

# **Successful extreme programming: Fidelity to the methodology or good teamworking?**

## **1. Introduction**

Teamwork is increasingly considered vital in modern software engineering. This belief is a central driver behind the agile development movement, and organizing the work on a more collective basis and involving the client throughout the design process became the hallmark of Extreme Programming (XP) (Beck et al., 2004 [1]) and other similar methodologies. Core XP practices such as pair programming, collective code ownership and short, frequent meetings are inherently collaborative, while the nature of the workflow under agile methods implies a greater level of task interdependency and collective ownership than traditional software development methods, which emphasize individualized responsibilities and task allocation. However, whilst greater collective actions should result from the adoption of agile methods, the need for interdependent working may place a premium on relationships that are cooperative, underpinned by trust, and free from destructive conflicts.

There is no guarantee that teams will even adhere to the XP methods (Fitzgerald et al., 2006 [2]; Mangalaraj et al., 2009 [3]). Equally, there is no guarantee that teams are cohesive and work well together. Nominally all groupworking is teamwork, as formally defined thus: “individuals who perform tasks in an interdependent fashion to meet the goals of an organization, and who can readily distinguish themselves from other work groups” (Turner et al., 2010, p. 720 [4]). But teamwork does not automatically arise. Teams may not be able to achieve the required level of

interdependent working or have the qualities that have been found to characterize high-performing teams. We are thus likely to find considerable variation in the fidelity in the use of agile methods and degree of teamworking amongst software engineering teams. This is a virtue for research purposes as it provides us with the environment for natural experiments designed to assess the relative importance of the agile software engineering methodology and teamwork. We can compare the effects of variability in adherence to XP methods with those in the degree of teamwork. We can thus establish whether it is adherence to the XP methodology per se that is the main active ingredient in the success of XP software development, or whether well-established team dimensions (e.g. Acuña et al., 2008 [5]; Campion et al., 1996 [6]; Cohen and Bailey, 1997 [7]; Sawyer, 2001 [8]) are more important to success, or finally if a combination of these is what really matters.

Perhaps successful teamwork has a catalytic effect on the methodology, enhancing the value of adherence to it; or alternatively, adherence to an XP methodology will lead inevitably to groups following good teamworking principles or at least having high morale. Additionally if team characteristics are important in one way or another, we would want to know if it is the general characteristics of teams or the distinctive agile team processes, such as the collective ownership of work, which are significant.

In this paper we report research that uniquely attempts to evaluate the relative roles of agile techniques and more general team factors in explaining differences in the performance of teams using the XP agile method. This includes assessing whether XP's core techniques only work well when the essential requirements for good teamwork are in place. The study is of 40 small commercial development projects that involve student teams using XP to varying degrees of compliance. We looked at team characteristics and eventual project outcomes to assess the relative role of XP

techniques, the teams' general characteristics and their cohesiveness. In so doing we are especially contributing to filling two main substantive gaps in the empirical software engineering literature on process innovations that Mangalaraj et al. [3] identified: their post-adoption performance and the role of teams in them. Such issues are core to any theory of agile methods, as a vital element of this must be its effect on the performance of individuals, projects and organizations.

## **2. Conceptual background**

### *2.1. Past research*

Past literature on agile methods has concentrated on advocacy (e.g. Highsmith and Cockburn, 2001 [9]; Nerur et al., 2005 [10]; Sharp, 2010 [11]), outlining its methods (e.g. Beck, 1999 [12]; Holcombe, 2008 [13]), and gauging reactions to it from developers and customers (e.g. Ilieva et al., 2004 [14]; Mannaro et al., 2004 [15]; Svensson and Höst, 2005 [16]). Evaluations of these methods have concentrated on specific practices such as collective ownership and coding standards (Maruping et al., 2009 [17]) or most notably pair programming (including Balijepally et al., 2009 [18]; Dybå et al., 2007 [19]; Hannay et al., 2009 [20]). Nonetheless, Dybå and Dingsøyr (2008 [21]) found 36 high-quality studies among the initial wave of studies (2003–5) that sought to evaluate aspects of agile development systematically and holistically. These concentrated on the introduction and adoption of agile methods, human and social factors, customer and developer perceptions, and comparative studies that assessed different development methods.

The studies within the human and social factors category concentrated on describing the characteristics of teams using agile methods, and in most cases XP. Robinson and Sharp (2004 [22]) highlight the importance of shared responsibility and trust within the group as contributing to what they observed to be effective extreme programming. Young et al. (2005 [23]) investigated the distribution of personality types required in XP teams. Such studies did not systematically link team characteristics to performance nor compare the relative importance of different dimensions of teams.

The studies most directly concerned with performance were in the comparative group, as they compared agile with traditional software engineering. Three out of four of these (Dalcher et al., 2005 [24]; Ilieva et al., [14]; Layman et al., 2004 [25]) found XP was superior in terms of productivity as measured by lines of code per hour, whilst in these and another study (Macias et al., 2003 [26]) quality was also better. Again, none of these studies evaluated whether certain features of agile methods have more effect on performance than others.

Subsequently, Conboy (2009 [27]) designed a study to isolate which features contribute to agility, as he was concerned that some of the methods associated with agile technology may not contribute to agility or cohere as well as was envisaged. He asked members of various project teams to rate the contribution of specific practices to a project's agility. The practices that were less likely to be seen as contributing to agility included on-site customers and pair programming. But the general relevance of teamwork to agility was not explored in this research.

The motivation behind Conboy's study was to develop a theoretical foundation for agile methods or at least to reduce its conceptual shortcomings, since "agile method practice has led research, with the creation, promotion, and dissemination of these

methods almost completely due to the efforts of practitioners and consultants” (Conboy, p. 329 [27]). His concern is to develop a theoretical foundation for agile methods, that is, an agile method rooted in practices that are known to increase agility and hence perform well in uncertain situations. Our concern with theoretical development is similar but our starting point is different, as we take the agile methodology as commonly espoused and investigate which elements of it contribute to the success of software engineering projects.

## *2.2. XP software development*

Our study is focused on the use of the XP approach to agile software engineering and comparing the contribution its core elements make relative to general team factors in the performance of software teams. We thus now introduce the XP methodology and how we are measuring its components and those of the core concepts of teams.

XP is a specific agile methodology for which the primary source for identifying its main features remains the Agile Manifesto of 2001 (Cunningham, 2001 [28]). The characterization of agile was based on what the Manifesto’s authors detected to be common to the emerging less-rigid approaches to software development. These practices were identified largely in contrast to traditional methods that were characterized as plan-driven and as dominated by requirements being agreed at the outset, the production of on-going documentation, and adherence to a highly specified plan of work.

The contrast between agile and plan-driven methods was characterized in the Agile Manifesto by domination within agile methods of:

- individuals and teams over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contract negotiation,
- responding to change over following a plan.

Whilst labels such as its “individuals and teams dominating over processes” sound glib – not least as processes are created and followed by people – they capture key elements of the method that the architects of the Agile Manifesto were trying to encourage.

Above all, agile methodology entails delivering software outputs in a number of stages, known as iterations, which are delivered into production, and this acts as a feedback mechanism on a project’s performance. Through these iterations, agile methods aim to deliver greater functionality as the frequent releases, coupled with active client involvement, ensure that agile teams react to any unexpected changes or correct problems arising from any misunderstandings about the client’s requirements at any stage of the project. The methodology also allows for uncertainty in the client’s mind about the users’ requirements. The heavy involvement of the client, coupled with pair programming and collective ownership, make much of the documentation associated with plan-driven methods unnecessary (Nerur et al., p. 75 [10]).

The vital importance of teams working to the agile methodology is most apparent in the 12 principles that the Agile Manifesto also outlines. Most salient is Principle 11 that “The best architectures, requirements and designs emerge from self-organizing teams” (Cunningham [28]). Allied to this, Principle 6 stresses the efficiency and effectiveness of face-to-face conversations within the team, and Principle 5 posits the

need to build teams with self-motivated individuals and give the teams and the individuals high levels of autonomy.

All software engineering typically involves working in teams and some level of client involvement in the design of the project. But the intensity and nature of both of these is assumed to be different in agile methods (Boehm, 2002 [29]; Cockburn and Highsmith, 2001 [30]; Highsmith and Cockburn [9]). Typically under the plan-driven method, individuals or teams work on one part of a larger project and the teams operate in a hierarchical organization with managers controlling the design and distribution of tasks. Under agile methods, teams will have autonomy over how they approach the project and division of labour, though they may have a working leader whose role is to coordinate and liaise with the rest of the organization of which they are a part.

Nerur et al. (p. 25 [10]) thus conclude that the empowerment and minimum individual role assignment of team members in agile teams “enables them to self-organize and respond with alacrity to emergent situations”. The client is nonetheless central to gathering requirements and feedback on iterations (Hoda et al., 2010, p. 78 [31]) and to stimulating creative and innovative solutions rather than formulaic applications of past solutions (Conboy and Morgan, 2011 [32]).

XP is an agile method that particularly emphasizes team processes and ascribes responsibility for the output to the team, as it prescribes three XP-specific team processes: collective coding standards, collective code ownership and continuous integration (Beck et al. [1]).

*Collective standards* requires that all developers write and maintain software code in a common and consistent form and thus provide a common base by which to understand units of code. *Collective ownership* allows any member to change any part

of the software code at any time and encourages all members to take responsibility for all the software code rather than, as in traditional methods, to be responsible for just part of the whole. *Continuous integration* entails continuous quality control as small pieces of work are tested frequently to provide continuous feedback on the project's progress and to improve the quality of software, with the expectation that it will reduce delivery time. It replaces the traditional practice of applying quality control only after completing all development.

The team-based practices are three of 12 core XP practices as defined by Beck et al. [1], which Williams et al., 2004a [33] classify into four categories (as shown in Table 1). The team-based practices are given the name *Teaming*. The other categories are foundations, customer planning and craftsmanship.

**[Insert Table 1]**

*Foundations* focuses on testing (unit, customer and test-first), refactoring (where code is redesigned without adding functionality), and pair programming (where in its pure form two developers work at one computer). Foundations is measured by five items: automated unit tests, customer acceptance tests, test-first design, pair programming, and refactoring. *Customer planning* centres on the involvement of the client in the planning and release of output to them. It is measured by the use of the planning game (where the customer helps to choose which functionality will be developed next), customer access (also known as the practice of having an onsite customer, is concerned with making it easy for developers to ask questions of the customer), short releases (when the product is delivered to the customer every two weeks or so), and stand-up meetings (which should be a ten-minute meeting each day). *Craftsmanship* is concerned with sustainable pace (the team puts in a consistent number of working hours), simple design (the implementation of the simplest possible

solution to any problem), and use of metaphor (a simple overall conceptualization of the proposed system), and is measured by these practices.

These four categories have formed the basis of a measure of adherence to XP developed by Williams et al. [33] (2004b [34]); see also Krebs, 2002 [35]) and are known as the Shodan Adherence Survey. This metric is seen by Williams et al. as vital for practitioners and researchers to benchmark their use of XP, as well as providing a useful tool for helping both to understand XP better.

The categories are particularly useful if we are to test the argument of proponents of agile methods (Beck, 1999 [12]) – and some of its detractors (Stephens and Rosenberg, 2003 [36]) – that sets of practices such as XP should be used as a package, and not piecemeal. If it is the case that the practices of XP are, as Stephens and Rosenberg say, like a “circle of snakes”, then when any practices are neglected the process as a whole is likely to fail (p. 2561).

On the surface there may appear to be tension between a prescription that XP or other agile methods should be applied in their entirety and agile’s encouragement of flexibility (Karlsson and Ågerfalk, 2009 [37]). If the tension is real and is resolved through the flexible adoption of a particular agile approach we would then expect a great range of approaches in practice such that almost anything that diverges from a rigid plan-driven approach is agile. However, the phrase ‘agile methodology’ is then in danger of becoming meaningless. In our view any tension is more imaginary than real, as agile practices are the means to be flexible throughout the development process in order to achieve high quality, cost-effective solutions, particularly where there are uncertain or changing customer requirements. In Hoda et al.’s [31] terms, “teams can cleave to the *principles* of Agile development” (p. 78) even if they tailor their use of practices to the context.

The use of a metric indeed carries an assumption that agile methods can be specified at least to the extent that they can be identified in use, particularly by adopters. This need not mean that they are rigid rules but rather, using Ilivari and Maansaari's (1998 [38]) characterization of the essence of software methodologies, they are ideals, not necessarily highly prescribed sets of regulations. Yet the assumption in using such a metric is that any tailoring of these ideas maintains the integrity of the approach. For example, that automatic testing is done, but that its form may vary so it is compatible with the language of the software, or that team meetings regularly occur though their timing will be consistent with the demands of the project and the team's extra-work commitments. Nonetheless it is precisely because software engineering teams may not use all the elements of an agile method to the full that we can assess whether the degree of adherence affects the outcomes of its work.

The implication of the principled or synergistic approach to agile methodology is that the more practices a project has, the more likely it will be successful and the greater the level of team performance. But it may still be the case that some dimensions of XP contribute more to the final outcome than others. Metrics such as the Shodan survey enable us to test this. They allow us to assess if agile methods are superior by identifying whether the best performing teams are adhering to the method and the poor performing teams are those that do not follow it comprehensively. But metrics also permit us to test if some practices are contributing to performance differentials more than others while others perhaps are adding little, so it makes no difference if teams discard them. There is also the possibility that, when tested alongside general measures of teamwork, the importance for performance of agile methods is found to be subordinate to its associated teamworking. We now consider the nature of teamwork and measures of it.

### 2.3. Teamwork

We treat Williams et al.'s [33] teaming element as the XP-specific team factor, and contrast this with general characteristics or processes of teams that have been found in seminal psychological research to be related to team performance. These general team factors typically refer to team characteristics such as *team cohesion*, *cooperation* and *sharing of workload*, which are found to varying degrees in all teams.

We follow the widely accepted approach to these factors, which is the *input–process–outcome* model of team effectiveness (Mathieu et al., 2008 [39]; McGrath, 1984 [40]). Inputs are antecedent factors that enable members' interactions, and processes are the team's interactions that are directed towards task accomplishment. In Mathieu et al.'s terms, the inputs “combine to drive team *processes*” (p. 412). *Outcomes* are team-level performance measures, but may include performance behaviours such as team process improvements. Several characteristics of a team, including its level of cohesion, can be considered as emergent properties of the team's interaction, which become part of the input to it functioning effectively.

The input–process–outcome model is the foundation of a theory of successful team performance. Campion et al.'s (1993 [41]) classification of team inputs and processes remains the most widely accepted and comprehensive conceptualization and operationalization of the key factors for successful outcomes of teamwork. They identify these factors through a review of literature on teams. It is these that will form our conception of general team factors.

### 2.3.1. Team inputs

Campion et al. [41] identify three types of team input: the task design, level of interdependence, and composition of the team. Of the task design inputs, two are especially important in the context of agile methods: (a) the *degree of participation*, the extent to which all members participate in key task and process decisions in the team, and (b) *task variety*, the extent to which members have variety in their work with concomitant chances to learn and share in the most interesting tasks. These factors are widely assumed in psychology to be associated with task performance as well as people's well-being (Campion et al. [41]; Hackman and Oldham, 1975 [42]; Warr, 2007 [43]).

The main form of interdependence is *task interdependence*, which is concerned with the extent to which members of the group depend on each other for the accomplishment of their work. It is argued that task interdependence should be associated with higher performance as it engenders a sense of collective responsibility and the rewards accruing from the group's accomplishments should be greater. It can also be assumed that these rewards will increase satisfaction because they increase the social contact people have with each other and the meaningfulness of work (Warr, pp. 193–202 [43]), and this in itself may enhance performance.

A frequently considered *compositional* factor is the degree of heterogeneity in the experiences, interests and abilities of team members. Diversity is particularly thought to be valuable when projects involve a multiplicity of tasks and require a range of skills. The evidence of this having a positive effect on performance thus far is, however, limited as some studies have found no relationship, including Campion et al.'s [41] study. Yet these studies may include tasks that were more standardized or

required less creativity than is the norm in software engineering, which may explain the lack of association.

Diversity in age (Campion et al. [41]) and tenure (Cohen and Bailey [7]) was found to be related to performance. Whilst the groups in our study did not have a great deal of diversity in age, and tenure was not applicable to them, these variables may be treated as proxies for experience. The diversity in experience of the groups was mainly in their use of software languages and methodologies, and to a lesser extent working in groups, but this was not extensive.

### 2.3.2. *Team processes*

Campion et al. [41] identified four team processes that may be positively associated with group performance: *potency* – the belief that a team can function effectively and fulfill its goals (Guzzo et al., 1993 [44]); *social support* – members helping and positively supporting each other; *workload sharing* – equality of input to the group, as opposed to social loafing or free-riding; and *cooperation* – effective communication, cooperative working and sharing of information. These are all commonly identified in the social psychology literature as contributing to effective teamworking, which should have beneficial effects on task accomplishment.

### 2.3.3. *Team cohesion*

Cohesion is, in lay terms, the chemistry of the group. It is concerned with the extent to which the members of “a group stick together and remain united in the pursuit of goals and objectives” (Carron, 1982, p. 124 [45]). Cohesion entails individuals perceiving that they belong to a particular group and having strong affective feelings towards the group so team morale is high (Guzzo et al. [44]). In the

agile team context, Whitworth and Biddle (2007 [46]) posited and showed that high cohesion is likely to be associated with “strong feelings of excitement”, that the team “gelled” and individuals “clicked” (p. 27).

Meta-analyses of team cohesion’s relationship with performance outcomes have revealed varying correlations, but the most recent by Beal et al. (2003 [47]) produced a corrected mean correlation of 0.17. Various dimensions of cohesion, for example task or interpersonal, were all found to be significantly related to performance. An earlier meta-analysis showed that this relationship was not moderated by group diversity, that is, cohesion does not have more effect in diverse teams (Webber and Donahue, 2001 [48]). A longitudinal study has shown that the direction of causality is from cohesion to team performance (Kilduff et al., 2000 [49]), but within the lifetime of a project it is likely that they have a reciprocal relationship, as high performance levels of subtasks increase cohesion, whilst groups struggling to achieve outputs may become disunited.

Whilst cohesion and the process characteristics are assumed to have direct effects on performance, it may be that they also, or even solely, moderate the relationship between key input variables. For example, Shin and Park (2009 [50]) found that group cohesion strengthened a positive relationship between the competency level of groups and performance in a Korean manufacturing company.

### **3. Models of the relationship between agile methods and teamwork in models of performance**

We have outlined the concepts used by Williams et al. [33] and Campion et al. [41] to characterize agile and team processes in order to provide us with a starting point for

empirical investigation of whether agile methodology can outperform other methods. Using our triad of concepts – XP practices, XP-specific team practices, and general team factors – we will now outline a set of competing hypotheses about the performance of projects, which are differentiated by the hypothesized relationships amongst the three concepts.

### *3.1. Additive models of agile performance*

If genuine teamwork is integral to the success of agile, and minimal adherence to its collective protocols is not sufficient as Highsmith and Cockburn say (2001 [9]), then the XP practices and team factors are complementary. Each adds an ingredient that the other does not. This, the simplest model of performance, thus expects the XP practices, the XP-specific team factor, and the general team factors (design, processes and cohesion) to each have some positive effect on performance. This can be formally stated as:

***Hypothesis 1:*** *XP practices, XP-specific team factor, and general team factors are each uniquely associated with team performance.*

### *3.2. Dominant factor models of agile performance*

However, a competing set of hypotheses to the simple additive model could firstly be founded on the assumption that one type of practice drives the performance of the

team and hence dominates the other. There are grounds for arguing that any one of the triad plays this role.

Consistent with group literature that has identified the team characteristics that affect team performance (e.g. Cohen and Bailey [7]; Campion et al. [6,41], we might expect general team characteristics to dominate, thus:

***Hypothesis 2:** General team factors are uniquely associated with team performance.*

However, within the team literature, Gersick (1998 [51]) in particular emphasized that the methodology adopted by the team may be most important for its development. If we follow this emphasis, we would expect that the XP practices would be most strongly associated with performance. We thus will test:

***Hypothesis 3:** XP practices are uniquely associated with team performance.*

Yet it may be that the distinctive features of XP practices are the way they engender collaboration through collective ownership. This would imply that XP-specific characteristics dominated, thus:

***Hypothesis 4:** The XP-specific team factor is uniquely associated with team performance.*

### *3.3. Multiplicative models of agile performance*

A second competing set of hypotheses can be constructed on the assumption that there are synergistic effects amongst the practices. The thesis that agile practices should be used as a package and that its social side is as important as its techniques, which is widely expounded in the literature (Stephens and Rosenberg [36]), typically implies that all the practices are synergistically related. The argument is that, on the one hand, the techniques that define XP are crucial but, on the other hand, their success depends on groupworking. This implies that the impact of the approach will be enhanced by effective teamwork, which in turn suggests a multiplicative model, not the additive model of Hypothesis 1, thus:

***Hypothesis 5:** XP practices and the interactions between them and (a) the XP-specific team factor and (b) general team factors, are associated with team performance.*

Since cohesion is a fundamental team concept, sometimes treated as the most important small group variable (e.g. Lott and Lott, 1965 [52]), it may be that this is critical in binding not only the team but also the workings of the agile techniques. This would imply that interactions involving team cohesion are the critical elements in a theory of agile performance.

It can be hypothesized that group protocols, norms and values will not develop or at least affect individuals if they do not perceive themselves to be part of a group and to value membership of it. This suggests that of the interactions in Hypothesis 5, those between XP practices and cohesion (within the general factor vector) ought to be the most powerful. It might even be argued that if the team is not cohesive, its use of XP methods will have little or no effect as they depend on the successful negotiation and

development of ways of working, that is, group processes adequately developing voluntary norms and shared understandings. This suggests that cohesion is the main effect and there is a strong interaction between XP methods and team factors, thus:

***Hypothesis 6:** Team cohesion and the interaction between it and XP practices are uniquely associated with team performance.*

If cohesion is so decisive, it might even be that divisive groups apply the agile method so badly that their performance is substantially adversely affected, in which case the interaction will dominate, thus:

***Hypothesis 7:** The interaction between team cohesion and XP practices is associated with team performance.*

### *3.4 Mediation models of agile performance*

A final approach is to posit one or other of the factors as driving both the performance and the level of the other factors. The most plausible possibility is that adherence to the agile protocols drives the teamworking and cohesiveness of the team. In commonly accepted research design terms, this implies that team factors mediate the relationship between the dominant factor, agile methodology, and performance.

An argument for this might begin with two assumptions. First, that a minimal level of compliance to an organizationally mandated development methodology, in this case XP, may be expected of project teams. Second, that the level of cohesion or even team characteristics is not independent of the degree of adherence to agile methods. In

group theory terms, the procedures and norms of the agile method can be treated as institutional norms that are mandated by external authorities (Scott, 2001 [53]). Since these act as demands and not just constraints on what the team should do, they may drive teams to develop strong norms and protocols. Initial discussion of the methods is likely to be decisive in this and particularly the establishment of the momentum through which a sense of belonging, team morale, and the more detailed norms required to embellish and enact the XP methodology develop. This logic implies that the XP methods are the main drivers of performance and that the team cooperativeness and perhaps other team characteristics follow from them.

This would mean that the team factors mediate the relationship between XP methods and performance, as specified by Hypothesis 3, and in the extreme have no effect on performance in the absence of high fidelity to XP methods. This can be represented thus:

***Hypothesis 8:** XP practices are uniquely associated with team performance, and (a) the XP-specific team factor and (b) general team factors, including cohesion, mediate this relationship.*

A more specific version of this hypothesis would posit that the XP-specific team factor affects team processes, and perhaps above all else the level of cohesion, and therefore these mediate the XP-specific team factor–performance relationship.

***Hypothesis 9:** The XP-specific team factor is uniquely associated with team performance, and team processes and cohesion mediate this relationship.*

#### **4. The study design**

We designed a study to test the nine competing hypotheses through assessing the role in the performance of software engineering teams of XP practices, the XP-specific team factor and general characteristics of the teams relative to the XP methodology. In outlining these competing hypotheses we are adopting the “method of multiple hypotheses” that Chamberlin (1890/1965 [54] presented to the US Academy of Science in 1890 as a way to encourage more rigorous and impartial research. Tests of the competing hypotheses are designed to reveal positive support for one hypothesis but in so doing they also enhance the support for this by excluding the alternatives.

##### *4.1. The objective of the study*

To test the competing hypotheses we assess the association between the performance of project teams and measures of their agile methods application, adherence to agile-specific team protocols and general team factors.

##### *4.2. The context of the study*

The study involves teams of undergraduate and postgraduate student software engineers, working within a university-owned software engineering house on commercial or internal projects where the XP method is the mandated development methodology. The house was set up with intertwined objectives, to provide: (a)

student projects that were not library- or laboratory-based but had outputs that would be applied; (b) clients with good value-for-money services particularly for small- and medium-size businesses in the local region; (c) the university with a laboratory for observing software development in action and hence pursue its interest in empirical software engineering; and (d) a small income stream to supplement the computer department's teaching and research incomes.

All the teams are co-located and work in one of two large computer rooms with small rooms available for team meetings or meetings with the client. The 'real life' nature of the projects means that students present their completed software to their clients at the end of the project with a view to their deploying it.

The clients submit an assessment of the teams' performance regardless of whether its solution was selected for implementation. Along with the teams' lecturers, the software house manager monitors their progress and helps them with any problems or queries they have.

Each team's mark for examination purposes is based on assessment by two lecturers, who first independently give the project a score and then meet face-to-face to agree a common assessment. The students in each team are given an individual grade, which is based on the assessment of the team's output but with some allowance being made for individual differences in contribution.

The team assessment takes account of the client's report. Criteria for assessing the product's performance included: the quality of the code; test documentation (including automated tests); ease of the product's use; understandability; completeness; innovativeness; robustness; and the happiness of the client with the solution. Although agile methods eschew documentation, and the teams produced little engineering documentation as the project progressed, they were expected to

produce guidance on the software's use and maintenance for the client. It is this that is being assessed in the documentation component of the assessment as it is viewed as a critical part of project delivery.

Our sample consisted of 40 teams, each with a minimum of three individuals, and the median was four. Projects ran for between 10 and 22 weeks, some concurrently, with students advised to work for 15 hours each per week, although students' timesheets typically reported a higher number of hours. All 40 teams in the two years of our research project provided all the data that we required. The total number of students who participated in this study was 141. The projects were of varying complexity and a variety of programming languages were used across the projects, but tended to be one or other of JAVA, PHP, JavaScript, and Flash/Actionscript.

Clients were external and fee payers, but some in our sample were within the software house or university. In the case of external clients the vast majority was small business owners but a small minority was larger businesses commissioning small projects to support their staff or explore a new technology. The commercial value of each ranged from \$20,000 to \$40,000.

The projects that the teams in our study completed fell into five categories:

1. Business applications for mobile devices; these often included a synchronization facility to a central database,
2. Data-driven websites with document management and scheduling-type features,
3. Ecommerce websites, often with a stock control element,
4. Desktop data-driven business software,
5. Online and desktop-based eLearning tools.

The students attended formal courses in agile methods and a set of sessions on the nature of the projects and agile methods, which included detailed examples of previous projects. Both these inputs included exercises that the students had to work through, some in teams, and upon which they received direct feedback. The teaching team was the same for all these courses.

#### *4.3. Data collection*

Performance, our dependent variable, was based on the average of the ratings of individuals in each team, which we obtained from university records. Aggregating values of such measures to the team level should only be done if there is sufficient agreement in the individual-level scores within the majority of teams. This can be assessed using the James et al. (1984 [55]) index of agreement. In the case of teams' individual performance ratings the median index of agreement was 0.997, which is well above the often-recommended cut-off point of 0.7 (James et al. [55]).

Information on the independent variables was collected by self-completion questionnaires distributed immediately after the teams had completed the project. The questions used are reported in Appendix 1. All individuals in the teams completed the questionnaires and the team scores for each of the measures were based on the average of the individuals' responses. These could all be aggregated to the team level as the median index of agreement varied from 0.67 to 0.99.

#### *4.4. Measures*

To measure XP practices, we used the Shodan survey measures of adherence to XP for *foundations*, *customer planning* and *craftsmanship*. We used the fourth core dimension of the Shodan measure, *teaming*, to measure the XP-specific team factor. In measuring each dimension separately, we are diverging from the approach of Williams et al. [33] who, while grouping their questionnaire items in the aforementioned four theoretical categories, treated them as measuring a unidimensional phenomenon: adherence to XP.

Our decision is justified by a factor analysis on the items, which found that a four-factor model fitted the data significantly better than a one-factor solution and that the four factors corresponded to the four separate dimensions of XP (this is reported in Michaelides et al. (2010 [56])). An 11-point scale from 0–100% (in 10% increments) was used for all the Shodan questions. We used the weights identified by the factor analysis to develop the measures of each dimension.

We used Campion et al.'s [41] measures of team inputs – the degree of participation, task variety, interdependence and heterogeneity of the team – and processes – potency, social support, workload sharing, and cooperation. The full details of these are in Appendix 1. Most questions included three items. For example, participation was measured by: “as a member of a team, I have a real say in how the team carries out its work”; “most members of my team get a chance to participate in decision making”; and “my team is designed to let everyone participate in decision making”.

For team cohesion, following Bollen and Hoyle's (1990 [57]) lead, we used a two-dimensional measure of the individual's perceived cohesion that determines the individual's sense of belonging and feelings of morale. We used Chin et al.'s (1999

[58]) survey questions, which they had adapted from Bollen and Hoyle's original questions (see Appendix 1).

#### *4.5 Analysis procedure*

We used the multiple linear regression model, the standard statistical technique for modelling and analyzing several variables, to test the competing hypotheses. In contrast to Pearson's correlation analysis which tests the association between two variables, for example performance and adherence to agile methods, regression analysis allows assessment of the relationship between one variable – the dependent variable – which in this case is performance, and a set of independent variables, these being XP methods and team characteristics. Regression assesses how the typical value of the dependent variable changes when any one of the independent variables is varied while the other independent variables are held fixed. Each model parameter (coefficient) for each independent variable is estimated to what it should be if all other independent variables were constant in the sample. For example, the model parameter for the independent variable cooperation is estimated from the data to what it would be if all teams had the same level of adherence to agile methods. This same level is set to the average adherence in the sample.

The significance of a regression model is gauged by using the F statistic, a measure of the model's goodness of fit, which tests whether a model with one or more independent variables is significantly better in explaining the variability of the dependent variable than a model without those independent variables. The same test can be used to contrast two models. For example, a model that includes a set of

interactions can be compared with one without the interactions to see whether it explains more variability in the dependent variable.

The significance of each of the model parameters is assessed using the t statistic, which tests whether a model coefficient is significantly different from 0. Although any estimate we obtain is unlikely to be exactly 0, the fact that these are obtained from a specific sample means that we need to assess the probability of obtaining the specific sample if these coefficients were in fact 0. This is done by dividing each coefficient by its standard error to calculate the t statistic which is then evaluated on the sample's t distribution according to the appropriate degrees of freedom.

In our case we are evaluating the strength of the relationship between three types of variables: XP practices, XP-specific team factor and general team factors. If for example all three types were found to be directly related to performance then this would support Hypothesis 1 that they all contribute to performance. If however only one of these types of factors is significantly associated with performance then the results offer support for one of the dominant factor models; for example if it is general team factor variables then Hypothesis 2 is supported but Hypotheses 3 and 4 are not confirmed. Further details of the specific tests for each set of hypotheses are given in the results section.

## **5. The results**

### *5.1. Descriptive statistics*

Descriptive statistics (mean and standard deviations) and correlations (Pearson product-moment) for the study's variables are displayed in Table 2. The majority of the correlations are significantly greater than zero. However, these are predominantly amongst the team variables, including XP-specific team factor (teaming). The two cohesion measures of belongingness and team morale are especially highly correlated (0.88, see Table 4, rows 10, 11), as they were in the original Bollen and Hoyle [57] study. Both are also highly correlated with workload sharing and social support. Of the XP practices' dimensions, foundations is not correlated with any other dimensions; craftsmanship is, however, correlated with the other two dimensions – customer planning and XP-specific team factor (teaming) – which are correlated with each other. Potency, cooperation, belongingness, team morale and teaming are significantly correlated with performance.

**[Insert Table 2]**

## *5.2. Testing Hypotheses 1–4: The additive and dominant models of agile performance*

We can test both the additive and dominant models of performance using a single analysis. The first stage of this established that there was no relationship between the majority of general team factors and performance; when we control for the other team characteristics, only cooperation is significantly related to performance, and of the two cohesion variables it is team morale not belongingness that is significantly related to performance. It further revealed that there was no need to control in the models for the age of the students, their degree type or the length of the project. The average age of the team members, whether the team was composed of undergraduate or

postgraduate students, and the length of the project were not significantly related to performance. Diversity in experience or skills was captured by the teams' own assessment through the heterogeneity measure.

We thus excluded insignificant variables and included all XP practice variables, XP-specific team factor and team cooperation and morale in our main analysis. Table 3 (columns 2–4) first reports a model just with the XP practices which shows that the XP-team factor is positively related to performance but foundations is negatively related to it. Model 2 (columns 5–7) included cooperation and morale and shows that cooperation is significantly related to performance but morale is not, and that XP-team factor is not longer significantly related to it, which suggests that cooperation may mediate the relationship between the XP-team factor and performance. In addition customer planning is positively related to performance in the second model which implies that cooperation is suppressing the effect of customer planning. By including cooperation in the equation we are adjusting the performance scores for cooperation, and in so doing revealing that customer planning is associated with performance, controlling for cooperation. Foundations remains negatively related to performance.

**[Insert Table 3 here]**

From these results we can conclude that the teams' adoption of XP practices and the way it functions are both associated with performance; no one factor dominates. Both the XP method and XP-specific team characteristics are important but of the general team factors, only cooperation is significant in our study. The negative association between foundations and performance, however, is not as predicted.

These results thus partially support Hypothesis 1, and hence Hypotheses 2, 3 and 4 can be rejected, as they show that the XP-specific team factor and at least one factor

from the other two types of variables, XP practices and general team factors, are significantly related to team performance. But the direction of the relationship between foundations and performance is the opposite of our hypothesis.

### *5.3. Testing Hypotheses 5–7: Multiplicative models of agile performance*

The second stage of our analysis to test Hypotheses 5 to 7 involved investigating the moderating effect of both types of team factors on the XP-practices–performance relationship. That is, to assess if the strength of the association between XP practices and performance is enhanced for example as team cohesion increases (as in Hypothesis 6). This is done by adding interaction terms to models used in stage one, such as the multiplication of team cohesion and the XP-specific team factor.

Tests for interactions with the significant variables from the analysis so far reveal no significant relationships. Thus, cohesion and cooperation do not strengthen the impact of customer planning or the XP-specific team factor on performance, nor do they reduce the negative relationship between foundations and performance. There is thus no support for Hypotheses 5, 6 or 7.

Further analysis of the interactions between the four dimensions of agility, however, revealed that the impact of foundations on performance varies with the degree of craftsmanship. The results, reported in Table 4, show that the interaction effect is significantly negatively related to performance. The implication is that as craftsmanship increases, the strength of the negative relationship between foundations and performance intensifies. Figure 1 displays this effect, showing that the best combination is low foundations and high craftsmanship. Teams practising, in Williams et al.'s [33] terms, craftsmanship, which are attempting to work at a

consistent sustainable pace and with simple design concepts, appear to find that following the foundations limits the results of their work more.

**[Insert Table 4 here]**

**[Insert figure 1 here]**

#### *5.4. Testing Hypotheses 8–9: Mediation models of agile performance*

Our final stage investigated whether the XP method drives the way the teams function and their level of cohesion, so that the XP–performance relationship is mediated by team factors as in Hypothesis 8, or whether the XP-specific team factor, mediated by team processes and cohesion, is the driver of performance as in Hypothesis 9. We used the standard procedure for testing mediation, outlined by Baron and Kenny (1986 [59], which entails ascertaining: first, that the mediated variable is related to the mediating variable, and this in turn is related to performance; and second, that the relationship between the mediated variable and the outcome variable is significant when the mediator is excluded in the variable but is either insignificant or its power considerably weaker when the mediator is in the model.

Our test for mediation reveals that team variables do not mediate the effects of XP practices on performance; that is, cooperation and the XP-specific team factor do not mediate the effect of customer planning or foundations on performance. Hypothesis 8 is thus not supported.

Team processes are not related to the XP-specific team factor so they cannot mediate the XP-specific team factor–performance relationship. However, it is mediated by team cooperation. Table 4 shows that the XP-specific team factor is

significantly associated with performance when cooperation is not in the model (see Table 5, block 1), and it is significantly associated with cooperation (Table 5, block 2) but it is no longer significantly associated with performance when cooperation is added to the model (Table 5, block 3). Cooperation thus mediates the relationship between the XP-specific team factor and performance. Hypothesis 9 is thus supported for one of the team process variables.

**[Insert Table 5 here]**

The results support Hypothesis 1 to the extent that examples of all three types of variables are uniquely associated with performance. Teamwork is important, but it is the XP-specific team factor, not general team factors, that is significant. Of the XP practices only customer planning is related to performance in the predicted positive direction. The negative relationship between foundations is not consistent with Hypothesis 1, whilst craftsmanship's only role is as a moderator of this relationship. In the case of team factors, only cooperation is related to performance. Whilst team factors do not mediate any relationship between the three XP practices, and hence Hypothesis 8 is not supported, the XP-specific team factor–performance relationship *is* mediated by a team factor – cooperation – so Hypothesis 9 has some support.

## **6. Discussion**

### *6. 1. Implications*

This study has shown that all XP practices (foundations, customer planning and craftsmanship) and the XP-specific team factor are significant for team performance. However, the XP practices influence performance in varying ways. Customer planning and the XP-specific team factor are significantly positively related to performance: the former relationship is direct, whereas the latter is mediated by cooperation. Foundations' relationship with performance is, however, negative. Craftsmanship plays a role by intensifying that relationship.

In addition to the XP factors, of the general team variables (task design, task interdependency, team composition, potency, social support, workload sharing, cooperation and morale) only team cooperation is related to performance. None of the large set of general team factors is important in explaining performance.

Moreover, cooperation's impact reflects its link to the XP-specific team factor. Its level is largely a reflection of the use of the XP practices and adoption of the core norms of collective working that constitute the XP-specific team factor, that appear to account for levels of cooperation in the team. Cooperation within the group may be seen as an emergent property of the group arising from the use of XP-specific team factors.

Overall the study indicates that XP factors do not emerge automatically from effective teamwork. Teams that stick to the core XP practices gain a competitive advantage even over teams that are cohesive and work well together. The lack of a role for the majority of the general team factors highlights the value of distinguishing these from the XP-specific team process. This, coupled with the fact that we have tested competing hypotheses, strengthens the conclusion that the agile methodology is important for explaining the effective performance of teams, and their level of cooperativeness, at least in this sample.

However, the negative relationship between foundations and performance reduces any claim that XP practices together represent what might be termed “the high performance software engineering method”. It appears that pursuit of some of the key processes involved in XP practices such as pair programming reduces the performance of the team as gauged by the product it produces. The negative relationship between foundations and performance could reflect how following some of the fundamental testing and pair programming practices of XP creates barriers to producing high quality products, at least in the time period that was allocated for the engineering. Our study only considers the quality at the point of delivery, but it may be that the real benefits of foundations are revealed in advantages for the user, and these only materialize after delivery. However, if no post-delivery benefits are forthcoming, then the implication is that these barriers have a stronger inhibiting effect on performance as craftsmanship increases. It would appear that in teams practicing craftsmanship, which are attempting to work at a consistent sustainable pace and with simple design concepts, following the foundations limits the results of their work more than those not doing this.

Our measures of XP practices can perhaps be treated as quantifying a continuum of low to high levels of adoption of the XP practices. If this is the case then the research can be interpreted as showing that (a) XP methods when fully adopted lead to superior performance over traditional methods and not just over more limited adoption of agile methods, and (b) the explanation for this is not reducible to their association with good teamwork. Perhaps though, one must be a little cautious in drawing this conclusion, not least as other researchers have argued that some planned-driven methods are compatible with agile methods (Boehm [29]; Dybå and Dingsøyr [21]; Dybå et al. [19]; Port and Bui, 2009 [60]); and more importantly, we do not know if

those teams that did not greatly embrace the XP methods were using the methods associated with traditional planned methods particularly well. Nonetheless, our more detailed observations of teams that scored low on the XP method showed that they worked in a way that was more akin to the planned approach than the XP method, as they typically delivered just a final working system without demonstrating iterative increments, and left the integration and testing of the system until late in the project.

According to the contextual or contingency (Conboy and Fitzgerald, pp. 6–7 [61]) approach to agility, it might be that teams adapted their use of agile methods to the context. It might be argued that the low adopters of agile methods had distinctive projects and they were adapting their use to their specific needs, and thus had we allowed for the type of project, the major contextual factor in this theory, the low adopters would not have underperformed. However, the variation in the projects was not in our view sufficient to expect an interaction between the degree of complexity and the XP measures to be significant, and certainly not of the cross-over type that would indicate that there is no main effect of XP principles. Other factors that have been mooted in the literature (e.g. Conboy [27]; Hoda et al. [31]; Stephens and Rosenberg [36]) as limiting the applicability of agile methods, such as the large size of a project, a big divide between the client, and the team being virtual, were not present in our study.

## *6.2. Strengths of the study*

Our focus on XP is, in our view, a major strength as (a) it is often quoted (perhaps with little evidence) as the most used agile method, and (b) to our knowledge it is the only method for which an adherence method (Shodan measures) has been developed.

The XP method was selected by the founder of the university software house largely because he found the continuous testing appealing, “offering a big advantage over other agile methods”. Also the nature of the work that the house would deal with was expected to be small projects where the clients might not have strongly clarified requirements for which agile was seen as especially appropriate.

Another strength of the study is that we did not rely on participants’ subjective assessment of their own performance, and thus we have avoided the problems of common method variance. The performance measure is based on the formal assessment of two people who initially made their judgements independently and take into account the client’s assessment of the project’s fitness for purpose and their own experience of the solution’s functionality. Their allowance for individuals’ contributions to the project is made on the basis of the manager’s observations over the length of the project. They are also able to allow for the complexity of the problem being addressed.

The measures of the independent variables, adherence to the XP method and teamwork have been developed and validated elsewhere. The measures of adherence to the XP method remain relatively novel, but their face validity seems strong, as they reflect the core elements of the method expounded by Beck [12], and Williams et al. [33] are one of the architects of it.

The measure with perhaps the strongest face validity is the teaming or XP-specific team factor and this has the strongest relationship with performance. This is heartening as it may suggest that investment in improving the other measures may yield stronger results, even if these results continue to show that there may be trade-offs or interactive effects between various dimensions. Future research could though consider whether Williams et al.’s [33] original dimensions could be improved.

A particular strength of the study relative to those that use students is that the projects were for real clients and had many of the characteristics of commercial software engineering. The university software house was deliberately established to conduct research studies such as this one, as it allowed the creation of homogeneous groups of students. In particular in this case all the students had received similar training in agile methods, had the same level of experience of working with agile methods, related to the manager in the same way and had the same opportunities to seek guidance on agile methods from him. They also had the same means of access to the clients.

The university location also gave us some advantages over industrial contexts, as we were able to control for factors that are not so readily controllable in other locations, and avoid compounding factors that exist in organizations, such as how the team relates to other teams, the hierarchical relations in the organization and the effects of career structures and aspirations on the team. The comparative studies reported by Dybå and Dingsøy [21] did not in fact control for these factors, nor for the fidelity to agile methods or differences in the degree of training or experience both within and between teams. Our study is also of co-located teams so we have none of the problems that are associated with virtual teams.

### *6.3. Weaknesses*

There are a number of potential weaknesses in our study. First, the performance measure is based on assessments that rely on subjective interpretations. We are not able to assess the inter-rater reliability of the assessments of the teams' performance, as the raw data on the independent assessments by individuals is not recorded. From

interviews with the raters, however, we discovered that divergences of 5% or above are extremely rare. We are also not able to validate the performance ratings through correlating them with the students' other exam results, as the only data available is their overall grade, and this is not independent of their project mark. Data on the students' own assessment of the project performance is also not available.

Alternative performance measures such as productivity measures based on lines of code would need to be weighted or complemented with data on the complexity of the project and demands of the client in order to standardize the scores or control for these. Measuring these would almost inevitably involve some level of subjectivity and thus would not necessarily be more reliable than direct subjective measures of performance or may have less face validity than these.

A second possible weakness is that we rely on team members to measure the independent variables, but the evidence of agreement on these gives some confidence in their validity and in some cases the level of measurement at the individual level is the right one for some of the concepts. For example, willingness of others to cooperate is a perceptual measure. Since measures based on external observation may be attempted they would rely on a judgement that the recipient of cooperation perceives it that way.

This is the first such study that has used Williams et al.'s [33] measures and they have fully represented all features of the agile method. For example, Michaelides et al. [56] show that the highest factor loadings on the foundations measure are concentrated on the testing items. Similarly, customer planning may have as much to do with planning in general as specifically with the customer. Finally, craftsmanship was primarily reflected in the question about sustainable pace, and simple design had only a small loading. This may be a factor behind the negative relationship between

foundations and performance and its moderation by craftsmanship. The testing interrupts the momentum of the groups' 'core' work and perhaps our developers were less enthusiastic about testing than others.

Third, omitted variables may play a role in the performance. Whilst Campion et al.'s [41] general team measures are based on a comprehensive literature review, other concepts such as team reflexivity might yet be significant, though they are likely to be correlated with some of the variables we found not to be important. We had no measures of the motivation of the students, its level or nature. Nor did we have measures that capture more of the client's relationship with the team and involvement in the project. These may be useful as they may help expose any potential tension between being highly client-focused and following other protocols to the letter. Moreover a key element of a team's approach towards their task might be the resolution of this tension. In a detailed comparison of three of the project teams in this study that were working on exactly the same problem, we in fact found that this was the key way in which their orientations differed. One team developed (through a mixture of explicit discussion and assumptions implicit in its actions) an approach which was highly client-focused and did not privilege all aspects of the XP method; another team, however, attempted to jointly optimize the product (client) and process (XP) criteria; the final one had a less integrated or focused approach (this case is reported in Wood et al. (2011) [62]).

Fourth, though the projects were not typical student assignments, the main limitation of the study is that it focuses on relatively inexperienced programmers. This may not, however, make them that distinct from engineers in industrial contexts, as Conboy and Fitzgerald (p. 14 [61]) suggest that the exposure of even experienced software engineers to agile methods is typically limited to one or two practices.

Finally, the small sample size of teams is a limitation and there are other constraints on our ability to generalize. However, with the exception of social support and potency, all insignificant general team factors have sufficiently small t-values to suggest that a larger sample would not reveal results that are likely to undermine our general conclusions. Nonetheless we have to be cautious about generalizing the findings to (a) other teams, (b) truly industrial contexts, and (c) agile methods other than XP. In particular we cannot be certain the results would be the same if the groups had been more diverse in terms of expertise. Such concerns about generalizability particularly apply to the negative foundation–performance relationship. Our engineers may have been faced with a trade-off between pursuing the foundations of XP and fulfilling all of the client’s requirements in the time they had available. It may be that this trade-off diminishes in importance as engineers become more adept at using XP methods, and hence that this result reflects the student or inexperienced nature of our sample. The intensity of the trade-off may also be less if deadlines are more fluid or engineers overrun delivery dates as may be possible in industrial contexts.

More research is required to validate our findings, especially to establish whether the negative impact of foundations and its apparent trade-off with craftsmanship is reproduced in non-student samples. A similar study using other agile methods and particularly Scrum (Schwaber and Beedle, 2002 [63]), which it may be argued is even more oriented towards promoting teamworking than is XP, would be particularly valuable.

## **7. Conclusion**

This study has contributed to two neglected areas in empirical software engineering: post-adoption performance and the role of teams in engineering methods and particularly innovative agile ones. It has shown that two key elements of the XP methodology – customer planning and its specific protocols for teamwork – are important to its success. The client and team foci of the XP method, which many take to be its distinguishing characteristics, are then its critical active ingredients (Highsmith and Cockburn [9]). Any success achieved by teams working with XP methods is not simply a reflection of their enhanced use of teamwork. In fact the team's cooperation is dependent on the use of XP-specific team protocols such as collective ownership and coding.

Adopting the view that theory develops in conjunction with empirical analysis, the implications for theory development are that it ought to focus particularly on the customer-focused and XP-team practices and the importance of cooperation. Further thought needs to be given to ways of conceptualizing XP practices and refining the Shodan measures (Williams et al. [33]), and more generally agile methods. Research could also extend beyond performance to other outcome variables such as the well-being and work–life balance of engineers and even users, and also to where and why agile methods are used.

As of now, the advice to practitioners would be that minimal use of XP methods or indiscriminate concentration of some of its elements is not likely to work. Rather, the more customer planning and team code management are adopted, the better the performance of the team in terms of producing a satisfactory user-friendly, high-quality solution for the client. Users of XP might consciously attend to avoiding potential tensions between adherence to its foundations and craftsmanship dimensions. The more detailed advice from Conboy and Fitzgerald [61] – for example

on the need to monitor whether staff members are being negligent or lazy in implementing agile methods – should also be heeded.

## **Acknowledgements**

The research reported in this paper was supported by a grant from the UK's Engineering and Physical Sciences Research Council (Reference: EP/D031516/1). We are grateful for comments on an earlier draft of this paper from Mike Holcombe, for assistance in the data collection from John Kahn, to Alison Geldart and Pat Hewson for their editorial assistance, and particularly to the students for their cooperation with the study.

## **References**

- [1] K. Beck, E. Gamma, C. Andes, *Extreme Programming Explained: Embrace Change*, second ed., Addison-Wesley Professional, Boston, MA, 2004.
- [2] B. Fitzgerald, G. Hartnett, K. Conboy, Customising agile methods to software practices at Intel Shannon, *European Journal of Information Systems* 15 (2) (2006) 200–213.
- [3] G. Mangalaraj, R. Mahapatra, S. Nerur, Acceptance of software process innovations: the case of extreme programming, *European Journal of Information Systems* 18 (2009) 344–354.

- [4] N. Turner, J. Barling, A. Zacharatos, Positive psychology at work, in: C.R. Snyder, S.J. Lopez (Eds.), *The Handbook of Positive Psychology*, Oxford University Press, Oxford, 2010, pp. 715–728.
- [5] S.T. Acuña, M. Gómez, N. Juristo, Towards understanding the relationship between team climate and software quality: a quasi-experimental study, *Empirical Software Engineering* 13 (2008) 401–434.
- [6] M.A. Campion, E.M. Papper, G.J. Medsker, Relations between team characteristics and effectiveness: a replication and extension, *Personnel Psychology* 49 (1996) 429–452.
- [7] S.G. Cohen, D.R. Bailey, What makes teams work: group effectiveness research from the shop floor to the executive suite, *Journal of Management* 23 (1997) 239–290.
- [8] S. Sawyer, Effects of intra-group conflict on packaged software development team performance, *Information Systems Journal* 11 (2001) 155–178.
- [9] J. Highsmith, A. Cockburn, Agile software development: the business of innovation, *Computer* 34 (2001) 120–127.
- [10] S. Nerur, R. Mahapatra, G. Mangalaraj, Challenges of migrating to agile methodologies, *Journal of Communications ACM* 48 (2005) 72–78.
- [11] J.H. Sharp, Global agile team configuration. *Conference on Information Systems Applied Research* 3 (2010) 1–13.
- [12] K. Beck, *Extreme Programming Explained: Embrace Change*, first ed., Addison-Wesley Professional, Boston, MA, 1999.
- [13] M. Holcombe, *Running an Agile Software Development Project*, Wiley, Hoboken, NJ, 2008.

- [14] S. Ilieva, P. Ivanov, E. Stefanova, Analyses of an agile methodology implementation, Proceedings of the 30th Euromicro Conference, IEEE Computer Society Press (2004) 326–333.
- [15] K. Mannaro, M. Melis, M. Marchesi, Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies, Proceedings of Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science, Springer, 3092 (2004) 166–174.
- [16] H. Svensson, M. Höst, Introducing an agile process in a software maintenance and evolution organization, in: Ninth European Conference on Software Maintenance and Reengineering (CSMR'05) 2005.
- [17] L.M. Maruping, X. Zhang, V. Venkatesh, Role of collective ownership and coding standards, European Journal of Information Systems 18 (2009) 355–371.
- [18] V. Balijepally, R. Mahapatra, S. Nerur, K. Price, Are two heads better than one for software development? The productivity paradox of pair programming, MIS Quarterly 33 (2009) 91–118.
- [19] T. Dybå, E. Arisholm, D. Sjøberg, J. Hannay, F. Shull, Are two heads better than one? On the effectiveness of pair-programming, IEEE Software 24 (6) (2007) 10–13.
- [20] J.E. Hannay, T. Dybå, E. Arisholm, D.I.K. Sjøberg, The effectiveness of pair programming: a meta-analysis, Information Software Technology 51 (2009) 1110–1122.
- [21] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: a systematic review, Information Software Technology 50 (2008) 833–859.

- [22] H. Robinson, H. Sharp, The characteristics of XP teams, in: Extreme programming and agile processes in software engineering, Lecture Notes in Computer Science, Springer Verlag, Berlin, 3092 (2004) pp. 139–147.
- [23] S.M. Young, H.M. Edwards, S. McDonald, J.B. Thompson, Personality characteristics in an XP team: a repertory grid study, Proceedings of Human and Social Factors of Software Engineering (HSSE), St. Louis, Missouri, ACM (May 16, 2005) 1–7.
- [24] D. Dalcher, O. Benediktsson, H. Thorbergsson, Development life cycle management: a multiproject experiment, Proceedings of the 12th International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05) 2005.
- [25] L. Layman, L. Williams, L. Cunningham, Exploring extreme programming in context: an industrial case study, Agile Development Conference, 2004.
- [26] F. Macias, M. Holcombe, M. Gheorghe, A formal experiment comparing extreme programming with traditional software construction, Proceedings of the Fourth Mexican International Conference on Computer Science (ENC 2003), 2003.
- [27] K. Conboy, Agility from first principles: reconstructing the concept of agility in information systems development, Information Systems Research 20 (3) (2009) 329–354.
- [28] W. Cunningham, Manifesto for agile software development, Available from <<http://agilemanifesto.org/>> Accessed 17 August 2012.
- [29] B. Boehm, Get ready for agile methods, with care, Computer 35 (1) (2002) 64–69.

- [30] A. Cockburn, J. Highsmith, Agile software development: the people factor, *Computer* 34 (2001) 131–133.
- [31] R. Hoda, P. Kruchten, J. Nobel, S. Marshall, Agility in context, in: M. Rinard (Ed.), *Proceedings of OOPSLA/SPASH'10*, October 17–21 (2010) 74–88.
- [32] K. Conboy, L. Morgan, Future research in agile systems development: applying open innovation principles within the agile organisation, *Information Software Technology* 53 (2011) 535–542.
- [33] L. Williams, W. Krebs, L. Layman, Extreme programming evaluation framework for object-oriented languages: version 1.4, North Carolina State University (NCSU) Technology Report, June 2004a, [ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc\\_anon/tech/2004/TR-2004-18.pdf](ftp://ftp.ncsu.edu/pub/unity/lockers/ftp/csc_anon/tech/2004/TR-2004-18.pdf). Accessed 17 August 2012.
- [34] L. Williams, W. Krebs, L. Layman, Extreme programming evaluation framework for object-oriented languages: version 1.3, NCSU Technology Report, 2004b.
- [35] W. Krebs, Turning the knobs: a coaching pattern for XP through agile metrics, Presented at Extreme Programming/Agile Universe, Chicago, IL (2002) 60–69.
- [36] M. Stephens, D. Rosenberg, *Extreme Programming Refactored: The Case Against XP*. APress, Berkeley, CA, 2003.
- [37] F. Karlsson, P. Ågerfalk, Exploring agile values in method configuration, *European Journal of Information Systems* 18 (2009) 300–316, doi: 10.1057/ejis.2009.20, published online 21 July 2009.
- [38] J. Iivari, J. Maansaari, The usage of systems development methods: are we stuck to old practices? *Information Software Technology* 49 (1998) 501–510.

- [39] J. Mathieu, M.T. Maynard, T. Rapp, L. Gilson, Team effectiveness 1997–2007: a review of recent advancements and a glimpse into the future, *Journal of Management* 34 (2008) 410–476.
- [40] J.E. McGrath, *Groups: Interaction and Performance*, Prentice Hall, Englewood, NJ, 1984.
- [41] M.A. Campion, G.J. Medsker, A.C. Higgs, Relations between work group characteristics and effectiveness: implications for designing effective work groups, *Personnel Psychology* 46 (1993) 823–850.
- [42] J.R. Hackman, G.R. Oldham, Development of the job diagnostic survey, *Applied Psychology* 60 (1975) 159–170.
- [43] P. Warr, *Work, Happiness and Unhappiness*, Lawrence Erlbaum, Mahwah, NJ, and London, 2007.
- [44] R.A. Guzzo, P.R. Yost, R.J. Campbell, G.P. Shea, Potency in groups: articulating a construct, *British Journal of Social Psychology* 32 (1993) 87–106.
- [45] A.V. Carron, Cohesiveness in sport groups: interpretations and considerations, *Sport Psychology* 7 (1982) 123–138.
- [46] E. Whitworth, R. Biddle, The social nature of agile teams, *Proceedings of AGILE 2007*, IEEE Computer Society, USA (2007) 26–36.
- [47] D.J. Beal, R. Cohen, M.J. Burke, C.L. McLendon, Cohesion and performance in groups: a meta-analysis clarification of construct relations, *Applied Psychology* 88 (2003) 989–1004.
- [48] S.S. Webber, L.M. Donahue, Impact of highly and less job-related diversity on work group cohesion and performance: a meta-analysis, *Journal of Management* 27 (2) (April 2001) 141–162.

- [49] M. Kilduff, R. Angelmar, A. Mehra, Top management team diversity and firm performance: examining the role of cognitions, *Organizational Science* 11 (1) (2000) 21–34.
- [50] S.-Y. Shin, W.-W. Park, Moderating effects of group cohesiveness in competency–performance relationships: a multi-level study, *Journal of Behavioral Studies in Business* 1 (2009), <http://www.aabri.com/jbsb.html>. Accessed 17 August 2012.
- [51] C.J.G. Gersick, Time and transition in work teams: toward a new model of group development, *Academic Management* 13 (1998) 9–41.
- [52] J. Lott, B.E. Lott, Group cohesiveness as interpersonal attraction: a review of relationships with antecedents and consequent variables, *Psychological Bulletin* 64 (4) October (1965) 259–309.
- [53] W. Richard Scott, *Institutions and Organizations*, second ed., Sage, Thousand Oaks, CA, 2001.
- [54] T.C. Chamberlin, The method of multiple working hypotheses, *Science* 15 (1890). Reprinted in *Science* 148 (1965) 754–759.
- [55] L.R. James, R.G. Demaree, G. Wolf, Estimating within-group interrater reliability with and without response bias, *Applied Psychology* 69 (1984) 85–98.
- [56] G. Michaelides, C. Thomson, S. Wood, Measuring fidelity to XP using the Shodan survey: a psychometric approach, *Empirical Software Engineering* 15 (6) (2010) 599–617.
- [57] K.A. Bollen, R.H. Hoyle, Perceived cohesion: a conceptual and empirical examination, *Journal of Social Forces* 69 (1990) 479–504.

- [58] W.W. Chin, W.M.D. Salisbury, A.W. Pearson, M.J. Stollak, Perceived cohesion in small groups: adapting and testing the perceived cohesion scale in a small-group setting, *Journal of Small Group Research* 30 (1999) 751–766.
- [59] R.M. Baron, D.A. Kenny, The moderator-mediator variable distinction in social psychological research: conceptual, strategic and statistical considerations, *Personal Social Psychology* 51 (6) (1986) 1173–1182.
- [60] D. Port, T. Bui, Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications, *European Journal of Information Systems* 18 (2009) 317–331.
- [61] K. Conboy, B. Fitzgerald, Method and developer characteristics for effective agile method tailoring: a study of XP expert opinion, *ACM Transactions Software Engineering Methods* 20 (2010) 1–30.
- [62] S. Wood, G. Michaelides, C. Thomson, Team approach, idea generation, conflict and performance, *Team Performance Management* 17 (7/8) (2011) 382–404.
- [63] K. Schwaber, M. Beedle, *Agile Software Development with Scrum*, Prentice Hall, Upper Saddle River, NJ, 2002.