Heje (2006) "Business Agility and Diffusion of Information Technology", European Journal of Information Systems, 15 (2), pp. 116-120.

Meso, P. and R. Jain (2006) "Agile Software Development: Adaptive Systems Principles and Best Practices", Information Systems Management, 23 (3), pp. 19-30.

Nandhakumar, J. and D. E. Avison (1999) "The Fiction of Methodological Development: A Field Study of Information Systems Development", Information Technology and People, 12 (2), pp. 176-191.

Nerur, S., M. RadhaKanta and M. George (2005) "Challenges of Migrating to Agile Methodologies", Communications of the ACM, 48 (5), pp. 73-78.

Overby, E., A. Bharadwaj and V. Sambamurthy (2006) "Enterprise Agility and the Enabling Role of Information Technology", European Journal of Information Systems, 15 (2), pp. 120-132.

Parnas, D. L. and P. C. Clements (1986) "A Rational Design Process: How and Why to Fake It", IEEE Transactions of software engineering, pp. 346-357.

Pries-Heje, J., R. Baskerville, L. Levine and B. Ramesh (2004) "The High-Speed Balancing Game. How Software Companies Cope with Internet-Speed." Scandinavain Journal of Information Systems, 16 pp. 11-54.

Schrage, M. (2004) The Struggle to Define Agility Last accessed: 11.11.2004 Last updated: Address: www.cio.com.

Schwaber, K. and M. Beedle (2001) Agile Software Development with Scrum, Prentice Hall, Upper Saddle River.

Truex, D., R. Baskerville and J. Travis (2000) "Amethodical Systems Development: The Deferred Meaning of Systems Development Methods", Accounting, Management and Information Technology, 10 (1), pp. 53-79.

Truex, D. P., R. Baskerville and H. Klein (1999) "Growing Systems in Emergent Organizations", Communications of the ACM, 42 (9), pp. 117-123.

van Oosterhout, M., E. Waarts and J. van Hillegersberg (2006) "Change Factors Requiring Agility and Implications for It", European Journal of Information Systems, 15 (2), pp. 132-146.

Vinekar, V., C. W. Slinkman and S. Nerur (2006) "Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View", Information Systems Management, 23 (3), pp. 31-42.