

Towards optimal software engineering: learning from agile practice

David Bustard · George Wilkie · Des Greer

Received: 29 April 2013 / Accepted: 30 May 2013 / Published online: 10 July 2013
© Springer-Verlag London 2013

Abstract In essence, optimal software engineering means creating the right product, through the right process, to the overall satisfaction of everyone involved. Adopting the agile approach to software development appears to have helped many companies make substantial progress towards that goal. The purpose of this paper is to clarify that contribution from comparative survey information gathered in 2010 and 2012. The surveys were undertaken in software development companies across Northern Ireland. The paper describes the design of the surveys and discusses optimality in relation to the results obtained. Both surveys aimed to achieve comprehensive coverage of a single region rather than rely on a voluntary sample. The main outcome from the work is a collection of insights into the nature and advantages of agile development, suggesting how further progress towards optimality might be achieved.

Keywords Software engineering · Agile software development · Software process improvement · Industry surveys

1 Introduction

Optimizing any human activity means finding a balance among the factors affecting the outcome, to achieve the best results overall. Typically, there are many such factors involved but these can often be distilled down to a ‘vital few’. For example, in competitive cycling, “the rider who generates the most power, for the longest duration, whilst weighing as little as possible, and slipping efficiently through the air, usually wins the race” [1]. In software engineering, the corresponding key factors for success are: (1) the quality of the end product; (2) the efficiency and effectiveness of the development process; and (3) the resulting satisfaction of both the software producers and those who use it—at both a technical and business level. In particular, the developers should be proud of what they have created, the work suitably rewarded, and the development process and product meet the needs and expectations of both the business customer and the end-users.

In the 1990s, achieving optimality was typically based on the use of process reference models, such as CMMI [2]. The general strategy was to compare an organization’s software development practices with the ideals described in these models, and then make improvements by addressing any differences found.

The agile paradigm, emerging in 2001 [3], introduced an alternative approach: one based on initial process tailoring and subsequent continuous improvement—achieved through ongoing review and follow-up response to deficiencies found. This is described in a Forrester report published in 2010 [4], as agile methodologies not being used ‘out of the box’ but instead, “teams are puzzling out the mix of methodologies and combining them to fit within their organizational realities.”

That approach suggests a reasonable path towards optimality, with organizations progressively experimenting with

D. Bustard (✉)
School of Computing and Information Engineering,
University of Ulster, Coleraine, UK
e-mail: dw.bustard@ulster.ac.uk

G. Wilkie
School of Computing and Mathematics,
University of Ulster, Jordanstown, UK
e-mail: fg.wilkie@ulster.ac.uk

D. Greer
School of Electronics, Electrical Engineering and Computer Science,
Queens University of Belfast, Belfast, UK
e-mail: des.greer@qub.ac.uk

agile techniques to identify the best combination for their own particular circumstances. Unfortunately, there is a difficulty here, in that there is currently no agreed reference model against which to make decisions, because both the principles and practices of software development are still evolving. In particular, a clear definition of ‘agile development’ has yet to emerge. For example, the definition from the Agile Alliance is effectively given in historic terms [5]: “In the late 1990s several methodologies began to get increasing public attention. Each had a different combination of old ideas, new ideas, and transmuted old ideas. But they all emphasized close collaboration between the programmer team and business experts; face-to-face communication (as more efficient than written documentation); frequent delivery of new deployable business value; tight, self-organizing teams; and ways to craft the code and the team such that the inevitable requirements churn was not a crisis.”

This is useful in identifying core attributes of agile practice but it makes no attempt to prioritize them, identify secondary attributes that may also be important, or suggest that the understanding of agile development is evolving.

More recently, Ambler, in the *Encyclopedia of Software Engineering* (Taylor & Francis) [6], offers the following definition: “agile software development is an evolutionary, highly collaborative, quality-focused approach to software development where potentially shippable working software is produced on a regular basis.” This is similar to the Agile Alliance definition, but suggests ‘regular’ rather than ‘frequent’ delivery of software, and raises the importance of product quality. Ambler then goes on to propose five criteria for a development team to be classified as ‘agile’ [6]:

- Produce working software on a regular basis
- Do continuous regression testing (and better yet, take a test-driven development approach)
- Work closely with their stakeholders, ideally on a daily basis
- Be self-organizing within an appropriate governance framework
- Regularly reflect on how they work together and then act to improve on their findings

This description expands on the initial definition but in doing so constraints the scope of acceptable agile practice to a level that seems overly prescriptive.

A third and final example can be found in the current Wikipedia description [7], which states that: “agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.” This summary stresses the evolutionary aspect of software development and highlights the importance of team organization

and structure but largely delegates the definition of agile development to ‘agile methods’. Currently, the most popular methods are Scrum [8–10], Kanban [11, 12], XP (eXtreme Programming) [13] and DSDM (Dynamic Systems Development Method) [14]. Such methods are well defined but users are encouraged to adapt them to the needs of specific circumstances [15] so again there is uncertainty as to what agile software development necessarily entails.

One approach to clarifying this uncertainty, and hence help in its resolution, is through the study of current practice in organizations. What are organizations doing, where have they found benefit, and what are the implications for software engineering in general? That is the focus of this paper. It describes successive studies in 2010 and 2012 in which the views of software development companies in Northern Ireland were surveyed. The main motivation for the work was (1) to gain insights into the essential nature of agile development to better understand its strengths and weaknesses; and (2) to identify the key elements of agile practice that might form the basis of a reference model for assessing and improving agility in organizations.

Section 2 of the paper describes the design and implementation of the study, covering its objectives, the resulting question-set produced and the survey process followed. Section 3 presents the results obtained in the context of agile development moving to a central role in software engineering, identifying the limitations of the study. Section 4 then examines the need for a greater clarification of agile concepts as a basis for making further progress towards optimal software engineering.

2 Survey design and implementation

The authors undertook surveys of Northern Ireland software development companies in 2010 and 2012. These differed from most agile surveys [16–19] in attempting to have comprehensive coverage of a region rather than rely on general volunteers.

The main purpose of the 2010 survey was to assess the level of agile adoption in the province. The intention in 2012 was to rerun the 2010 survey with the same set of companies, adding questions as necessary to help understand how their agile practice was maturing. The sub-sections that follow describe the process that was followed in each case and the design of the questionnaires involved.

2.1 2010 Survey

The first step in 2010 was to identify the companies that might participate. This was achieved by building the survey around the software development companies that had either engaged with Invest Northern Ireland (INI—the local

economic development agency) or who had joined Momentum, the trade association for the ICT industry in Northern Ireland. Fifty-six companies were identified in this way.

To give authority to the survey, the initial invitation to participate was issued by INI, using email directed at the CEO, CIO, CTO or senior representative in each organization. Participation was, however, optional. About half of the companies contacted agreed immediately to take part but others needed further encouragement, bringing the final total up to 37.

The next stage of the survey was to issue and oversee the completion of the questionnaire. Given that there were three researchers involved (the authors) and the companies to be surveyed had a relatively narrow geographical spread, it was a straightforward decision to gather information face-to-face, through structured interviews. The direct contact approach made the size and wording of the questionnaire less critical and helped achieve (1) a high response rate; (2) full completion of each questionnaire (recording additional notes, as appropriate); and (3) consistency in the interpretation of each question.

2.2 2012 Survey

In principle, refreshing the results of the 2010 agile survey in 2012 simply required the original 37 participants to update their 2010 responses, and answer a few additional questions if using agile development routinely. In practice, however, a relatively large number of organizational changes had occurred in the interim, with the effect that only two-thirds of the 2010 contacts replied to the invitation to take part in the new study. Some contacts had changed jobs, but major organizational changes had also occurred. For example, two companies were no longer in operation, one had transferred its development activity outside Northern Ireland, and six had been acquired by larger companies, including IBM, HP and Oracle.

With effort, contact was eventually established with 32 of the original 37 participants. New companies had also been identified, and as the goal was to have comprehensive regional coverage, as many as possible were recruited; an additional eight companies brought the final total up to forty.

As in 2010, the intention was to have face-to-face meetings with each company but in cases where there had been little change since the previous survey the original questionnaire was simply updated and returned. For most participants, however, the changes from 2010 were significant.

2.3 Questionnaire

The questionnaire had two main sections: one gathering summary information about each organization and the other capturing details of its experience with agile software

development. The questions on the organization covered how long the company had been in operation, its business area (including types of software produced), its autonomy (independent or part of a larger organization), its size (technical staff and overall numbers), and any conformance to external standards, such as ISO 9001 or CMMI.

The number of technical questions posed depended on the agile experience of the organization. Twenty questions were put to those companies that used agile development routinely. The first twelve established when agile use began, why and how it was introduced, the extent of its use, involvement in specific agile methods, selection of team members, team sizes and locations, and the number of projects completed so far.

More detailed questions then followed, focusing mainly on the product, process and people benefits involved. Specifically, the questions covered:

- *Identification of product benefits:* The impact of agile development on the software attributes cited in the ISO 9126 Quality Model [20] (functionality, reliability, usability, efficiency, maintainability, and portability), as an indication of perceived product benefits. Survey respondents replied on a scale of -2 (much worse) to $+2$ (much improved) against each of the ISO 9126 product characteristics.
- *Identification of process benefits:* An assessment of the advantage or disadvantage in using agile development across over 30 individual software engineering activities, including aspects of requirements analysis, system design, and project management, again on a scale from -2 (much worse) to $+2$ (much improved).
- *Identification of people benefits:* An assessment of the satisfaction of clients, developers and their managers, on the same scale of -2 (much worse) to $+2$ (much improved).
- *Alignment with agile principles:* The level of agreement or disagreement with the 12 Principles of Agile Development on a scale from -2 (strongly disagree) to $+2$ (strongly agree). This was to clarify each company's understanding and implementation of the agile approach.
- *Use of agile techniques:* The level of experience, if any, with the 12 practices of XP [11], as an indication of each company's use of specific agile techniques. Replies were on a scale from 1 (less than 1 year) to 5 (5 or more years).
- *Use of agile tools:* Details of any specific tools used in the support of agile development, to identify the most popular and assess their influence on practice.

To make a direct comparison between the 2010 and 2012 surveys, the 2010 questionnaire was used in 2012 but extended with questions related to maturity of practice. As the questionnaire was already lengthy, care had to be taken when adding new material to avoid deterring participants. It was decided, therefore, to have just one additional page

and motivate it with a single research question: “How close are organizations to achieving optimal benefit from an agile approach?” The aim was to clarify areas of practice where additional research and development was needed.

Detailed questions were then asked on this theme, inviting participants to rate their organization’s performance on eight aspects of agile development on a scale from ‘minimal’ to ‘optimal’, covering:

- The maturity of its engineering practice—was it well established and well understood?
- The effectiveness of its development process—was it efficient and delivering high quality products?
- Gains from specific agile techniques—were techniques known and used appropriately?
- Gains from available agile tools—were tools known and used effectively?
- Avoidance of possible side effects—were there any negative outcomes from agile use?
- Use in unfavorable circumstances—was the agile approach used even if the development/business environment did not lead naturally to an agile approach?
- Use across the organization—was the agile approach used outside software development as, in principle, it is applicable to any type of project [10]?
- Ensuring compliance with external standards, if appropriate—does the agile approach make it more or less difficult to demonstrate compliance?

These additional questions effectively changed the emphasis of the survey from examining adoption practice to reviewing the more advanced aspects of agile development used in those organizations that applied it routinely.

2.4 Information gathering

A substantial amount of information was gathered at each interview. For a company involved in agile development, completing the questionnaire took roughly an hour; for others, with little or no involvement, it was much shorter. To help ensure that there was sufficient reflection on the answers provided, it was decided to issue the questionnaire (electronically) in advance of an interview, with a request to draft responses as far as possible. At the interview, the answers would then be confirmed, additional notes taken, and any missing sections completed.

As a check on the suitability of the questionnaire, it was piloted with three companies with whom the authors had worked previously. No significant changes were necessary following this evaluation.

The organizations surveyed covered a wide range of business sectors, including insurance, retail, telecoms, financial, aerospace, scientific, defense, healthcare, oil and gas, edu-

cation, security, and government. The software development industry in Northern Ireland is composed, predominantly, of small-to-medium sized enterprises (SMEs), with around 70 % employing fewer than 50 staff. The distribution of company sizes in 2012 was similar to that in 2010. One significant difference, however, was that while there was an even split between multinational and indigenous companies in 2010, this had changed to a 60:40 ratio in favor of multinational companies in 2012, mostly as a result of acquisitions though some new companies had also been established. Acquisitions require some re-alignment of practices between the organizations involved. This is initially disruptive, but can accelerate adoption, as was noted in three cases. The presence of multinational companies and the fact that many of the others operated at a global level, suggested that any conclusions drawn from this local survey would have wider significance.

3 Agile approach in software engineering

When the first survey was carried out in 2010 it was unclear if agile development was replacing the traditional planned approach or simply offered an alternative that could be selected if circumstances were favorable. The survey results indicate, however, that companies are tending to make it their base technique, and using it even in circumstances where a traditional approach would be a reasonable choice. The tentative conclusion, therefore, is that the agile model is moving to a central role in software engineering. Below are three related areas of evidence from the surveys to support that view. These discuss: (1) the increased adoption of agile methods; (2) a reduction in the factors inhibiting agile use; and (3) increased benefits from the agile approach.

3.1 Increased adoption levels

The first reason for concluding that agile is replacing planned development as the standard approach to software engineering is that the level of agile adoption is high and rising. In both the 2010 and 2012 surveys, each company was asked to put its experience with agile software development into one of four categories, ranging from ‘no experience’ to ‘general approach taken across the organization’. Table 1 identifies the number of companies in each category for each survey.

The main conclusion from the 2010 survey was that the number of companies using an agile approach routinely for development was around 50 %. Table 1 shows a significant shift towards greater agile use in 2012, with the peak moving from Category C to Category D. Of the 32 companies participating in 2010, five moved from C to D, one from B to C, and one from A to B; because of acquisition, one also moved directly from A to D. Exceptionally, one company

Table 1 Levels of experience in agile use

Response	Number of companies	
	2010	2012
A. No direct experience so far	6	3
B. Some experimental use	7	9
C. Routine use for some projects	15	10
D. General approach taken across the organization	9	20

moved away from agile development (C to B) but that was only because of key staff loss in downsizing.

The five companies that were unable to take part in the 2012 survey (3 no longer in operation) covered all categories: 1 A, 2 Bs, 1 C and 1 D. The eight new companies had at least experimental use of agile development and their distribution reflected the overall 2012 profile: 2 Bs, 2 Cs, and 3 Ds.

3.2 Reduction in inhibiting factors

In 2010, several companies described agile as “just another tool in the box” and chose to use it when a project seemed appropriate. Typically, that meant that requirements were flexible and somewhat uncertain. That seemed sensible and indeed was taken as an indicator of agile maturity at that time. By 2012, the picture had changed, however, with some respondents now opting to take an agile approach to every project, making whatever adjustments were necessary when conditions were unfavorable. One company, for example, used an agile approach to a large fixed-price contract that was based on a detailed upfront requirements specification. Following the agile approach, they regularly delivered working software for evaluation, which everyone found beneficial. To deal with the inevitable evolution in requirements that occurred during the project, the specification was renegotiated when necessary, either to swap one requirement for another or charge for additional work. This process was initially difficult but improved as levels of trust increased.

Companies with contracts in the public sector are particularly affected in this way, with the additional burden of an imposed hierarchical project management structure [21]. This position is now easing, however, with an explicit endorsement of the agile approach cited in the 2011 UK Government ICT Strategy [22]. This includes the statement “Government will apply agile methods to ICT procurement and delivery to reduce the risk of project failure. Agile methods allow projects to respond to changing requirements and ensure that the solutions meet business requirements.” The implementation plan for this strategy set the ambitious target of using agile techniques in 50 % of major ICT-enabled government programs by April 2013 [23].

Another conclusion drawn from the 2010 survey was that very small companies did not really need any defined development methods and were largely agile anyway in that they worked together closely and had a strong product focus. Evidence from the 2012 survey, however, suggests that there are gains for all organizations and indeed that small companies can gain most as it is easier for them to adjust their development process and to co-ordinate development activity. One company, for example, cited significant benefit in a project with just one developer, working for a month, using Scrum over two 2-week sprints.

3.3 Improved benefit levels

Another advantage of agile development over the waterfall approach is that it is rated more favorably in almost every assessment category used in the surveys. Moreover, this rating increased from 2010 to 2012, as company experience increased. In 2010, while two companies had taken an agile approach to development in one form or another for over 10 years, over a third of the respondents were still in the first 18 months of routine use. Reflecting that range, roughly 10 % of companies had completed over 50 agile projects¹ while 40 % had five or fewer. As might be expected, the level of experience had risen substantially by 2012, with most companies having completed at least 6–10 projects, and a quarter now in the largest 50+ category.

For respondents, there were perceived benefits in the quality of software produced and the process used to create it. Moreover, the benefits were experienced by all stakeholders. With process benefits, for example, each company (using agile development routinely) was asked to rate various process-related aspects of its approach on a scale of −2 (much worse) to +2 (much improved), assessed against previous experience with non-agile development. The questions related to activities with requirements, design, coding, testing, maintenance, and overall developer productivity. In 2010, there were 24 respondents, giving a maximum cumulative score in any question of +48, and a corresponding minimum of −48. In 2012, there were 30 respondents. Here, their scores have been scaled to the 2010 −48 to +48 range so that the results can be compared directly, as shown in Fig. 1.

All of the cumulative totals in the graph are positive, meaning that agile methods used are considered an improvement over non-agile approaches in all of the process areas assessed. Moreover, the score in 2012 was roughly 15 % higher than in 2010, suggesting a growing satisfaction with the process involved. Nevertheless, although seven of the ten areas are stronger, there was less satisfaction with requirements gathering and requirements management. Coping with evolving

¹ For the purpose of the survey, a ‘project’ was defined as a phase of development that concluded with a software release.

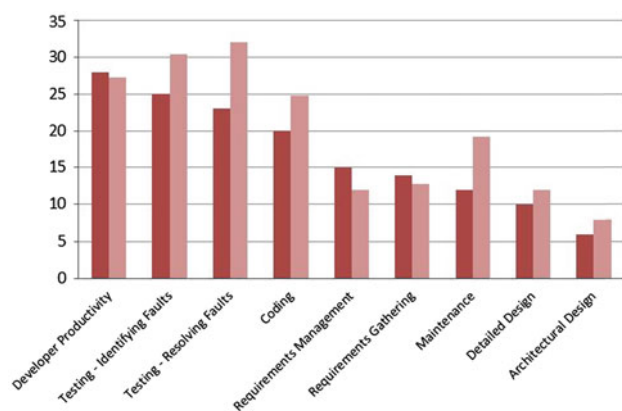


Fig. 1 Perceived process benefits of agile approach (sample = 24 & 30)

requirements is an inherently difficult aspect of agile development [24,25] and the drop in satisfaction may simply be as a result of a greater appreciation of that difficulty now that more projects have been completed. In 2012, four of the participating companies felt that the handling of requirements overall was worse with agile than in their previous approach.

Coding, testing and maintenance all show significant enhancement, suggesting that companies are becoming more comfortable with the basic development process. Also, agile approaches typically improve testing, particularly through test-driven development. Several companies commented that this aspect of agile had been their area of greatest gain.

The lowest ranking activities were in design—particularly architectural design, where only 10 of the 30 responding companies in 2012 felt there was improvement over traditional approaches. In fact, several companies reported that their teams had to work harder to achieve a satisfactory architectural design and three companies felt that agile methods were a retrograde step (up from two in 2010). The highly iterative nature of the development activity can lead to wandering designs if attention is not paid to this early on. Some companies devote entire iterations to architectural design. Likewise, refactoring activities help in continuous improvement, so bringing overall software designs back into check. This reflects the view of Cohn [26] that at least some architecture should take priority over functional development at the start of a project. It is an ongoing area of debate [27], however, and further investigation seems desirable to clarify how best to balance up-front design with evolving system development.

A similar pattern emerged in the assessment of perceived improvement to the software product through an agile approach, as summarized in Fig. 2.

As might be expected, product functionality and usability are the two qualities most often enhanced through an iterative agile approach, where the focus is very much on the value of the developed product. Note that the rating of all six

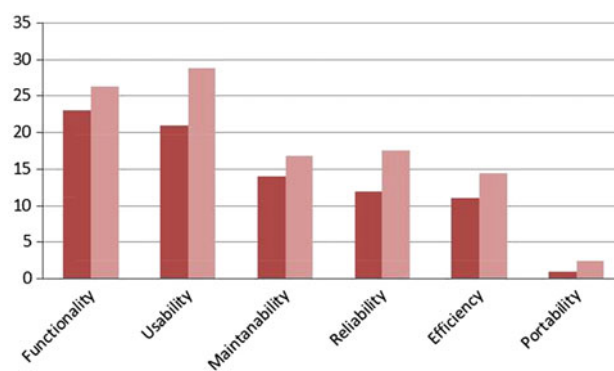


Fig. 2 Perceived product benefits of agile approach (sample = 24 & 30)

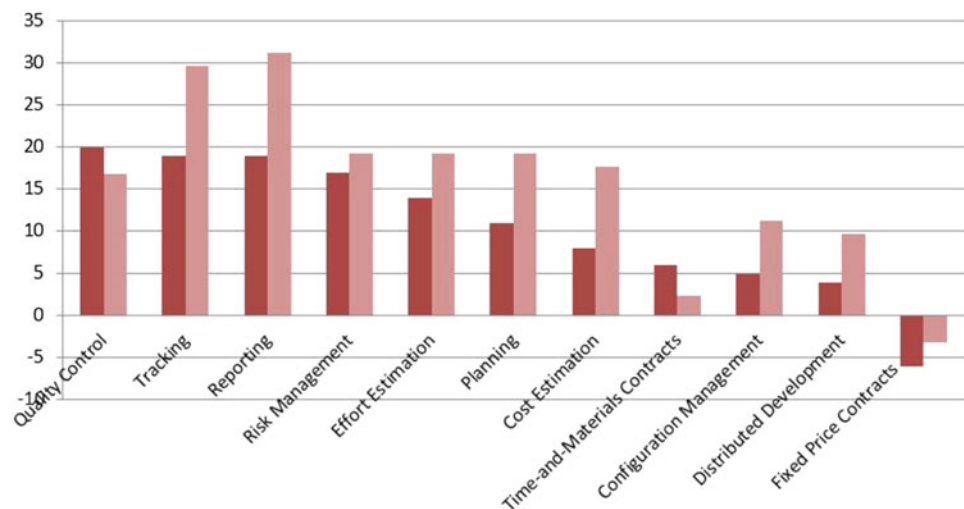
aspects of product quality increased from 2010 to 2012—by nearly 30 % on average—with only one company reporting any negative impact in any area: namely reduced ‘efficiency’ because of a generally weaker treatment of non-functional requirements in an agile approach.

The surveys asked questions to shed light on the degree to which project management (efficiency and/or effectiveness) had been improved using agile methods. Questions related to: performance of both effort and cost estimation; planning; tracking progress; reporting progress; managing risks; configuration management; managing geographically distributed teams; quality control; and contract negotiation (fixed price versus time-and-materials). Figure 3 presents the results.

Again, there is a positive story here, with all but two results showing an increase in satisfaction with project management between 2010 and 2012, and the scores for each area up by nearly 50 % on average. The reduction in the rating for quality control may seem anomalous in that product quality assessment is up. In looking more closely at the data, however, those companies reporting a drop in quality control also have product issues. This needs further investigation but seems connected to an increased emphasis on code production and programmer productivity in these companies. Finding the right balance between product quality and productivity is an on-going challenge in the agile approach.

Note that this is the only graph showing a negative cumulative score, which is for fixed price contracts. Three companies with experience of fixed price contract negotiation (common to both surveys) felt that agile software development performed significantly worse than their previous approaches. This was because of the perception that too much customer engagement, combined with the somewhat unpredictable nature of subsequent evolutionary development cycles, leads to difficulties with longer-term planning and estimation, which then hampers commitment to fixed-price contracts. It is also due to a natural fit between contracts and extensive upfront requirement analysis and the command

Fig. 3 Perceived management benefits of agile approach (sample = 24 & 30)



and control approach implied by contracts. The Agile Manifesto on the other hand explicitly states that priority should be given to “customer collaboration over contract negotiation.” For a discussion of alternative risk-sharing contracts, see [28].

These results indicate that managers generally benefit from an agile approach. The other two important stakeholder groups to consider are developers and customers.

3.3.1 Developer satisfaction

‘Developer interest’ was reported as a strong driving force in companies adopting an agile development approach. Anecdotally, one company indicated that “using an agile approach has added a lot to the software engineering role. Software developers like it and there is more engagement as a result.” With such buy-in from developers, it is interesting to explore some of the possible reasons. The survey asked three specific questions. These covered:

- *Visibility of progress*: did a clearer view of project progress provide a greater sense of achievement?
- *Clarity of role*: how important was the greater clarity of the developer role in the process?
- *Reduced rework*: was development more satisfactory through lower levels of rework?

The results are shown in Fig. 4.

The level of developer satisfaction increased by about 18% over the same measures assessed in the 2010 survey, with improvements in all areas and the same order of importance maintained. Better ‘visibility of progress’ scored highest, with over 85% of companies in 2012 reporting benefits.

Rework was the lowest scoring aspect in this category. About 15% of companies reported having more rework with

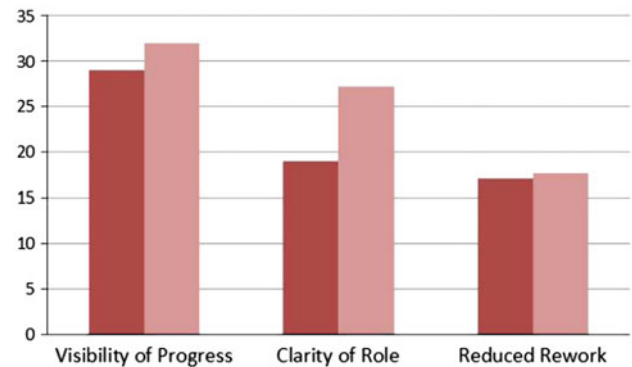


Fig. 4 Perceived developer satisfaction with agile approach (sample = 24 & 30)

agile approaches than with previous approaches, but 70% had reduced levels.

The clearer view of role is the most improved area, which seems to be directly related to the highly visible way that developer activity is presented and discussed.

3.3.2 Customer satisfaction

The survey asked four questions related to customer satisfaction: Did customers receive faster responses when agile methods were used? Was there better engagement? Did customers benefit from a reduction in the overall costs as a result of employing agile methods? Finally, did the customer end up with a better quality product? The cumulative results for these questions are presented in Fig. 5.

Overall benefits were reported in all areas except ‘reduced costs’, though the level of satisfaction increased in the 2012 survey. Roughly 50% of companies felt that agile development reduced costs, 40% were neutral and 10% reported increases.

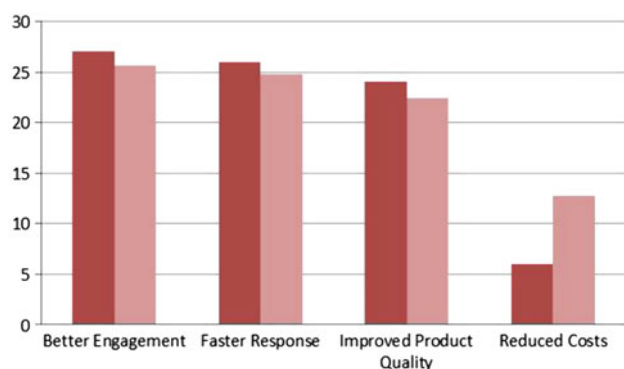


Fig. 5 Perceived customer satisfaction with agile approach (sample = 24 & 30)

In both surveys, about 80 % of respondents felt that the end product quality was improved through their agile process. The ultimate goal of any software company must be to deliver better product quality. This can improve customer satisfaction and build company reputation.

3.4 Survey limitations

In interpreting the above results a number of limitations should be taken into account:

- The study focused solely on the software industry in Northern Ireland, which is a very small geographical region. In this sense, it is unclear as to how generalizable the findings may be. However, many of the survey respondent companies were subsidiaries of large multinational organizations, where the internal practices may be expected to follow organization-wide directives. Therefore, it seems reasonable to believe that the results, whilst restricted to a small region, nevertheless reflect working practices and experiences representative of a much larger geographical base.
- The survey unit was a ‘company’ rather than its individual staff, with their personal experiences and opinions of agile working practices. This would be useful to investigate as part of a program of future work. However, it means that the answers to some of the questions in the survey will reflect the views of the individual respondent and may not always correspond to the consensus among other staff.
- The survey sampling process attempted to contact representatives of all software companies in the region but not all responded. In 2010, based on knowledge from other sources, it seemed likely that many of these companies had no agile involvement and this assumption was made in estimating agile adoption across the region. By 2012, some of the non-responding companies clearly had agile experience but chose not to contribute. This may be

because of difficulty with the approach so the missing information from this group could bring down the levels of benefit reported.

- The results combine views from all companies that were using an agile approach routinely for projects. There is no distinction, for example, between (1) small startups and large multinationals, (2) product-oriented and service-oriented companies, and (3) novices and experienced users. Such details were gathered, but the data has yet to be examined for evidence of correlation between circumstances and benefits. However, there were no obvious relationships sensed during the survey, so any qualification of the results that might emerge would be relatively minor.

4 Agile definition

Another significant insight from the surveys was the need for the agile concept to be more clearly articulated. In 2010, this seemed less important because of the widespread use of Scrum, which if adopted universally and in full, would have provided a de facto agile definition. In practice, the influence of Scrum seems to be diminishing. Of the 24 companies in the 2010 survey using agile development routinely (categories C and D), 11 were using Scrum, 8 a combination of Scrum with XP, and the remaining 5, various mixtures of Scrum, RUP, and DSDM. Thus, Scrum was the clearly dominant framework for the agile approach at that time. This matched the findings of the Forrester report, covering the same period, which identified Scrum as the overwhelming favorite methodology among adopters because of its simplicity, practicality and rapidly growing popularity [4].

The pattern in 2012 appeared similar, with 22 of the 30 companies in the C or D categories mentioning some link to Scrum. However, there was significant tailoring involved, with some companies now feeling that they had effectively developed their own methodology. Also, four companies mentioned the replacement of Scrum with a lean software development approach based on Kanban [11, 12]; one recently established company was set up with Kanban from the outset, and believed it to be superior to Scrum.

The relationship between agile and lean software development was an issue in the survey. For some respondents, the terms seemed interchangeable; for others they represented different development alternatives that were evolving separately; and a third group believed that ‘agile’ was the overarching concept, with lean ideas informing agile practice. This third interpretation was the one that respondents were encouraged to use, with agile development defined in the questionnaire as a way of “creating software in short iterative cycles, with regular reflection on both the product and the

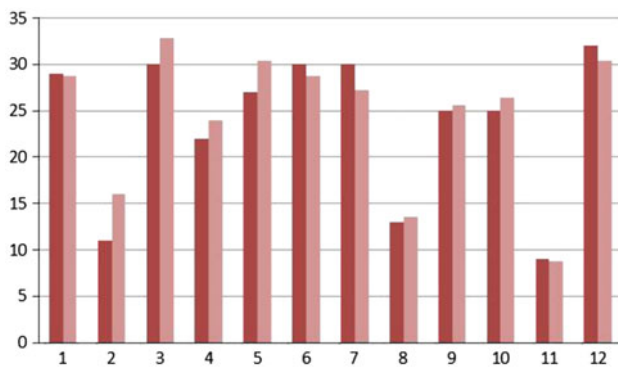


Fig. 6 Average agreement with the 12 agile principles (sample = 24 & 30)

process used in its creation”—this encapsulates both Scrum and Kanban.

Examples of the tailoring of Scrum revealed by the survey included: (1) running sprint planning in parallel with implementation so that one sprint could follow immediately after another; and (2) detaching the sprint retrospective from the sprint cycle to be handled more broadly as part of general process improvement across the organization. Such changes raise the question of their impact on the effectiveness of the revised process. It is therefore desirable to have clearer guidelines on what constitutes an agile approach so that the consequences of any variation introduced are understood.

As implied in Sect. 1, the agile Manifesto and associated Principles [3] do not define the agile approach adequately. One issue is that the Manifesto is too abstract (vague) to be a basis for confirming compliance. The Principles should complement the Manifesto and provide that detail but the authors seem to have little confidence in its content. For example, Cockburn (one of the original 17 authors), writing shortly after the Principles were developed suggested that: “These statements should evolve as we learn people’s perceptions of our words and as we come up with more accurate words ourselves. I will be surprised if this particular version isn’t out of date shortly after the book is published” [13, p. 219]. Unfortunately, no such revision has yet appeared. Further dissatisfaction with the Principles was indicated in the discussion session at the 2011 reunion celebrating the creation of the agile paradigm [29], where it was admitted that there was less focus on the Principles than the Manifesto in 2001, implying that some further thought was needed.

Participants in both the 2010 and 2012 Northern Ireland studies were asked to rate their support of the principles. This was mostly to confirm their understanding of an agile approach but had the interesting side effect of highlighting three principles that had much less support, as shown in Fig. 6.

The principles affected were:

- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage (2)
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely (8)
- The best architectures, requirements, and designs emerge from self-organizing teams (11)

This result suggests that these principles, in particular, require further thought, though perhaps support for Principle 2 may be increased simply by replacing the word ‘welcome’ with ‘accept’!

A recently published survey by Williams [30] shows a very similar pattern of results from a sample set of 326 participants. In particular, Principle 11 has least support and Principles 1, 3, 5, 7, and 12 have strong support. The only two notable differences are in Principle 2, which had a higher ranking in the Williams’ survey (7th of 12) and Principle 6 (the importance of conveying information face-to-face), which had a lower ranking (10th).

Williams also carried out a second survey with the same participants proposing adjustments to the principles. The changes involved rewording eight principles (1, 9, 10 unchanged) and deleting Principle 3, because it is largely the same as Principle 1. This proposal, however, does not seem to have been well received, as the overall conclusion to the paper is that the original principles “still deliver solid guidance for software development teams and their projects.”

The work does suggest an underlying belief that improvement is possible, though perhaps not through a direct adjustment to the existing Manifesto and Principles.

5 Conclusion

Despite agile software development being considered a mainstream paradigm from around 2010, the concept, and hence its strengths and limitations, are still relatively unclear. One way to improve clarity is through studying actual practice across a wide range of organizations. This paper describes two such studies, performed using detailed surveys of software development practice in companies in Northern Ireland in 2010 and 2012. The main aim of the first survey was to identify the level of agile adoption in the region, while the second focused on understanding how agile principles and practices were maturing. Although the focus was on a relatively small region, it was believed that the results would be of wider significance because the companies were largely competing in a global marketplace and, by inference, reflected the attitudes and practices of that marketplace. Having a rea-

sonable match with the Williams' survey on agile principles [30] provides some support for this view.

The paper focused on two main observations drawn from the surveys:

- The agile paradigm seems set to replace the traditional waterfall model as the standard approach to software development; and
- A clearer definition of agile development, identifying its intrinsic benefits and limitations, is needed to help organizations gain optimal benefit from its use.

Each observation was supported by results from the surveys. Current work is concerned with the design of a new framework for agile development, building on the Agile Manifesto and Principles, incorporating the key elements of the currently popular agile methods, and returning to principles identified in the Toyota Way [31] that was a strong influence on the current agile movement. This work also extends agile concepts to the organizational context in which software is produced [32].

Acknowledgments This work draws on the results of surveys sponsored by Invest Northern Ireland in 2010 and 2012. The authors are grateful to the companies who participated, particularly those individuals who gave their time generously in helping to complete the questionnaires. The companies involved were: 8Over8, AllState (NI), Andor, Asidua, Autonomy, Biznet Solutions, Bombardier, British Telecom, BTi Systems, Citi Belfast, Consilium Technologies, Core Systems, CyberSource, Datactics, DTS, Equiniti-ICS, Etain, Fidessa, First Derivatives, HBoS, HP (3PAR), ICONI Software Ltd, IBM (Intelliden), IBM (Q1 Labs), inTune Networks, Kainos, Kana (Lagan), Kofax (Singularity), Liberty IT, Lucid, Mediasmiths Forge, mFormation, Navinet, NiSoft, NYSE, OmniSoft, Openwave, Opt2Vote, Oracle (ATG), Parity Solutions Ltd, Replify, SkillSoft, StreamOn, TADL, Tascomi, TextHelp, Tyco International, and Wellington IT Solutions.

References

1. BBC (2013) How Dave Brailsford and Team Sky stormed cycling. <http://www.bbc.co.uk/sport/0/cycling/21331484>. Accessed June 2013
2. Chrissis MB, Konrad M, Shrum S (2011) CMMI for development: guidelines for process integration and product improvement. In: SEI series in software engineering, 3rd edn. Addison Wesley, Boston
3. 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 RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) Manifesto for agile software development. <http://agilemanifesto.org/>. Accessed June 2013
4. West D, Grant T (2010) Agile development: mainstream adoption has changed agility. Forrester Research, Chicago
5. Agile Alliance (2013). <http://www.agilealliance.org/the-alliance/what-is-agile/>. Accessed June 2013
6. Ambler SW (2010) Agile software development. In: Encyclopedia of software engineering, library. Taylor & Francis. doi:10.1081/E-ESE-120044477
7. Wikipedia. Agile software development. http://en.wikipedia.org/wiki/Agile_software_development. Accessed June 2013
8. Schwaber K, Beedle M (2008) Agile software development with Scrum, Pearson, Education
9. Schwaber K (2004) Agile project management with Scrum. Microsoft Press, USA
10. Sutherland J, Schwaber K (2013) The Scrum papers. <http://scrumfoundation.com/library>. Accessed June 2013
11. Poppendieck M, Poppendieck T (2003) Lean software development: an agile toolkit. Addison-Wesley, Boston
12. Anderson D (2010) Kanban. Blue Hole Press, Bahamas
13. Beck K, Andres C (2004) Extreme programming explained: embrace change. 2nd edn. Addison Wesley, Boston
14. Tudor DJ, Tudor IJ (2010) The DSDM Atern Student Workbook, 2nd edn. Galatea Training Services Ltd., UK
15. Cockburn A (2001) Agile software development. Addison Wesley, Boston
16. Ambler SW (2013) Surveys exploring the current state of information technology practices. <http://www.ambysoft.com/surveys>. Accessed Jan 2013
17. VersionOne, State of Agile Development, 6th Annual Survey, http://www.versionone.com/state_of_agile_development_survey/11/. Accessed Jan 2013
18. Vijayasathya LR, Turk D (2008) Agile software development: a survey of early adopters, J Inf Technol Manag 19(2):1–8
19. Laanti M, Salo O, Abrahamsson P (2011) Agile methods rapidly replacing traditional methods at Nokia: a survey of opinions on agile transformation. Inf Softw Technol 53(3):276–290
20. ISO/IEC 9126–1 (2001) Software Engineering product quality—Part 1: Quality Model, International Organisation for Standardisation
21. Office of Government Commerce (2009) PRINCE2:2009 Manual—Managing successful projects with PRINCE2
22. UK Cabinet Office, Government ICT Strategy, <http://www.cabinetoffice.gov.uk/content/government-ict-strategy>. Accessed June 2013
23. UK Cabinet Office, Government ICT Strategy—Strategic Implementation Plan, http://www.gov.uk/government/uploads/system/uploads/attachment_data/file/85971/govt-ict-sip.pdf. Accessed June 2013
24. Paetsch F, Eberlein A, Maurer F (2003) Requirements engineering and agile software development, In: Proceedings Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2003, IEEE, pp 308–313
25. Ramesh B, Cao L, Baskerville R (2012) Agile requirements engineering practices and challenges: an empirical study. Inf Syst J 20(5):449–480
26. Cohn M (2005) Agile estimating and planning. Prentice Hall, New Jersey
27. Abrahamsson P, Ali Babar M, Kruchten P (2010) Agility and architecture: can they coexist? Software. IEEE 27(2):16–22
28. Mølloken-Østfold K, Furulund KM (2007) The relationship between customer collaboration and software project overruns. Proc. Agile pp 72–83
29. Sutherland J, Agile Manifesto 10 year reunion: full video, <http://scrum.jeffsutherland.com/2011/11/agile-manifesto-10-year-reunion-full.html>. Accessed June 2013
30. Williams L (2012) What agile teams think of agile principles. Commun ACM 55(4):71–76
31. Liker JK (2004) The Toyota Way: 14 management principles from the world's greatest manufacturer. McGraw-Hill, New York
32. Bustard DW (2012) Beyond mainstream adoption: from agile software development to agile organizational change. In: Proc 19th International Conference and Workshops on Engineering of Computer-Based Systems (ECBS 2012) Novi Sad, Serbia. IEEE Press, NJ, pp 90–97