Accepted to the 17th International Conference on Information Systems Development (ISD2008). Paphos, Cyprus, August 25-27, 2008.

Using Agile Methods? – expected effects

Stefan Cronholm

Department of Management and Engineering, Linköping University, Sweden.
stefan.cronholm@liu.se

Abstract: This paper focuses on the movement from traditional to agile methods. What are the expected benefits of using agile methods instead of traditional ones? The paper compares identified benefits in traditional and agile methods and takes a critical attitude in order to reveal possibilities and risks with the expressed benefits in agile methods. The paper also tries to answer the questions of what benefits are lost and what benefits are preserved when moving to agile methods**.**

Keywords: agile methods, traditional information system development methods

1. Introduction

This paper focuses the concept of software methods and on the movement from traditional methods to agile methods. This movement is based on a criticism of traditional methods. According to Nandhakumar & Avison (1999) traditional methods are too mechanistic to be used in detail. Truex et al. (2000) are more dogmatic and claim that traditional methods are merely unattainable ideals and hypothetical "straw men" that provide normative guidance to utopian situations. Furthermore, methods that are different to grasp will remain unused (Wiegers, 1998). Baskerville et al. (2003) claim that "to compete in the digital economy, companies must be able to develop high-quality software systems at "Internet speed"—that is, deliver new systems to customers with more value and at a faster pace than ever before".

The common understanding of traditional methods is that they requires too much of planning activities, are too sequential and there are too much work with documentation. These experiences could be seen as anomalies that need attention. In this way, development of agile methods could be seen as cumulative, that is, agile methods are built on existing traditional methods where the 'good' parts are kept and the bad parts are omitted or replaced.

On the other hand, the enormous popularity of agile methods does not justify an uncritical review. These methods should be reviewed and criticized in the same way as other type pf methods are reviewed. There are provocative statements that motivate a review such as: is this popularity to that system developers don't need to care about boring documentation and instead can get an outlet for their creativity? How could Scrum be seen as a panacea against delays and increased costs? (Juell-Skiels, 2007).

The aim of this study is to identify and compare the differences between expressed effects in traditional and in agile methods in order to present expected effects. The research question therefore reads: what are the expected effects of using agile methods? An effect is defined as “a change which is a result or consequence of an action or other cause” (AskOxford, 2008). Cambridge Dictionaries (2008) provides a similar definition: “the result of a particular influence”. The “change” or the “particular influence” is in this case the movement from traditional methods to agile methods.

The concept of traditional methods has been discussed for several decades (i.e. Checkland, 1981; Yourdon, 1989; Jayaratna, 1994; Kruchten, 1999; Ambler, 2002; Juell-Skiels, 2007). It is hard to find one common definition of the concept of traditional methodology. According to Jayaratna (1994), a methodology provides “an explicit way of structuring one’s thinking and actions”. It is also hard to find a handy definition of the concept of agile method. A broad definition is presented by Cockburn (2002) who defines the process of agile development as the “use-of-light-but-sufficient-rules”. According to Abrahamsson et al. (2002) the academic research on the topic of agile methods is still scarce, most of the publications are written by consultants or practitioners.

After this introductory section a brief description of the research method is presented in section 2. Section 3 presents identified effects of the traditional respectively the agile methods. Finally, in section 4 the conclusions are presented.

2 Research approach

The idea is to identify and compare expected effects of using agile methods and of using traditional methods. In order to answer the research question the research has been carried out in three steps: 1) identify and analyze expected effects in traditional methods, 2) identify and analyze expected effects in agile methods, 3) compare identified effects in order to understand what are the “added effects”, “lost effects” and “preserved effects”. Expected effects are identified in the literature by searching for statements or utterances about traditional and agile methods. The term “expected effect” is used since the effects identified are presented in literature. If the effects would have been identified from real development projects the term perceived would have been used. The research approach can be characterized as inductive, but theoretical support is used in order to explain and define the identified effects.

The utterances from the traditional methods have been gathered from familiar and well-known method contributions such as Object Modeling Technique (OMT), (Rumbaugh, 1991), Structured Analysis and Structured Design (SASD), (Yourdon, 1989) and Soft Systems Methodology (SSM), (Checkland & Scholes

(1999). The gathering of utterances from the agile methods has started by reading the principles behind the "Agile Manifesto" (Beck et al., 2001). Important contributions to the concept of agile have then been traced from the Agile Manifesto. Examples of agile methods analyzed are: Rational Unified Process (RUP), (Object Management Group, 2008), Extreme programming (XP), (Beck 1999), Scrum (Schwaber, 1995), Crystal (Cockburn, 2002) and Adaptive Software Development (ASD), (Highsmith, 2000).

Finally, the identified effects are compared by using a Venndiagram where three sets are identified: 1) Effects that only exist among the agile methods – this is an added effect! 2) Effects that only exist among traditional methods – this is lost effect! 3) The intersection between the two method types consists of effects that exist among both the method types – this is a preserved effect (see fig. 1).

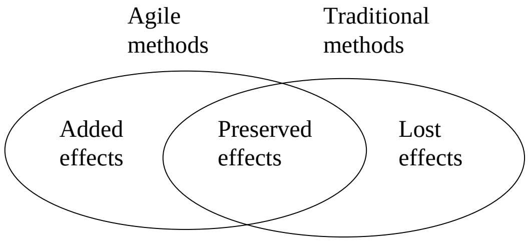

Fig. 1. The relation between identified effects

According to Abrahamsson et al. (2002) there are no sharp boarders between traditional and agile methods. Of course there can be overlapping parts, but all the chosen methods could primarily be categorized as traditional or agile. An effect could be viewed as a benefit, as a drawback or as both. If an identified effect should be valuated as a benefit or a drawback is out of the scope of this paper. Such a judgment has for example to consider: the actual context where the method will be used, the maturity of the method users, the subjective attitudes of the method users and the managers´ (and other stake holders) perspective on the development process.

3 Findings

3.1 Expected effects of traditional methods

The identified expected effects of the traditional methods are: governing, 'rationality', 'structure', 'standardization' and 'flexibility'. The first identified effect is ‘governing’. ‘Govern’ is defined as “to have a controlling influence on something” (Cambridge Dictionaries Online, 2008). Goldkuhl & Lyytinen (1982) claim that strategies and methods governing peoples’ actions in social practices. A method can be viewed as an instrument for action and an instrument governs the action in certain ways (Goldkuhl, 2005). To govern or to steer can be viewed both as something positive and something negative. For example when method users are confused and don’t know what or how to do there is of course a need for governing. On the other hand, ‘governing’ can be perceived as negative when the method users want to do something else than the method recommends. That is, when the method users have better action alternatives than the suggested ones. Too much governing can thereby lead to frustration.

The second identified effect is ‘rationality’. ‘Rationalism’ is defined as “the belief or principle that actions and opinions should be based on reason rather on emotions or religion” (Cambridge Dictionaries Online, 2008). Boland & Pondy (1983) claim that historically is software development characterized by rationality and the actions performed are justified on rational grounds. A software method is aiming at a rational process that does not contain any gaps. Weber (1978) talks about practical rationality that consists of three types of sub rationalities: instrumental rationality means the appropriateness of the means to given ends, rationality of choice means the setting of ends in relation to values and normative rationality means the evaluation and application of ethical principles in action. The three rationalities are put together under the label of methodical-rational conduct of life.

The third identified effect is ‘structure’. ‘Structure’ is defined as “the arrangement of and relations between the parts of something complex” (Oxford English Dictionary, 2008). Jayaratna (1994) defines “method” as “an explicit way of structuring one’s thinking and actions. A methodology should tell you ‘what’ steps to take and ‘how’ to perform those steps but most importantly the reasons ‘why’ those steps should be taken, in a particular order”. From this definition we can see that ‘structure’ is one important concept when discussing methods. Jayaratna (1994) tells us that using methods means to act in a structured way. A method informs about what, when, how and why something should be done.

The fourth identified effect is ‘standardization’. ‘Standardization’ is defined as “usual rather than special, especially when thought of as being correct or acceptable” (Cambridge Dictionaries Online, 2008). Cronholm (1995) claim that organizations adopt methods in order to achieve more standardized way of performance. One important purpose is to formalize actions in order to reduce undesired variation, and to control and to anticipate actions (Mintzberg, 1983). Organizations thereby views methods as an instrument for institutionalizing and are looking for uniform ways of structure the developing process through a wider use of methods. Using methods will also lead to simplified maintenance of documentation. Using method based and computer-supported tools will also lead to an increased standardization (Cronholm, 1995). Examples of such tools are Rational Rose Enterprise (IBM Software, 2008) and Systems Architect (Popkin Software, 2008).

The fifth identified statement is ‘flexibility’. ‘Flexible’ is defined as “able to change or to be changed easily according to the situation” (Cambridge Dictionaries Online, 2008). This means that the way the work will be performed should be able change according to the nature of the assignment. Since methods per see are instruments for standardization, the room for flexibility will of course decrease. Goldkuhl et al. (1998)] acknowledge that there are situations when the appropriateness of methods can be challenged. Many methods are developed with the aim of making them generally applicable. However, the differences in development circumstances seem to be extensive and also growing due to the general technological and organizational evolution. The diversity of development and change situations gives rise to a need to combine and integrate different methods (Harmsen, 1997). Method integration means to integrate two or more methods into a new one. The aim of this process is to achieve a method that is more suitable due to the problem that should be solved.

It seems that the expected effects of the traditional methods stems from a management and organizational perspective. The character of the effects is close to concepts as control and method institutionalizing. Traditional methods contain support for ‘governing’, ‘standardization’, ‘rationality’ and ‘structure’. At the same time there must be room for ‘flexibility’ which could be seen as a something opposite or a contrast.

3.2 Expected effects of agile methods

The identified effects of the agile methods are: ‘rationality’, ‘team work’, ‘adaptability’, ‘structure’, ‘less documentation’, ‘late changes’, ‘simplicity’, ‘creativity’ and ’improvisation’. The first identified effect is ‘rationality’ (see also section 3.1). Rationality´´ is discussed in terms of productivity and effectiveness Abrahamsson et al. (2002). An important meaning of ‘rationality’ is that the project should deliver software early in the process and frequent. The clients should gain value rapidly and therefore an incremental strategy is adopted. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly (ibid.). These findings are in line with Baskerville et al. (2003) who also have experienced that need of releasing features more often.

When most people are thinking of rationality they are thinking of being rational in order to reach the goals. The goals are usually expressed as functions or demands that the IT-system should posses. In the Dynamic Systems Development Method (DSDM) (1997) there is goal rationality but the goals are instead expressed in terms of fixed time and fixed resources. That means that the functionality of IT-system can be reduced (or increased) depending on how the process proceeds.

The second identified effect is ‘team work’. ‘Team work’ is defined as “when a group of people work well together” (Cambridge Dictionaries Online, 2008). A strong encouragement is that business people and developers should work closely to each other. Face-to-face conversations are encouraged and the project should be carried out with a constant pace where there are no delays or no interruptions (Abrahamsson et al. (2002). Furthermore, there should be self-organizing teams and developers should be trusted.

The third identified effect is 'adaptability'. 'Adapt' is defined as "to change something to suit different conditions" and "adaptability" is defined as "a necessary quality in an ever-changing work environment" (Cambridge Dictionaries Online, 2008). In the agile methods there is an imperative to be adaptive or flexible. One of the fundamentals ideas in Extreme Programming (XP) is that there is no process that fit every project as such, but rather practices should be tailored to suit the needs of individual projects (Beck, 1999). According to Grenning (2001), adoptability means that not all parts of the method are used. Rather, only parts that are needed should be chosen.

The fourth identified effect is 'structure' (see also section 3.1). Almost every agile method is divided into a sequence of different phase. For example the methods of the Crystal family contain guidelines of policy standards, work products "local matters", tool and standards and roles (Cockburn, 2002). There is also an imperative that methodology users should iterate. ASD is one of the agile methods that don't contain much structure. One of ideas is in ASD is to provide guidance enough to prevent projects from falling into chaos. That means that the absence of structure needs to be replaced by improvisation and creativity.

The fifth identified effect is 'less documentation'. The system developers are encouraged to keep the code simple and straightforward Abrahamsson et al. (2002). Related to this effect is the saying the agile method users should 'travel light'. According to Ambler (2002) one of the greatest misunderstandings about agile methods is that this means that you don't have to create any documentation at all. What it means is that the documentation burden is should be kept as minimal as possible (ibid.). This effect is line with the effect of 'rationality' since it supports the imperative of early deliverance.

The sixth identified effect is 'allow late changes'. McCauley (2001) argues that the underlying philosophy of process-oriented traditional methods is that the requirements are completely locked in and frozen before the succeeding phases. There is a need for flexible and adaptable methods that allow developers to make late changes in specifications. Highsmith & Cockburn (2001) mean that changes should be allowed throughout the whole development process. The utterance, it is better that developers are "responding to change over following a plan" (Beck et al., 2001), is in line with the previous ones. The use of iterations and prototyping is proposed means in order to allow late changes.

The seventh identified effect is 'simplicity'. The agile method user is encouraged to invent simple solutions (ibid.). The reason is that early solutions should be easy to change and that there should be less to change. Program code should be kept as simple as possible in order to support early delivery (Takeuchi. & Nonaka, 1986). Rapid evolution and speed goes hand in hand with simplicity.

The eighth identified effect is ‘creativity’. ‘Creativity’ is defined as “producing or using original and unusual ideas” (Cambridge Dictionaries Online, 2008). A recommendation is that agile methods should not offer too much guidance that suppresses emergence and creativity (Abrahamsson et al., 2002). According to Stolterman (1991), (traditional) method developers should accept that everything couldn’t be governed by rationality since you can’t predict which actions that will be successful. This statement is close to findings by Introna & Whitley (1997). They claim that there is a risk for an exaggerated belief in using methodologies and that those methodologies can’t bring light to all characteristics of the situation that need to be discovered.

The ninth identified effect is ‘improvisation’. ‘Improvisation’ is defined as “to invent or make something, such as speech or a device, at the time when it is needed without already having planned it”, (Cambridge Dictionaries Online, 2008). Stolterman (1991) claims that it is impossible to plan for and structure all future situations that can arise in an development project. There is a risk for over structuring work processes and there must be room for improvisation and spontaneity. Too much structure reduces the action possibilities and could also lead to a reduced job satisfaction. This over all message of ‘improvisation’ is that it is hard to (or even impossible) to structure everything in detail. Probably, it is not desirable. Improvisation can be seen as ad hoc work. That means that the development process is not based on a coordinated policy. According to Nandhakumar & Avison (1999) developers’ work practices have an improvisational character where team members are protective of their professional autonomy.

The analysis has revealed that agile methods are favoring effects such as adaptability, flexibility and less management control. The utterance “the clients should gain value rapidly” (Abrahamsson et al. 2002) brings forward the concept of client within a description of an effect. These effects could be seen as a reaction against most of the traditional methods. The followers of agile methods means there is risk of slavishly following all recommendations in traditional methodologies. You should always ask yourself questions like; what methodology parts are needed and why.

3.3 Comparison of effects

In order to gain an overview of the findings concerning the comparison the findings a summarized in the following list: the added effects are: flexibility (adaptability) less documentation, team work, allow late changes, simplicity, creativity and improvisation. The lost effects are: governing and standardization. Finally the preserved effects are: rationality and structure.

3.3.1 Added effects

The effect of ‘flexibility’ is identified in both the traditional and in the agile methods. Despite this, ‘flexibility’ is categorized as effect of the agile methods (see discussion in section 3.2). The occurrence of flexibility among the traditional methods could be questioned since the possibility to be flexible is quiet reduced. The corresponding effect in the agile methods is ‘adaptive’. To be adaptable is one of the foremost recommendations within the agile methods.

The effect of ‘less documentation’ could be seen as a reaction against traditional methods. Traditional methods are more documentation intensive since there are a lot of proposal for how to describe and illustrate analysis results. Those who are in favor of documentation procedures are usually claiming that documentation is not a routine-like work; it is an important analysis step. The main argument for documenting the IT-system is that the documentation will support maintenance work. ‘Traveling light’ is an imperative within the agile methods. It means that only necessary documentation should be produced. The implicit statement is that traditional methods produces unnecessary documentation.

The effect of ‘team work’ is one of the most high-lighted in the agile methods. Of course, it is possible to identify this effect within the traditional methods, but it is not emphasized in the same way. Another added effect is ‘late changes’. Late changes are something that is allowed and accepted in the agile methods. Keeping ‘things simple’ is a condition for the possibility to allow late changes. The ultimate idea in the traditional methods is that late changes shouldn’t be needed. Late changes are a sign of an insufficient requirements analysis.

The effect of ‘creativity’ is another added effect identified. Stolterman (1991) means that rationality could be viewed as a contrast to ‘creativity’. Furthermore, an open attitude toward the situated nature of development processes is important (Bratteteig and Stolterman, 1997). These perceptions are also in line with discussions of situated actions (Suchman, 1987). One underlying assumption for bringing forward creativity as a key effect is that the software development process is viewed as an artistic process (Ehn, 1989). Stolterman (1991) claims, that creativity has not been studied enough and is something that still occurs in the darkened.

Finally, ‘improvisation’ is seen as an added effect. ‘Improvisation’ is perceived as an opposite to ‘structure’. ‘Structure’ is one the foremost expected effects among the traditional methods and therefore could ‘improvisation be seen as an added effect among agile methods.. According to Abrahamsson et al. (2002) ‘improvisation’ is one issue that agile methods are dressing explicitly.

3.3.2 Lost effects

Both traditional and agile methods could be viewed as they are governing the method users since they consist of recommendations of what to do and how to do it. The difference is that traditional methods are governing in a more detailed way than the agile ones. As discussed in section 3.1 governing could be viewed as something positive and as something negative.

‘Standardization’ is one effect that not is identified among the agile methods. One main purpose with standardization is to formalize actions on a detailed level in order to reduce undesired variation and to control and to anticipate actions. Agile methods are very much the contradictory. There is a desire for variation! There is an imperative for being adaptive and flexible, to act according to the special conditions that exist in every project and to tailor projects to suit individual needs.

3.3.3 Preserved effects

‘Rationality’ is high-lighted both in traditional and agile methods. The concept is in both cases referring to the process; there is an imperative to be rationale. The main difference is that in agile methods the claim of the meaning of ‘rationality’ is that project should deliver software earlier and more frequent than in traditional methods. But, a too high focus on early delivery could lead to an unnecessary amount of iterations that instead will slow down the development process.

‘Structure’ is another concept that is common in both agile and traditional methods. Using (traditional) methods is to act in a structured way. A method informs about what, when, how and why something should be done (Jayaratna, 1994). There is also structure in the agile methods since there usually is a division into sequential phases. However, the structure is not that detailed and the method users are not encouraged to follow the structure in the same predicted way as in the traditional methods.

4 Conclusions

It is clear that many of the added effects in agile methods could be seen as a reaction of the traditional ones (see section 3.3.1). New demands, such as fast delivery, could be viewed as a protection against delayed software projects and increased costs. Tight couplings between the client and the development team with face-to-face meetings are another reaction that is done in order to secure quality. The added effects are examples of anomalies from the paradigm of traditional methods. The recommendation “fast delivery” is one of the most important. There is also a risk of being “too fast”. According to Cronholm (2005), early requirement specifications are not something that always is well-formulated or well-defined. Requirements need to mature and develop over time and therefore couldn’t the development process be shortened too much. It is important that the quality aspect is preserved.

The effect ‘Standardization’ is identified as a lost effect. Instead of ‘standardization’ concepts like variation, flexibility and adaptability are high-lighted. Standardization does not necessary has to be viewed as something “bad”. It is a well known fact that standardization provides advantages in terms of communication between project members and less vulnerability in projects. In addition, a standardized method or process is probably more supportive for novice method user than a non-standardized.

Another difference identified is that the individual and team aspect is more focused in agile methods. That is, there has been a movement of centralized management or organizational control towards decentralized units as teams or individuals. According to Nandhakumar & Avison (1999) traditional methods are used "as a necessary fiction to present an image of control or to provide a symbolic status". Furthermore, they report that developers' work practices have an improvisational character where team members are protective of their professional autonomy and where social control has more influence to their practice than the suggested methodologies. This increased attention to the individual and team aspect could be seen as an opposition to institutionalized processes which are viewed as a threat against adaptability and flexibility. One claim considering the team aspect reads: "Rationalistic and control-oriented design approaches may hamper innovative group design" (Bratteteig and Stolterman (1997).

On the other hand too much freedom, which is the negative side of 'professional autonomy', could cause chaos and uncontrollable projects. One example of the agile methods is ASD (Highsmith, 2000). ASD seems to contain very vague imperatives or recommendation for *how to do*. According to Abrahamsson et al. (2002), ASD is fundamentally about "balancing on the edge of chaos" and ASD is not providing detailed method support that could "suppress emergence and creativity". A general idea of a method is to provide clear recommendations of *how* to proceed. One definition of the concept of method reads "an approach to performing systems development projects, based on a specific way of thinking, consisting of directions and rules, structured in a systematic way in development activities with corresponding development products." (Brinkkemper et al., 1998). Based on this definition it seems doubtful if ASD could be classified as a method. Based on the effects identified, it seems like the traditional methods have a focus on the supplier (the developer) and are developed from a management and organizational perspective. The agile methods have focus both on the supplier and on the client (see section 3.2). That means that the agile methods provide an external focus as well as an internal focus.

Methods are usually based on earlier success stories, that is, they are based on successful experiences from practice. Furthermore, methods are not static "objects". New insights of how to develop IT-systems are continually experienced. The identified added, preserved and lost effects are an example of a cumulative method development. Acknowledging that method development is cumulative, another way of understanding the progress of agile methods is to apply the theories about Hegel's method of logic. The theory is based on the concept of advancing contradictory arguments, of thesis and antithesis (Croce, 1985). First, there existed no formal development methods (the thesis). The emergence of the traditional methods could be seen as a reaction to that no method support was available (the antithesis). Second, the anomalies/dissatisfaction with the traditional methods (the thesis) has worked as a condition/driver for constructing the agile methods (the synthesis). There has been a cumulative method development since there has been a movement from no methods, via traditional method to agile methods.

One message in this paper is that the agile methods should also be questioned and criticised in the same way as the traditional ones. Outcomes of reviews of agile methods will provide input to the ongoing process of cumulative method development.

References

Abrahamsson, P., Salo, O., Ronkainen, J., and Warsta, J. (2002). Agile Software Development Methods". VTT Electronic, Espoo. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf , site accessed Jan 26. 2008.

Ambler, S. (2002). Agile Modeling: Effective Practices for Effective Programming. Wiley & Sons, Inc, New York.

AskOxford.com (2008). http://www.askoxford.com/?view=uk. Site accessed Jan 15, 2008.

Baskerville, R., Levine, L., Pries-Heje, J., Ramesh, B., & Slaughter, S. (2003). Is Internet-speed software development different? IEEE Software, 20(6), 70-77.

Beck,K. (1999). Extreme Programming Explained: Embrace Change, Addison-Wesley, Reading, Mass.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K. and Sutherland, J, Thomas, D. (2001). Manifesto for Agile Software Development.http://agilemanifesto.org/, Site accessed Jan 26, 2008.

Boland, A. J. and Pondy, L. R. (1983). Accounting in organizations: a Union of Natural and Rational Perspectives, Accounting, Organizations and Society, 8, 223-234.

Bratteteig, T., Stolterman, E. (1997). Design in groups - and all that jazz. In Kyng, M. & Mathiassen, L. (eds.), Computers and Design in Context, The MIT Press, Cambridge, Mass.

Brinkkemper, S., Motoshi, S. and Harmsen, F. (1998). Assembly Techniques for Method Engineering. In Proceedings of the Tenth International Conference on Advanced Information Systems Engineering (CAiSE*98), Pernici B & Thanos C (eds.). Pisa, Italy.

Cambridge Dictionaries Online. (2008). http://dictionary.cambridge.org/. Site accessed Jan 15, 2008.

Checkland, P. (1981). Systems Thinking, Systems Practice, Wiley, Chichester, UK.

Checkland P. and Scholes, J. (1999). Soft Systems Methodology in Action. Wiley & Sons. Toronto.

Cockburn, A. (2002). Agile Software Development, Addison-Wesley, Boston, 2002.

Croce, B. (1985). What is Living and What is Dead of the Philosophy of Hegel, University Press of America.

Cronholm, S. (1995). Why CASE Tools in Information Systems Development?", In proceedings of the 18th Information Systems Research In Scandinavia (IRIS 18), Dahlbom B, Kämmerer F, Ljungberg F, Stage J & Sörensen C (eds). Gjern, Denmark.

Cronholm S. Goldkuhl G (2005). Communication Analysis as Perspective and Method for Requirements Engineering. In Requirements Engineering for Sociotechnical Systems, (José Luis Maté and Andrés Silva eds.). Idea Group Inc.

DSDM Consoritum. (1997). Dynamic Systems Development Method, version 3, Ashford, Eng.

Ehn, P. (1989). Work-oriented design of computer artifacts, Lawrence Erlbaum, Hillsdale, NJ.

Goldkuhl, G., Lind, M. and Seigerroth, U. (1998). Method integration: the need for a learning perspective. IEE Proceedings Software, Vol 145 No 4 pp 113-118.

Goldkuhl, G. and Lyytinen, K. (1982). A language action view of information systems. SYSLAB report no 14, SYSLAB, University of Stockholm, Stockholm, Sweden.

Goldkuhl, G. (2005). Socio-Instrumental Pragmatism: A Theoretical Synthesis for Pragmatic Conceptualisation in Information Systems, In Proceedings of the 3rd Intl Conf on Action in Language, Organisations and Information Systems (ALOIS), University of Limerick.

Grenning, J. (2001). Launching "XP at a Process-Intensive Company", IEEE, 18:3-9.

Harmsen, A. F. (1997). Situational Method Engineering. PhD dissertation. Moret Ernst & Young Management Consultants, Utrecht, The Nederlands.

Highsmith, J. A. (2000). Adaptive Software Development: A collaborative approach to Managing Complex Systems, Dorset House Publishing, New York, NY.

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

IBM Software. (2008). Rational Rose Enterprise. http://www-306.ibm.com/software/awdtools/developer/rose/ enterprise/index.html. Site accessed Jan 26, 2008.

Introna, L. D. and Whitley, E. A. (1997). Against method-ism. In Information Technology & People, 10 (1), 31-45. MCB University Press.

Jayaratna, N, (1994) Understanding and Evaluating Methodologies, McGraw-Hill Book Company, London.

Juell-Skiels, G. (2007). In Swedish: Irrläror frälser oss inte., Computer Sweden, Oct 29, 2007

Kruchten, P. (1999). The Rational Unified Process: an Introduction, Addison Wesley Inc. Reading, MA.

McCauley, R. (2001). Agile Development Methods, Poised to Upset Status Quo, SIGCSE Bullentin 33(4):14-15.

Mintzberg, H. (1983), Structure in Fives: Designing Effective Organizations, Prentice-Hall, New Jersey.

Nandhakumar, J. and Avison J. (1999). The Fiction of Methodological Development - a Field Study of Information Systems Development, Information Technology & People 12(2): 175-191.

Object Management Group. (2008). UML Resource Page. http://www.uml.org/. Site accessed Jan 26, 2008.

Oxford English Dictionary. (2008). http://www.askoxford.com/. Site accessed Jan 26, 2008.

Popkin Software. (2008). Systems Architect. http://www.telelogic.com/campaigns/popkin/index.cfm. Site accessed Jan 26, 2008.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991). Object-Oriented Modeling and Design, Prentice Hall, Englewood Cliffs, New Jersey.

Schwaber, K. (1995). Scrum Development Process, OOPSLA 1995 Workshop on Business Object Design and Implementation, Springer-Verlag.

Stolterman, E. (1991) The Hidden Rationale of Design Work (In Swedish: Designarbetets dolda rationalitet), PhD Thesis. Dept. of Information Processing, Umeå University, Sweden.

Suchman, Lucy A. (1987). Plans and situated actions: The problem of human-machine communication, Cambridge University Press, New York.

Takeuchi, H. and Nonaka, I. (1986). The New product Development Game, Harward Business Review, Jan/Feb:137-146.

Truex, D. P., Baskerville, R. and Travis J (2000), Amethodical Systems Development: the deferred meaning of systems development methods, Accounting, Management and Information Technologies, Volume 10(1):53-79.

Weber, M. (1978). Economy and Society, University of California Press, Berkeley.

Wiegers, K. E. (1998). Read My Lips: No New Models, IEEE Software 15(5):10-13.

Yourdon, E. (1989). Modern Structured Analysis. Prentice Hall, Englewood Cliffs, New Jersey.