Examining Perceptions of Agility in Software Development Practice

Sergio de Cesare[1], Mark Lycett[1], Robert D. Macredie[1], Chaitali Patel[2], and Ray Paul[1]

[1]School of Information Systems, Computing and Mathematics,
Brunel University, Uxbridge, Middlesex, UB8 3PH
United Kingdom

[2] Agilisys Ltd.,
26-28 Hammersmith Grove,
London, W6 7AW
United Kingdom.

Abstract

Agility is a facet of software development attracting increasing interest. The purpose of this paper is to investigate the value of agility in practice and its effects on traditional plan-based approaches. Data collected from senior development/project managers in 62 organizations is used to investigate perceptions related to agile development. Specifically, the perceptions tested relate to (a) belief in agile values and principles, (b) value of agile principles within current development/organization practice. These perceptions are then examined in the context of current practice in order to test perceptions against behavior and understand the valued aspects of agile practice that are implicit in development today. The broad outcome indicates an interesting marriage between agile and plan-based approaches. This marriage seeks to allow flexibility of method while retaining control.

Introduction

Organizations undertaking software development are often reminded that successful practice depends on a number of non-technical issues that are managerial, cultural and organizational in nature [4, 8]. These issues cover aspects from appropriate corporate structure, through software process development and standardization to effective collaborative practice. Since the articulation of the 'software crisis' in the late-1960s, significant effort has been put into addressing problems related to the cost, time and quality of software development via the application of systematic processes and management practices for software engineering. Early efforts resulted in prescriptive structured methods, which have evolved and expanded over time to embrace consortia/company-led initiatives such as the Unified Modeling Language and the Unified Process alongside formal process improvement frameworks such as the International Standards Organization's 9000 series, the Capability Maturity Model and SPICE.

More recently, the philosophy behind traditional plan-based initiatives has been questioned by the agile movement, which seeks to emphasize the human and craft aspects of software development over and above the engineering aspects [1, 2]. Agile practice is strongly collaborative in its outlook, favoring individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan (see Sidebar 1). Early experience reports on the use of agile practice suggest some success in dealing with the problems of the software crisis [12], and suggest that plan-based and agile practice are not mutually exclusive [10]. Indeed, flexibility may arise from this unlikely marriage in an aim to strike a balance between the rigor of traditional plan-based approaches and the need for adaptation of those to suit particular development situations. With this in mind, this article surveys the current practice in software engineering alongside perceptions of senior development managers in relation to agile practice in order to understand the principles of agility that may be practiced implicitly and their effects on plan-based approach.

Details of the Survey

To elicit an understanding of current practice and perceptions related to agility, an online questionnaire was designed [11] with the following objectives in mind:

- To review current software development practice.
- To assess perceptions related to agile principles and values.
- To examine the implicit/explicit integration of agile principles with mainstream development/organizational practice.

E-mail was used to contact senior development management at 970 organizations in the United Kingdom, requesting that they participate in the research. The selection of participants was based on the contact list available and research noting that senior management influence productivity [3]. The survey itself comprised of 32 questions that were organized into three sections addressing (a) background/demographics, (b) current software development practice and (c) perceptions related to agility. Online presentation of the questionnaire ensured that the section on current practice had to be completed before the section on perceptions of agility could be viewed and completed in order to reduce bias.

INSERT TABLE 1

Table 1 summarizes the demographics of the 62 organizations that responded to the online survey (equating to a response rate of 6%). Overall, the respondents had on average more than 10 years development experience, had variation in their development experience (having worked in more than 4 software development roles), and worked in organizations that had more than 500 employees. The majority of responses received were from IT Directors, Senior Architects and IT Managers. In addition, responses from the larger organizations covered in-house development departments of a variety of sizes from fewer than 10 employees to more than 500 employees. Last, and importantly, only about a quarter of the respondents used an explicitly recognized agile method as detailed below.

General View of Software Development

The objective of understanding the state-of-practice was to provide a benchmark in order to assess the gap with the concepts of agile practice. The use of software development processes provided the framework for enquiry and a number of questions were asked to provide correlation points for agile practice (discussed later). The major outcomes of interest are that:

- Most organizations employ a software development process but implement it in a flexible/configured manner.
- The use of framework-based processes (i.e., RUP like approaches that allow contextualization of processes) appear to be growing in popularity.
- Development processes are supported by the strong use of iterative lifecycles, but change is implemented in a manner that does not necessarily fit with such iterative cycles.

In detailed terms, more than half the sample claim to use a formal software development process within their organization, either developed in-house (56%) or commercially branded (18%). Of the respondents that use formal software development processes (74%), the majority indicated that they based their processes on framework-based approaches which cater for flexibility. These approaches included the Rational Unified Process (RUP) 30%, Dynamic Systems Development Methodology (DSDM) 18% and eXtreme Programming (XP) 8%. Interestingly, organizations using a formal software development process (a) tailor/configure them strongly (82%), lending weight to the adage that 'people don't adopt a methodology, they adapt it' (attributed to Tom DeMarco) and (b) use different processes for different types of project (79%). Tailoring of software development processes once seen as a negative activity that violates the 'rigor and consistency' of software development processes (see [6, 9]) now appears to be seen in a positive light as necessary and is embraced. The following opinion highlights a practitioner's view on flexibility: "Although a method provides a useful framework, it has to be flexible in order to reflect real-world constraints and priorities".

The levels of effort put into tailoring/configuring process varied: 16% stated they spent 1% percent of overall project time on tailoring the software development process; 67% of respondents stated the time to tailor processes ranged from 1% to 5% of overall project time; 19% stated they spent 10% – 20% of the project time and 2% stated that they use 40% of the project time. The type of people involved in process tailoring were senior management, project managers, architects and the development team – where there appeared to be a decision maker, the development team was always involved in the decision. Lastly, 'project characteristics' were the predominant criteria on which tailoring decisions were made (44%).

Of the respondents that do follow some form of software development process, the majority stated that their processes are based on an iterative and/or incremental software development lifecycle (68%), the remainder being based on the more sequential waterfall model. Nearly all respondents stated that they cater for changing requirements, even late in the project (98%). Of that proportion, only some follow the dynamic prioritization path proposed by iterative/incremental development (39%) - the majority of respondents continue to use traditional change control request (61%).

Perceptions of Agile Principles

The objectives of questioning around agile principles were (a) to ascertain perceptions related to the value and importance of agility and (b) to relate that information back to the context of current practice in order to understand the match between principle and practice – do organizations practice what they state as important. In summary, respondents were presented with background information related to agile values and principles and asked to rank the principles in terms of importance from an organizational perspective and then rate its importance from a personal perspective. Summary results are shown in Table 2; these findings were then linked back to earlier questions to match perceptions with practice. The major points of interest are that:

- The most widely valued agile principles relate to the frequent delivery of working software, daily interaction between business people and developers and the importance of face-to-face communication.

- The highest levels of equivocation exist around principles related to changing requirements late in the development cycle and working software being the primary measure of progress.
- The lowest areas of organizational importance were that the needs of the development team must be prioritized in line with cost, quality and time and that architecture and design should emerge and not be imposed.

In detailed terms, the results showed Principles 1, 5 and 7 (see Table 2) as unequivocal in their perceived importance (60, 84 and 60% respectively). When examining the frequent delivery of working software the results from current practice showed that the majority of respondents delivered on three monthly cycles (31%). Significantly, however, a number of respondents delivered on monthly cycles (24%) and two-week cycles (24%) – smaller percentages delivered on weekly cycles (5%) or produced only a final release (17%). In communication terms, the frequency of communication between Business and IT ranged from approximately once a month (13%), through once or twice a week (56%), to communication on a daily basis (29%). The primary means of communication within software development practice is face-to-face within development team (79%) - few respondents use e-mail (17%) or telephone (4%) as a primary communication tool for example.

INSERT TABLE 2

While the delivery of working software was seen as important, the results related to software being the primary measure of progress were much more ambiguous. Here, findings from the data collected in practice showed a strong reliance on traditional project metrics. While a number of project respondents do use working software as the primary measure of progress (26%), a significant number use other project milestones and deliverables (54%) or completion of process/lifecycle stages (20%). Similarly, while the majority of respondents state that they cater for change late in the development cycle, change does not appear to be a particularly welcome factor. Interestingly, however, there is a strong correlation between the respondents who embrace change (40%) and their use of dynamic change control techniques (38%).

Low areas of organizational importance indicate that a significant number of respondents (47%) value human facets of development over and above typical facets of cost, quality and time. This view is borne out to a given degree when the data is correlated with the view that projects should be built around motivated individuals (53%) and practice indicating that (a) the development team is involved with decision making (64% on average) and (b) individuals are empowered to make decisions on work directly related to them (89%). In contrast, however, the results note that, while the development team is involved in tailoring the software development process (64%), team skills are the least used tailoring criteria (22%).

The results also indicated that architecture and design were viewed from a top-down perspective and not as artifacts that should emerge from practice and dialogue. While this is interesting, the respondents' comments indicate a poor understanding of the question and we have thus discounted data in this regard. Lastly, other interesting but less obvious results are:

- Project/process reflection has some organizational 'presence'. Here practice results show that the primary techniques are 'lessons learnt' analysis (50%), self-reflection (24%) and as an outcome of formal monitoring and measuring (13%). Those respondents who stated that no explicit reflection was carried out cited time limitation as the cause.
- Documentation is still seen as important. Here, however, the majority of respondents viewed their current levels of documentation as necessary and sufficient for their needs (50%) though a significant number saw their documentation as necessary but not sufficient for their needs (38%). In correlation terms, the majority of respondents who note documentation as both necessary and sufficient use formal but tailored development processes.

Implications

All surveys carry health warnings and this example is not immune. Overall, the sample size of the data can be considered limited in that it addresses a UK audience and the response rate is small at 6%. With such limitations in mind we have sought to present the results without bias in order to allow the reader to

draw their own implications. Overall, however, a number of interesting conclusions may be drawn in relation to the state of practice and the implicit use of agile principles:

- A number of respondents state that they develop software in an ad-hoc manner (26%). From one perspective, this finding may be considered worrying in light of the ongoing issues of the software crisis. From another perspective, however, ad hoc does not necessarily equate to disorganized and may be related to the use of 'rules of thumb' used implicitly across development [see 7], where formal process is considered too heavy. Indeed, a significant number of respondents in this respect work in small development environments (63% have fewer than 9 employees) even within large organizations.
- The trend of adapting/tailoring software development processes to context is on the increase. In relation to an earlier study [see 5], the findings suggest an increase in the number of organizations that tailor/adapt their development process and a decrease in the number of organizations that follow a commercial process in a prescriptive fashion.
- A large number of the organizations surveyed now base their software development on iterative and/or incremental lifecycles (68%). The frequency of software delivery ranges from every two weeks to every three months (the majority) and by far outweighs delivery only on final release. Change control, however, is still firmly locked in a traditional project management loop.

Table 3 provides a summary of agile themes that are relevant in terms of their adoption in practice. The findings suggest that there is an uptake of the principles of agile development, though in larger organisations this uptake is countered by the perceived need to maintain rigor and control of the process. In practice, the adoption of agile principles is implicitly tied with process tailoring, configuration and iterative lifecycles - the correlation between these factors and the agile principles noted as important is strong. Process flexibility therefore emerges as the 'marriage' between agility and plan-based approaches, increasingly supported by framework-based process approaches such as RUP.

INSERT TABLE 3

References:

1. Beck, K., et al., *Manifesto for Agile Software Development*. 2001.

2. Cockburn, A., *Agile Software Development*. The Agile Software Development Series, ed. A. Cockburn and J. Highsmith. 2002: Addison Wesley.

3. Dutta, S., M. Lee, and L.V. Wassenhove, *Software Engineering in Europe: A Study of Best Practices.* IEEE Software, 1999. **16**(3): p. 82 - 90.

4. Fafchamps, D., *Organisational Factors and Reuse.* IEEE Software, 1994. **11**(5): p. 31-41.

5. Fitzgerald, B., *The Use of Software Development Methodologies in Practice: A Field Study.* Information Systems Journal, 1997. **7**(3): p. 201-212.

6. Glass, R., *Through a Glass, Darkly. Methodologies: Bend to Fit?* The Software Practitioner, Data Base Advances, 1996. **27**(1): p. 14 - 16.

7. Glass, R.L., *In Seach of Meaning (A Tale of Two Words).* IEEE Software, 2002. **19**(4): p. 134 - 136.

8. Griss, M.L. and M. Wosser, *Making Reuse Work at Hewlett-Packard.* IEEE Software, 1995. **12**(1): p. 105-107.

9. Hardy, C.J., J.B. Thompson, and H.M. Edwards, *The use, limitations and customization of structured systems developement methods in the United Kingdom.* Information and Software Technology, 1995. **37**(3): p. 467-477.

10. Lycett, M., et al., *Resolving the Tensions of Agility in Standardized Practice.* IEEE Computer, 2003. **36**(6): pp.79-85 June 2003.

11. Patel, C., Lycett, M., Macredie, R.D., and de Cesare, S., Perceptions of Agility and Collaboration in Software Development Practice. In Proceedings of the 39th Annual Hawaii International Conference on System Sciences (Kauai, Hawaii, January 4-7). Los Alamitos, California: IEEE Computer Society Press, 2006.

12. Reifer, D.J., *How Good are Agile Methods?* IEEE Software, 2002. **19**(4): p. 16 - 18.