An Investigation of the Target Environment for Agile Methods

Diane Strode[1] and Alexei Tretiakov[2]

[1)] Faculty of Business and Information Technology
Whitireia Community Polytechnic
Porirua, New Zealand
d.strode@whitireia.ac.nz

[2)] Department of Information Systems
Massey University
Palmerston North, New Zealand
a.tretiakov@massey.ac.nz

Abstract: The agile methods were investigated to determine the factors making up their target environment. A theoretical model of the target environment for agile methods was developed and then tested using a multi-case study. Data from nine software development projects, both agile and non-agile, was gathered from project leaders. Then a cross-case analysis of each factor in the theoretical model explored the relationship between each of the environmental factors and the extent of agile method usage. This led to a refined model of the target environment. The empirical data showed that specific environmental factors correlate with effective use of an agile method. We also report that methods are tailored for use, and that, although agile methods are distinct from ad hoc development, the extent of agile method use must be measured before conclusions from agile method studies can be considered valid.

1 Introduction

The agile methods are systems and software development methodologies that have found widespread adoption since 2000. They aim to expedite software development using iterative and incremental development, by supporting development in multi-disciplinary teams, and by reducing project risk [Ag01], [St05]. There is a need for research comparing the agile methods-as-published with their actual use in order to understand how they are used in practice [ELS05] and the conditions where they are more likely to be successful [TFR05]. Empirical research into how agile methods are used in practice is growing [ASS05], [HS05], [HTM05], [LWC04], [SR04], and research into how an agile method is explicitly tailored to fit the organisation or project is now available [ASS05], [HS05] along with research describing how XP-like practices can evolve in an organisation from a traditional base [HTM05]. Our research takes a different perspective and investigates how agile methods are tailored for use in order to help define the target

environment for agile methods. The literature on agile methods states that the agile methods are suitable in specific project environments [Be00], [BT04], [Co02], [Hi00], [St97] and this study provides empirical evidence to support this idea.

We investigated the factors affecting the use of agile methods using a mixed qualitative and quantitative research methodology suitable for investigating systems development methodologies *in-situ*. First a theoretical model of the environmental factors reported to affect the use of an agile method was developed and then tested using cross-case analysis of data gathered from systems development projects [Yi03]. We developed the theoretical model from the existing literature [BT04], [Co02], [Hi03], [NMM05], [Re02], [RME03], [RJ00], [TFR02], [We02] while putting emphasis on the primary publication describing the method. For each method, we took the first comprehensive description given in a book as the primary publication. We analysed the five earliest published agile methods using an analytical comparative framework adapted from that of Avison and Fitzgerald [AF03]. We selected these published methods because according to the literature they are widely adopted [Ag01], [Ch01], [Pr05]. Specifically, we studied Dynamic Systems Development Method (DSDM) [St97], Extreme Programming (XP) [Be00], Scrum [Sc95], Adaptive Software Development (ASD) [Hi02], and Crystal Methods (Crystal) [Co02], which we decomposed into their component parts including the philosophy (paradigm, assumptions and values, perspective, objectives, domain, target), modelling methods, techniques (such as test-driven development or pair programming), tools, scope, outputs, practice (background, roles and responsibilities, difficulties, skill levels), and the extent to which the method can be adapted to a situation (tailorability). The full analysis is provided in Strode [St05]. Based on this analysis we developed a comprehensive theoretical model of the target environment for agile methods (see Table 4). This model is a proposed set of organisation, technology, domain, project, and people factors that relate to the successful use of an agile method.

In the following sections, we describe the research method used. Then we present our results on the extent of method tailoring, followed by the result of our assessment of the target environment model and finally conclusions are drawn.

2 Method

The unit of analysis was a single software development project. Nine projects from different organisations were selected. Five projects used an agile method (agile projects) and four projects used a non-agile method (non-agile projects) which was either the Rational Unified Process or ad hoc development (no formal or well-defined development method). Of the agile projects, two relied on method combinations: one used XP combined with RUP and another used Scrum combined with XP.

Project leaders (the respondents) completed a questionnaire and a semi-structured interview and further organisational data came from documentation available on the respective organization's web site. We gathered data on each of the criteria identified in the theoretical model (see Table 4). In addition, we gathered organisational culture data, data on the development method used, the development techniques used, and on the

extent to which each technique was used. The questionnaire contained a list of all identified techniques organised into groups of similar techniques (see Table 3) to make it easy to read and to encourage the respondents to consider carefully their responses. We identified the techniques included in each method from our analysis of the five agile methods.

The projects were from various types (government and privately owned) and sizes of organisation. We selected projects according to a convenience sampling strategy and to ensure the accuracy and currency of the data assembled, we required that each project meet the following criteria:

1. The project had a well-defined purpose.
2. The project aimed to produce a software application (rather than limited to requirements collection etc.).
3. The project had both a beginning and an end date.
4. The project was either completed, or underway with at least 1/3 of the estimated project duration completed.

In addition we ensured that the respondents were experienced in systems development, could give a perceptive and accurate overview and history of the project, and had in-depth knowledge of the development methodology used.

The research was based on three propositions:

1. The full agile method is not used in practice; the method is tailored-for-use.
2. An agile method is tailored less when it is used in its target environment, compared with when it is used in an environment that significantly differs from its target environment.
3. There are specific environmental conditions, constituting a target environment, that are suitable for agile methods overall (we did not focus on differentiating the conditions particularly suitable for specific agile methods).

3 Agile Method Tailoring

Each of the agile methods is made up of a unique combination of development techniques and this allowed us to assess each of the agile projects to determine the extent of tailoring of the method. The following calculations were carried out:

a. The fraction of techniques used:

$$\% \text{ fraction of techniques used} = (T_c / T_{mc}) \times 100 \quad (1)$$

Where: T_c = count of techniques used on the project

T_{mc} = count of all techniques present in the method

Only those techniques that belong to the selected method were included (i.e. if the respondent selected techniques that do not belong to their nominated method then this technique was not included in the count).

b. The extent of agile method usage:

$$\% \text{ agile method usage} = (\sum T_q / (3T_{mc})) \times 100 \quad (2)$$

Where: T_q = extent of usage of a particular technique

The extent of usage of each technique was 0, 1, 2, or 3 (never used, seldom used, often used, always used the technique) as assessed by the project leader. Only those techniques that belong to the selected method were included (i.e. if the respondent selected techniques that do not belong to their nominated method then this technique was not included in the summation). Agile method usage and tailoring are related in that the higher the extent of agile method usage is for a particular method; the less the method is tailored.

Table 1 shows the extent of agile method usage for each agile project surveyed. Project Delta was assessed twice, once against XP and also against Scrum since a combination of techniques from the two methods was used on that project. These results show that each agile project tailored the method. For example the fraction of techniques used by Alpha, Theta and Zeta was 100% but the extent of usage was lower, 96%, 63% and 94% respectively. The other projects also tailored their method not only by reducing the fraction of techniques they chose to use but also the extent to which they used them. These results show that the method is tailored in each agile project and there is a range of tailoring reflected in the extent of agile method usage taking values from 96% to 56%. Proposition 1 from section 2: the method is tailored-for-use, was true in all projects using agile methods.

Project	Alpha	Beta	Delta	Delta	Theta	Zeta
Development method used	XP	XP	XP Scrum	XP Scrum	XP RUP	DSDM
Compared with	XP	XP	XP	Scrum	XP	DSDM
Total techniques in method	19	19	19	13	19	13
Number of techniques used on project	19	18	16	11	19	13
The fraction of techniques used (1)	100%	95%	84%	85%	100%	100%
Total quantity of techniques in method	57	57	57	39	57	48
Total quantity of techniques used on project	55	32	38	26	36	45
The extent of agile method usage (2)	96%	56%	67%	67%	63%	94%

Table 1: The extent of method tailoring on agile projects

To compare all projects both agile and non-agile, the extent of agile method usage was calculated for the non-agile projects based on how many techniques they selected from

the XP method. To determine the most suitable representative agile method we assessed the extent of agile method usage for each method. This provided five sets of data. Ranking the usage data showed a consistent pattern in the non-agile projects for all methods except for Crystal (see Table 2). We chose XP out of the four methods displaying a consistent pattern, as this method has the largest number of techniques, and thus was likely to result in the most accurate measure.

Project	Iota	Tau	Chi	Rho
Method used	RUP	Ad hoc	Ad hoc	Ad hoc
XP	40% (2)	54% (1)	30% (3)	18% (4)
DSDM	38% (2)	54% (1)	29% (3)	19% (4)
Scrum	36% (2)	38% (1)	21% (3)	3% (4)
Crystal	50% (1)	43% (2)	23% (3)	0% (4)
ASD	42% (2)	50% (1)	31% (3)	0% (4)
The bracketed number is the project rank from lowest usage (4) to highest (1)				

Table 2: The extent of agile method usage on non-agile projects

The pattern of usage of agile techniques by non-agile projects was as expected; the non-agile projects used less agile techniques than the agile projects (compare Table 1 and 2). This provides evidence that agile methods are distinct from ad hoc development and RUP development. However there is overlap in technique usage. This can be observed for example, in the use of XP. Beta, who nominated XP as their method (see Table 1) used XP at the 56% level and Tau, who used no development method, used XP techniques at the 54% level (see Table 2).

There are five techniques used on the projects that we found are not unique to agile development as they are also used in non-agile projects (see Table 3). They are customer-on-site, 40 hour week, requirements are prioritised, and design and coded solution are kept as simple as possible. These are simple actions and not techniques requiring training (such as test-first development or user stories), so this is not an unexpected result.

These results show the difference between agile and non-agile projects with respect to the techniques used. However, we found that an ad hoc or RUP project will use some of the techniques belonging to the agile methods because of the ubiquitous nature of some of the techniques. In addition, when an organisation nominates a method, they may be using it fully or partially. This indicates that when carrying out studies of agile methods researchers must carry out some quantitative assessment of the extent of agile method usage to verify that the organisation is using an agile method rather than ad hoc development, before drawing any conclusions about possible effects of agile method use.

	Techniques from the five agile methods as presented in the questionnaire	Used on all non-agile projects	Used on all agile projects
1	Concurrent development		
2	Iterative development		+
3	Time boxing (iterations of set length)		+
4	Incremental development		+
5	Evolutionary prototyping		
6	Small releases of software product		+
7	Component development		+
8	Test first development		+
9	Daily builds of complete system		+
10	Automated regression testing		+
11	Refactoring of code		+
12	Testing throughout each iteration		+
13	Software inspections		+
14	Customer on-site	+	+
15	Method coach on site		+
16	Tester(s) collocated with team		+
17	Customer focus groups		
18	Rooms organised for pair programming		
19	Whole team works in same office/floor		+
20	Dedicated meeting space		
21	Pair programming		+
22	Coding to an agreed standard		+
23	Collective ownership of code		+
24	40 hour week	+	+
25	Sprint Goal		
26	Daily team meetings		
27	Iteration planning meeting		+
28	Planning game		+
29	Reflective workshops for adaptation		
30	User stories		
31	System metaphor developed		
32	Only what has direct business value		+
33	Requirements are prioritised	+	+
34	Changes to requirements are negotiated		+
35	Joint Application Development (JAD)		
36	Design is kept as simple as possible	+	+
37	Coded solution is kept as simple as possible	+	+
38	Risk assessment at each iteration		+
39	Product Backlog		
40	Sprint Backlog		
41	Release Backlog		
42	Milestones to track progress		+
43	Product Backlog Graph metric		
44	Sprint Backlog Graph metric		

	Techniques from the five agile methods as presented in the questionnaire	Used on all non-agile projects	Used on all agile projects
45	Function point counts		
46	Project post mortem		
47	Feasibility study		
48	Business study		
49	Resource requirements analysis		
50	MOSCOW rules		
52	Unique methodology at start		
53	Tailored existing methodology at start		

Table 3: The techniques used on projects

4 The Environmental Conditions for Agile Methods

To address proposition 3, that there are specific environmental conditions that are suitable for agile methods overall, each criterion of the target environment model was analysed individually. Spearman's correlation coefficient (for ordinal data in the questionnaire, such as "involvement in making decisions", ranked from 0 to 5) or Pearson's product moment correlation coefficient (for ratio data in the questionnaire, such as "number of full-time developers", expressed by a number) was calculated to investigate the correlation between the usage of agile method techniques (as calculated for proposition 1) and the environmental criterion in the target model. Where statistical tests were used the hypothesis tested was always of the form:

H_o there is no relationship between the project environment factor (size of team, style of communication etc.) and the extent to which the agile method is tailored for use.
H_1 there is a relationship between the project environment factor and the extent to which the agile method is tailored for use.

The questionnaire contained a section on organisational culture that included additional questions about factors not present in the target environment model. These questions were included because they come from a set of related questions which provide a comprehensive assessment of organisational culture [Pa03]. We found three factors that were not present in the theoretical model, which show statistically significant correlation (Spearman's rho, 2 tailed test, significant at 0.05 or 0.01 level) with the extent of agility on the projects:

Item A: The organisation is very results oriented. A major concern is with getting the job done. People are very competitive and achievement oriented.
Item B: The leadership in this organisation is entrepreneurial, innovative, and risk taking.
Item C: The glue that holds the organisation together is loyalty and mutual trust. Commitment to this organisation runs high.

These three factors are added to the target environment model as follows: Item A became: *The organisation is results oriented*. The wording of item B is used as stated and item C became: *the organisation is based on loyalty and mutual trust and commitment*. The results of the analysis of data gathered to support the target environment model for agile methods is shown in Table 4. Some criteria were assessed using more than one question in the questionnaire. Results were reported as 'supported' when all questions related to the criterion gave a statistically significant correlation (Spearman's or Pearson's rho, 2 tailed test, significant at 0.05 or 0.01 level). The 'weakly supported' results indicate that one of two questions assessing the criterion did not show a statistically significant result or, in the case of criterion 20, a student t-test was used and was significant at the 90% level, but not at the 95% level. For criterion 3 no projects reported a 'mainly informal' communication style indicating that the agile projects have a balanced or informal style, but this was not unique to the agile projects. From these results a refined model was developed (see Table 5) which amalgamated certain criteria (when the criteria were determined to be very similar), excluded others (that were found to be agile method techniques rather than environment factors), and added items A, B and C, the new organisational culture factors.

Factor	Criteria	Target environmental model for agile methods	Result 1	Result 2
Organ-isation	1	The organisation values feedback and learning.	++	A
	2	The organisation values teamwork.	+	A
	3	The organisation values face-to-face communication.	+	N
	4	The organisation enables empowerment of people.	++	A
	5	The organisation is flexible and participative and encourages social interaction [Amalgamate with 2].	++	A
	6	Social interaction in the organisation is trustful, collaborative, and competent [Amalgamate with 1].	++	A
	7	Communication in the organisation is informal [Amalgamate with 3].	+	N
	8	The management style is that of leadership and collaboration.	++	A
	9	The size of the organisation is large/ small.	-	
Tech-nology	10	Any of: Internet application domains, shrink wrapped software, web-based systems, component delivery, component development, component assembly, client/server systems, networked systems, web-deployed applications.		
	11	Automated testing is used. [EXCLUDE]		
	12	Object-oriented technology.		

Dom-ain	13	Any of: application frameworks for external use, e-commerce and e-business, data warehouse, and products for the Internet software market.		
	14	The domain is interface intensive systems		
	15	The domain is business problems, business systems and applications	++	N
	16*	The domain is non-critical projects/ Critical new business initiative.		
Project	17	Projects that are complex.		
	18	Projects with vague requirements		
	19*	Projects with constant changes in requirements/ Projects with stable requirements.		
	20	Projects undergoing constant change.	+	A
	21	Projects with intense time pressure.	-	
	22	Projects with teams of 2 to 10 developers. [EXCLUDE]		
	23	Projects where documentation is minimised.	++	N
	24*	Projects involving new development/ Precedented systems/ Projects that are not constrained by an existing computing environment.	-	
	25	Projects involving any of fix-price contract software development, in-house development, outsourced software	++	N
	26	Projects using incremental development. [EXCLUDE]		
	27	Projects using iterative development. [EXCLUDE]		
	28	Projects where the project manager acts as a facilitator. [Amalgamate with 5]		
	29	Projects teams are collocated. [EXCLUDE]		
People	30	Users are actively involved in the project		
	31	Developers are experienced		

Key

++	Supported
+	Weakly supported
×	Evidence to show this is not true
-	Evidence is inconclusive
A	More common in agile methods projects
N	Not unique to agile methods projects

[EXCLUDE] this criterion is excluded in the refined model because it is a technique that forms part of one or more specific agile methods

* these items were found to contradict one another in the literature

Table 4: Model of the target environment for agile methods and results

Criteria from table 7		Result 1	Result 2
1	The organisation values feedback and learning. Social interaction in the organisation is trustful, collaborative, and competent.	++	A
2	The organisation values teamwork is flexible and participative and encourages social interaction. The project manager acts as a facilitator	+	A
4	The organisation enables empowerment of people.	++	A
8	The management style is that of leadership and collaboration.	++	A
Item A [new]	The organisation is results oriented.	++	A
Item B [new]	The leadership in the organisation is entrepreneurial, innovative, and risk taking.	++	A
Item C [new]	The organisation is based on loyalty and mutual trust and commitment.	++	A
20	Projects undergoing constant change.	+	A
3	The organisation values face-to-face communication and communication in the organisation is informal.	++	N
15	The domain is business problems, business systems and applications	++	N
23	Documentation is minimised.	++	N
25	Projects involve any of fix-price contract software development, in-house development, outsourced software	++	N
Key ++ Supported + Weakly supported A More common in agile method projects N Not unique to agile method projects			

Table 5: Refined model of the target environment for agile methods

5 Conclusion

This study has confirmed the three propositions listed at the end of section 2. In all cases the agile methods were tailored for use, as Fitzgerald found when studying object-oriented and structured development methodologies [Fi00]. We also found that the agile methods are tailored almost to the extent of ad hoc development in some cases. This means that studies of agile methods must quantify agile method use to be sure that conclusions drawn about the use and effects of agile methods are for the affect of agile method use rather than ad hoc development.

Proposition 2 and 3 are supported because we found a correlation between the extent of usage of an agile method and certain environmental factors. From this, we conclude that successful use of an agile method is likely to require the following environmental conditions to be present. The organisation, its management and the development teams

must value feedback and learning; social interaction must be trustful, collaborative, and competent. The organisation must be results oriented and based on loyalty, mutual trust, and commitment. Teamwork must be valued, and a flexible, participative, and encouraging approach to social interaction is necessary. The management style must be that of leadership and collaboration, and the project manager must act in a facilitating role with the staff empowered to make decisions. The leadership in the organisation must be entrepreneurial, innovative, and risk taking. Care must be used in drawing these conclusions as the effect of the environment on the method may not be exclusively in one direction. It is possible that adoption of an agile method has improved the social interaction in an organisation. Longitudinal studies of agile method adoption are needed to clarify this relationship. In addition we found evidence for 'projects undergoing constant change' as a factor in successful agile method use.

All projects reported using informal, face-to-face communication, and that the domain was business problems, business systems, and business applications, and all projects involved either fixed price contract software development, in house development or outsourced software. We found these factors are not unique to projects using agile methods. 'Documentation is minimised' was also a factor common to all agile projects but this was not assessed on non-agile projects. These factors may be necessary for successful use of an agile method but they are not unique to agile projects.

This study drew conclusions based on a small number of projects and additional studies of agile and non-agile projects are needed to confirm these results. A study of the effect of combinations of environmental factors or possible interaction between factors would also provide further understanding of the target environment where agile methods are more likely to be successful.

References

[Ag01] AgileAlliance: Manifesto for agile software development. 2001. Retrieved 17 February, 2003, from http://www.agilemanifesto.org.

[AF03] Avison, D. E.; Fitzgerald, G.: Chapter 27: Methodology comparisons. In Information systems development: Methodologies, tools and techniques (3rd ed.). McGraw-Hill Publishing Company, London, 2003.

[ASS05] Aydin, M. N.; van Slooten, K.; Stegwee, R. A.: On the adaptation of an agile information systems development method. Journal of Database Management, 2005, 16(4); pp. 24-40.

[Be00] Beck, K.: Extreme programming explained: Embrace change. Addison-Wesley, Boston, 2000.

[BT04] Boehm, B.; Turner, R.: Balancing agility and discipline. Addison-Wesley, Boston, 2004.

[Ch01] Charette, R.: The decision is in: Agile versus heavy methodologies. e-Project Management Advisory Service, Cutter Consortium, 2, 2001.

[Co02] Cockburn, A.: Agile software development. Addison-Wesley, Boston, 2002.

[DP03] Dube, L.; Pare, G.: Rigor in information systems postivist case research: Current practice, trends, and recommendations. MIS Quarterly, 2003, 27(4); pp. 597-635.

[ELS05] Erickson, J.; Lyytinen, K.; Siau, K.: Agile modeling, agile software development, and extreme programming: The state of research. Journal of Database Management, 2005, 16(4); pp. 88-100.

[Fi00] Fitzgerald, B.: Systems development methodologies: The problem of tenses. Information Technology and People, 2000, 13(3); pp. 174-185.

[HS05] Henderson-Sellers, B.; Serour, M. K.: Creating a dual-agility method: The value of method engineering. Journal of Database Management, 2005, 16(4); pp. 1-23.

[Hi02] Highsmith, J.: Agile software development ecosystems. Addison-Wesley, Boston, 2002.

[Hi03] Highsmith, J.: Agile software development - why it is hot! In Marchesi, M; Succi, G; Wells D; Williams, L. (Eds.), Extreme programming perspectives. Addison-Wesley, Boston, 2003; pp. 9-16.

[Hi00] Highsmith, J. A.: Adaptive software development: A collaborative approach to managing complex systems. Dorset House Publishing, New York, NY, 2000.

[HTM05] Hilkka, M.-R.; Tuure, T.; Matti, R.: Is extreme programming just old wine in new bottles: A comparison of two cases. Journal of Database Management, 2005, 16(4), pp. 41-61.

[LWC04] Layman, L.; Williams, L.; Cunningham, L.: Exploring extreme programming in context: An industrial case study. Paper presented at Agile Development Conference. 2004; pp. 32-41.

[NMM05] Nerur, S.; Mahapatra, R.; Mangalaraj, G.: Challenges of migrating to agile methodologies. Communications of the ACM, 2005, 48(5), pp. 73-78.

[Pa03] Paperone, C. R.: Applying the competing values framework to study organizational subcultures and system-wide planning efforts in a military university. Unpublished Doctor of Philosophy, The Pennsylvania State University. 2003.

[Pr05] Pressman, R. S.: Software engineering: A practitioner's approach (6 ed.). McGraw-Hill Higher Education, Boston, 2005.

[Re02] Reifer, D. J.: How to get the most out of extreme programming/agile methods. Paper presented at Extreme Programming and Agile Methods - XP/Agile Universe, Chicago, IL, USA. 2002.

[RME03] Reifer, D. J.; Maurer, F.; Erdogmus, H.: Scaling agile methods. IEEE Software, 2003 July/August; pp. 12-14.

[RJ00] Rising, L.; Janoff, N. S.: The Scrum software development process for small teams. IEEE Software, 2000, 17(4); pp. 26-32.

[Sc95] Schwaber, K.: Scrum software development process. 1995. Retrieved 3 October, 2004, from http://www.controlchaos.com/old-site/scrumwp.htm

[SR04] Sharp, H.; Robinson, H.: An ethnographic study of XP practice. Empirical Software Engineering. 2004, 9(4); pp. 353-375.

[St97] Stapleton, J.: DSDM dynamic systems development method. Addison-Wesley, Harlow, England, 1997.

[St05] Strode, D. E.: The agile methods: An analytical comparison of five agile methods and an investigation of their target environment. Unpublished Master of Information Sciences (Information Systems), Massey University, Palmerston North. 2005.

[TFR02] Turk, D.; France, R.; Rumpe, B.: Limitations of agile software processes. Paper presented at the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, XP'2002. Aghero, Italy. 2002.

[TFR05] Turk, D; France, R.; Rumpe, B.: Assumptions underlying agile software-development processes. Journal of Database Management, 2005, 16(4); pp. 62-86.

[We02] Wendorff, P.: Organisational culture in agile software development. Paper presented at the Product Focused Software Process Improvement: 14th International Conference, PROFES 2002, Rovaniemi, Finland. 2002.

[Yi03] Yin, R. K.: Case study research (3rd ed.): Sage Publications, Thousand Oaks, 2003.