An Analysis of the Implementation of Agile Software Development Practice in Irish Industry

Empirical research in a sample of Irish Industry

Trish O'Connell
Galway-Mayo Institute of Technology, Galway, Ireland.
trish.oconnell@gmit.ie
National University of Ireland, Galway, Ireland.
Department of Information Technology
trish.oconnell@nuigalway.ie

Abstract— **On reading the vast amount of literature that has been written on Agile methodologies such as Scrum and XP one is invariably faced with a number of guidelines associated with successful implementation. The Agile Manifesto has stated various values and principles that need to be inculcated into any Agile software development undertaking. We might then expect that software development organisations that self-report as being Agile are abiding by the recommended precepts of their chosen Agile method, be this Scrum or Extreme Programming (XP). This paper presents empirical research that was undertaken in a sample of Irish software development organisations with a view to determining if the Agile precepts were being followed by organisations that self-described their software development process as either Scrum or XP.**

Keywords-Agile; Scrum; XP; Agile guidelines

I. INTRODUCTION

"In the absence of any *a priori* knowledge, it is generally believed that if companies claim to be Agile then they are, in fact, following the precepts and guidelines of their chosen Agile methodology" [1]. In terms of the chosen approach/methodology this could refer to Scrum [2], eXtreme Programming [3], Crystal Clear [4] or indeed "any of a plethora of Agile practices" [1].

Agile software development takes an iterative approach to developing software where feedback from stakeholders is combined with a team based approach to deliver software artifacts, which are of genuine value to the Customer. The salient principle is recognition of the unpredictability of the software development process and an acknowledgement that to overcome this, a flexible, quick-response methodology must be employed.

It would appear that software development organisations have embraced Agile development wholeheartedly. According to the 7th Annual State of Agile Development Survey [5], which was compiled in 2013, more than 84% of those surveyed claimed their organisations were practicing Agile development. Agile has been in vogue since 2001 with the launch of the Agile manifesto [6]. Even non-technical Customers have heard about Agile. It appears to have become a 'must have' in the same way as the ISO9000 quality management standard was a de rigueur requirement for enterprises to do business in the nineties. But for all of the organisations claiming to be Agile one must wonder as to whether it has simply become a buzzword that is used to allay the trepidations of Customers. Buglione [7] states, "Agile is become one of most known and used ICT buzzwords of last 10 years." Perhaps software development organisations are using the buzzword but neglecting to adhere to the 'spirit' of the Agile manifesto.

Ambler & Lions argue for a more disciplined approach to mainstream agile practices such as Scrum and Extreme Programming (XP), stating that "mainstream agile methods don't provide enough guidance for typical enterprises" [8].

With this in mind it was decided to conduct "some quantitative research into aspects of actual Agile implementation in a sample of Irish software industry with a view to gaining an understanding of the level of compliance to documented Agile precepts" [1].

This paper seeks to ascertain whether organisations that lay claim to being Agile, do, in fact, adhere to the acknowledged precepts and guidelines of the chosen Agile method or whether the implementation is more of an *ad hoc* approach, as the author believes to be the case.

Section II of this paper examines the background to Agile. Section III takes an in-depth look at two of the foremost Agile methodologies in use, Scrum (which is really an agile project management approach to software development) and Extreme Programming (XP). Section IV of this paper outlines the research that was conducted including a breakdown of the study methodology and the participants. Section V presents the results of the research and lead into Section VI, where the findings are presented. Section VII presents a discussion of the results and leads into Section VIII, which examines the limitations of the research and plans for future work. Finally, Section IX presents the conclusions.

II. LITERATURE REVIEW

Since the advent of software development there have been many different models proposed to improve the development and delivery of software to the Customer. One of the major issues faced by organisations is that most often Customer requirements are not clear at the outset or are misunderstood by the designers. Added to this the fact that the requirements are rarely ‘set in stone’ means the designers are effectively trying to cope with constant flux or what Dooley [9] refers to as “software requirements churn.” Developing software is not an easy task.

Set within this context there are the many diverse models for software development; some, like the Waterfall model and its derivatives are referred to as the Traditional models. They treat software development as a linear sequence of prescribed phases, which unfortunately does not always reflect the reality of developing a software product or system. Similarly, the Unified Process is also an iterative and incremental approach to software development. However, in the last decade a new paradigm has emerged called Agile software development.

The term Agile was coined in 2001 and whilst its true meaning relates to the ability of the software development organisation to adapt to requirements churn, i.e., to be agile with regard to change, the term Agile is also used as an umbrella term to describe a range of software development methodologies, which use an iterative approach to developing software. Cockburn [4] states “Agile processes can take on late-changing requirements exactly because of early and frequent delivery of running software, use of iterative and timeboxing techniques, continual attention to architecture, and willingness to update the design.”

Whilst most know that Agile is an iterative approach to software development, which takes account of the changing requirements and lack of predictability that are inherent in a software project, the actual precepts of Agile might not be quite so well-known.

The traditional approach to software development was largely plan based insofar as a list of requirements were presented by the Customer at the outset and, once this was agreed to by the Developer, the Customer would normally sign off on a Requirements document. From then on, the development followed a sequential path in which discrete phases fed into each other, e.g., the requirements phase was followed by the architectural definition and so on until the product or system was delivered to the Customer.

However, requirements change and often what was delivered was not what was ultimately desired. For this reason it was acknowledged that there had to be a better way to develop software.

In 1970, Dr. Winston Royce presented a paper [10], in which he outlined the inherent risks in adopting a phased or sequential approach to software development. He argued that it was safer (in terms of minimizing risk) to use an iterative approach and also he suggested that it was “important to involve the Customer in a formal way so that he has committed himself at earlier points before final delivery” [10]. Paradoxically, the sequential approach to which Dr. Royce was referring has become what today is called the Waterfall method. It would appear that its proponents have missed the fact that Dr. Royce was using the model to describe what might be called, at best, a flawed model of software development.

In 2001, however, when seventeen key software developers (including Jim Highsmith, Martin Fowler, Mike Beedle, Ken Schwaber, Jeff Sutherland and Kent Beck to name but a few) met to discuss ‘lightweight’ development methods they agreed on what became referred to as the Agile Manifesto [6]. Thus, Agile promotes an adaptive approach to planning and an evolutionary, iterative approach to software development. Changes in requirements are both expected and managed and the Customer is seen as key to the successful outcome.

Unsurprisingly, each of the signatories of the Agile Manifesto had their own perception of what was agreed and consequently within the space of a few years there were “many different approaches to implementing Agile and each has its own ‘vanilla’ version” [1]. Sutherland [11] refers to “different approaches for implementing the core values from the Agile manifesto.”

Section III of this paper now examines the fundamentals of two of the most used Agile methods: Scrum and Extreme Programming (XP). Before proceeding, however, it must be acknowledged that Scrum and XP can be thought of as complementary Agile methods given that Scrum is a product development methodology whereas XP is an engineering methodology.

III. AGILE METHODS

As previously stated, are many different approaches to implementing Agile and each has its own ‘vanilla’ version. Sutherland [11] explains, “Each Agile methodology has a slightly different approach for implementing the core values from the Agile Manifesto, just as many computer languages manifest the core features of object-oriented programming in different ways.” The methodologies chosen for the study were Scrum and Extreme Programming (XP), since preliminary research into this domain identified these as the most prominent of the Agile methodologies currently in use for software development. Salo & Abrahamsson [12] refer to Scrum and XP as “perhaps best known agile methods.” Scrum will be treated first.

A. SCRUM

The first use of the method that would become known as Scrum was by Ken Schwaber in the 1990s at his company Advanced Development Methods. At approximately the same time Jeff Sutherland, who was working at Easel Corporation, is credited as being the first to refer to the approach as Scrum. In 1995, the two developers jointly presented their Scrum methodology at OOPSLA 1995 [2]. The developers’ contention was that software development’s “linear nature has been its largest problem” [2]. Schwaber argued that the Waterfall process “does not define how to respond to unexpected output from any of the intermediate processes” [2]. The Spiral model was similarly criticized for “each of the phases consisting of linear, explicitly defined processes” [2]. The solution to the non-linearity of the software process was to acknowledge the software development process as “complicated and complex” [2]. Accordingly, a method which would allow teams of developers to “operate adaptively within a complex environment using imprecise processes” [2] was required.

The Scrum process is essentially an evolutionary, incremental framework. It is a team based software development approach, which uses a time-boxed adaptive artifact termed a Sprint. According to Millett et al. [13], Scrum takes an "iterative approach to software development." The Scrum process for software development is depicted graphically in Figure 1. It shows the key elements of the Scrum methodology much of which is referenced in this section.

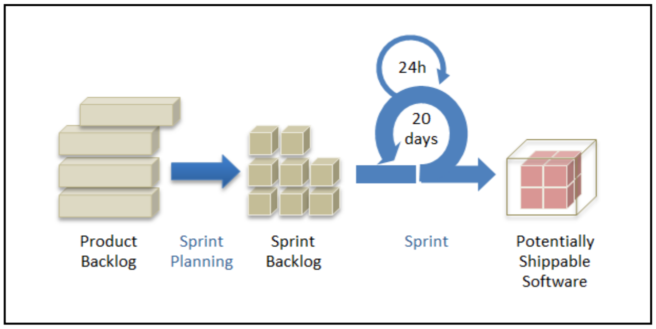

Figure 1. The Scrum Process

In Scrum, the team is responsive to its environment throughout the development. The developers are accorded unlimited flexibility and creativity during the development iterations. Knowledge is transferred among the team during the development and Schwaber [2] estimated that the probability of success using this approach would be high.

In accordance with its origins (the term Scrum comes from the sport of Rugby in which a group of players work together to move the ball up the field and over the try line to score points and win the game from the opposing team) it should be clarified that in addition to the various activities of the Scrum development process there are three roles, which help the team achieve success.

The Scrum Master is responsible for the team process, helping the team to achieve success and use Scrum correctly.

The team consists of cross-functional developers who between them possess all off the expertise necessary to deliver a potentially shippable increment of the product.

The Product Owner is a key role in Scrum as it is this individual who supplies the requirements that will comprise the product or system. Effectively a Scrum team works for a fixed duration (known as a Sprint) on product requirements. which are initially "contained in an ordered list known as the Product Backlog" [2]. At the beginning of each Sprint, the requirements are prioritized into a list known as the Sprint Backlog with the aim of completing an agreed set of deliverables by the end of the Sprint. Deemer et al. [14], explain further, "During the Sprint, the chosen items do not change. Every day the team gathers briefly to inspect its progress, and adjust the next steps needed to complete the work remaining. At the end of the Sprint, the team reviews the Sprint with stakeholders, and demonstrates what it has built. The development team obtains valuable feedback that can be incorporated in the next Sprint. Scrum emphasizes working product at the end of the Sprint that is really "done"; in the case of software, this means code that is integrated, fully tested and potentially shippable."

To add to this brief précis of Scrum this paper will now focus on some of the specific aspects that relate to the implementation of Agile Scrum. Barari [15] advises that "it is important to follow the guidelines defined in Scrum but the ultimate goal is to deliver what you promised." With regard to the guidelines, Schatz & Abdelschafi [16] state quite categorically that "there aren't many rules in Scrum but you need to adhere to the ones that (do) exist." These will be examined in the next section.

1) ASPECTS OF SCRUM SOFTWARE DEVELOPMENT

As mentioned above the core precepts for the implementation of Scrum are taken from Schwaber & Sutherland ([2][11]). It may be said that "the rules of transitioning software development from a plan-driven approach to an Agile approach are not set in stone" [1]. This is largely based on the fact that due to perceptual filters no two individuals will likely have the same interpretation of an agreed principle. However, there is much commonality attached to the writings on Scrum [11][17][18][19]; consequently, the next section will examine the activities of the Scrum development process as it is addressed by the authors listed above.

a) PRODUCT OWNER

There is complete unanimity on the requirement for the Product Owner role in Scrum. As described above the Product Owner has a key responsibility in the development process to supply the product requirements. According to Deemer et al. [14], "The Product Owner is responsible for maximizing return on investment (ROI) by identifying product features, translating these into a prioritized list, deciding which should be at the top of the list for the next Sprint, and continually re-prioritizing and refining the list. The Product Owner has profit and loss responsibility for the product, assuming it is a commercial product. In the case of an internal application, the Product Owner is not responsible for ROI in the sense of a commercial product (that will generate revenue), but they are still responsible for maximizing ROI in the sense of choosing – in each Sprint – the highest-business-value lowest-cost items."

With traditional methodologies the role of the developer was to elicit the Customers' needs in the form of the Requirements document. It was assumed, firstly, that the Customer knew exactly what he/she wanted and secondly, the Customer was able to document his/her needs in sufficient detail that the developer(s) would be clear on what was required. In contrast, Rico et al. [20] suggest that Agile advocates "listening to and interacting with Customers to ascertain their needs." This changes the requirements gathering from that of elicitation to a dialogue in which the role of the Product Owner is to interact with the Customer, to effectively be client-facing. Beyer [21] sees the Product Owner as "the Customer representative" and outlines his responsibility to "find out what the stakeholders and end users actually need" [21]. According to Schwaber [2], the product Owner is "responsible for representing the interests of everyone with a stake in the project and its resulting system." Stober & Hansmann [18] define a Product owner who "represents the stakeholders, such as Customers." The Product Owner, then, effectively represents what Hauser & Clausing [22] refer to as the "Voice of the Customer." Pichler [19] states, "The Product owner must develop an intimate understanding of Customer and user needs, and how these needs can best be met." He suggests that the process for achieving this "is to involve Customers and users early and continuously in the development process" [19]. Furthermore, Pichler [19] recommends "asking Customers to provide feedback on prototypes, inviting Customer representatives to sprint review meetings and releasing software early and frequently are great ways to learn from Customers." Royce [10] also advises "it is important to involve the Customer in a formal way so that he has committed himself at earlier points before final delivery."

Whilst this is all encouraging there appears to not be a prescribed methodology to further these aspirations. Consequently, it is left to the software development organisation to establish the link between the Product Owner and the Customer. It would, however, appear that the requirements elicitation activity performed by the Product Owner is crucial to successful software development with Scrum.

In addition to the Product Owner, Scrum is very clear on the importance of involving the Customer. This is because lack of lack of user involvement is a primary cause of project failure. The CHAOS report of 2010 [23] stated: "projects that lack user involvement perform poorly."

b) CUSTOMER INVOLVEMENT

According to the published history of the Agile manifesto [6], the assembled group of developers espoused "a set of compatible values, a set of values based on trust and respect for each other and promoting organisational models based on people, collaboration, and building the types of organisational communities in which we would want to work" [6].

The proponents of Scrum, including Cobb [24], advocate "as much Customer collaboration as possible" but he counsels that the "Product Owner represents the voice of the Customer and is expected to provide overall direction to guide the project toward producing the value to satisfy Customer needs" [24]. This should most likely involve close collaboration with Customers and stakeholders.

Rico et al. [20] answer the question "How is Customer collaboration performed in agile methods?" They suggest the answer is "With right-sized, just-enough, and just-in-time interaction"[20]. With Scrum, Customer needs are captured in the form of epics and user-stories which form features within a product backlog. After some features have been implemented, Customer collaboration takes place after approximately 30 days in what is known as a Sprint review.

It would appear obvious that Customer involvement is crucial. Thus it is apposite to state that Customer involvement is an essential prerequisite to a successful software development process. However, it should be noted that the Product Owner and the Customer are but a part of the Scrum methodology. Without a team of cross-functional individuals to realize the Customer's vision the project would not get off the ground.

c) TEAM ORGANISATION

Traditionally teams were formed by managers in the organisation who appointed team leads and assigned personnel to the various roles that were required to develop a software product or system. Often teams failed to get results due to inappropriate team leadership, interpersonal dynamics or unclear objectives.

The Agile approach, according to Cooke [25] is to "rely on the mutual trust (and dependency) that emerges between stakeholders and delivery team members: delivery teams depend upon the expertise of stakeholders to accurately communicate and prioritize the business requirements; and stakeholders equally depend upon the expertise of the delivery team members to regularly produce outcomes that meet these requirements." It is this co-dependency between the team members and also between the team and its' Customer that makes Agile so powerful.

In order to achieve success, the Agile team must trust and depend on its members to perform their tasks to the best of their ability. Having this self-reliant and inward focus is undoubtedly a core strength of an Agile team.

Cooke [25] explains further, "Stakeholders are responsible for guiding the business priorities and for measuring the outcomes of each iteration, but they are not the people who determine the volume of work that can be achieved in that short time-frame. Instead, stakeholders defer to the multi-skilled delivery team to advise them on the actual work required to achieve their objectives, the estimated time for each task and what the delivery team can realistically achieve in an iteration given their current workload and other commitments."

Tata and Prasad [26] would appear to reinforce the need for self-organizing teams by explaining that "Self-management can indirectly increase team effectiveness by increasing team members' sense of responsibility and ownership of work."

d) SOFTWARE RELEASE

An important activity that is frequently mentioned by authors on Scrum is the activity of releasing software to Customers often and early. This is what Koch [27] refers to as "continuously and often". He cites the rationale behind this as to "increase (the) motivation for all participants, allow for easier discussion of the current status and therefore increase chances to uncover necessary changes and efficient possibilities for incorporating them."

Pichler [19] concurs, recommending "asking Customers to provide feedback on prototypes, inviting Customer representatives to sprint review meetings and releasing software early and frequently are great ways to learn from Customers."

Thus, it would appear that there is evidence for this precept to be included in order to achieve a successful software development using Scrum.

e) SETTING TEAM PRIORITIES

According to Rawsthorne [28], the Product Owner (he refers to the simplest version of the Product Owner role as the Business Owner) "sits between the Stakeholders and the team." He cites the function of the Product Owner as being to "prioritize/order the work that the stakeholders want into a single Value Backlog." Furthermore, he explains the Product Owner "moves the items from the Stakeholder's Value Backlog to the Teams Work Backlog at a rate that will not overload the team" [28]. Khalil et al. [29], state, "in Scrum a single person must have final authority representing the Customer's interest in backlog prioritization and requirements questions. This person must be available (to the team) at any time to during the Sprint planning meeting and the Sprint review meeting." The setting of the team's priorities is a crucial part of the activity of the Scrum software development process.

Having outlined the guiding principles in Scrum the same approach will be taken with Extreme Programming (XP).

B. EXTREME PROGRAMMING (XP)

Extreme Programming (XP) - was introduced in 1999 by Kent Beck [3]. Rico et al. [20], describe that in its earliest incarnation XP featured "just-in-time evolution, self-chosen tasks, aggressiveness, model-driven development, and communications." Over the last decade, however it has evolved to incorporate practices which include "onsite Customers, pair programming, test-driven development, and open workspaces." Williams & Kessler [30] explain that "XP is a minimalist approach so it is essential that many of the practices actually get done." With that in mind, the fundamental aspects of XP (as depicted in Figure 2) will now be examined.

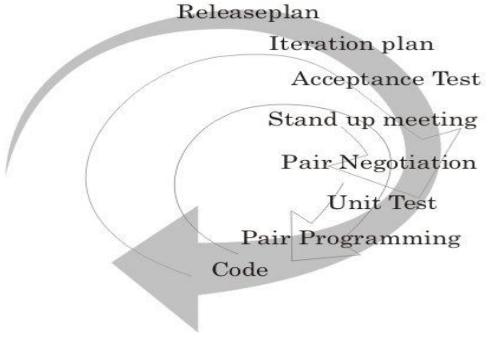

Figure 2. Planning and feedback loops in XP

2) ASPECTS OF XP SOFTWARE DEVELOPMENT

Both Scrum and XP are iterative albeit that the Sprint in Scrum would typically be slightly longer; a typical Sprint may last from two to four weeks whereas generally the iterations in XP would be from one to two weeks. In the same way as Scrum has its own defining aspects that are clearly identifiable as Scrum viz. Daily Scrum, Sprint Planning, Sprint Review, etc. so too, does XP have its own precepts. However, Kniberg & Skarin [31], state that "Scrum is less prescriptive than XP."

In XP, the role of the Product Owner is assumed by the Customer. The concept of a self-organizing team is still valid but whereas in Scrum there are no prescribed engineering practices, in XP there are some definitive requirements. These fundamental aspects of XP software development activities will now be examined in more detail.

a) CUSTOMER ON-SITE

Within Extreme Programming, Customer needs are captured in the form of user stories. The Customer is actually a full-time member of the project and communicates with the developers throughout the project." Cooke [25] concurs, "The most effective way to ensure ongoing business value is to *directly involve* key internal and external stakeholders in the process. In theory, representative stakeholders participate as *active members* of the Agile team during the process, providing the team with real-time input and hands-on feedback at two key points in the process:

- At the start of each iteration to describe and prioritise their business requirements
- At the end of each iteration to review and assess outputs against their stated requirements."

Ideally, stakeholders need to be on hand or perhaps, more importantly, they need to be available to the team in order to respond to developers' questions and review work as it is being completed. Cooke [25] opines, "The more available stakeholders are to the Agile team throughout the process, the closer that each deliverable will be to meeting the true needs of the organisation."

In XP, the Customer is actually part of the development team. When analyzing the role of the Customer it is interesting to refer to Beck's [3] intention that a "real Customer must sit with the team, available to answer questions, resolve disputes, and set small-scale priorities" … "someone who will really use the system when it is in production." Beck [3] would seem to advocate the presence of the Customer as a form of immediate feedback to the developer effectively being, according to Martin et al. [32] "someone who steps up and takes responsibility for the requirements." Furthermore, Martin et al. [32] argue that in XP the Customers "are charged with delivering what every developer wants: clear requirements, declared outcomes, and a helping hand with the messy world outside. XP makes development simpler by assigning some of the most slippery tasks to the Customer."

Thus, the Customer is critical to the success of delivering Agile software using the Extreme Programming (XP) methodology.

Stober & Hansmann [18] would appear to concur, stating “the project team always needs to keep in close contact with the Customer to ensure that the project is meeting the Customer's expectations at any given time.”

b) PAIR PROGRAMMING

Cockburn [4] classifies XP as a “high discipline” process since there are various prescribed practices outlined.

One of the key activities most often associated with XP is that of Pair Programming. This is where two developers work together at one workstation. Williams & Kessler [30] describe pair programming as “an integral part of XP.”

Quite apart from the obvious benefit of two problem solvers working on the same issue and discussing optimal solutions there is the added bonus of continuous code review. Williams & Kessler [30] caution that “it is dangerous to do XP without pair programming.”

c) SELF-ORGANIZING TEAM

Shore [33] explains that XP teams are self-organizing and cross-functional. Moe et al. [34] use the label "self-organizing" teams as a synonym for "autonomous teams" and for "empowered teams". They refer to the work of Guzzo & Dickson [35] and explain such teams as “teams of employees who typically perform highly related or interdependent jobs, who are identified and identifiable as a social unit in an organisation, and who are given significant authority and responsibility for many aspects of their work, such as planning, scheduling, assigning tasks to members, and making decisions with economic consequences.”

d) SETTING PRIORITIES

In XP it is the Customer who prioritizes the work to be done. Griffin [36] explains, “Customer ability to determine what developers will work on next greatly increases the Customer’s sense of confidence in their development unit, and sense of being listened to. The Customers began to feel for the first time that they were truly in control, and that they would receive what they needed when they needed it.”

e) OPEN PLAN WORKSPACE

Robinson & Sharp [37] explain that an open plan workspace, which is often synonymous with XP software development is “symbolic of the culture. The physical setting is open plan: open and public to all in the team.” The obvious rationale behind this practice is that it fosters good communication between the team. Recht & Nielson [38] state, “Communication is the basis of XP. Any problem occurring can invariably be traced back to lack of communication either between the developers or between the developers and the Customer. As such, it is important never to let communication become a secondary priority.”

Having examined both Scrum and XP and highlighted their various key activities, practices & roles and the rationale behind them the next stage is to describe the research that was conducted to ascertain the level of compliance to these clear and unambiguous Agile precepts.

The next section presents the research method adopted in addition to the research vehicles and participants used in the study.

IV. THE RESEARCH

Countless academic papers, textbooks and instruction manuals have been written describing the various Agile approaches/techniques but, to date, to the best of this author’s knowledge, with the exception of research conducted by Salo & Abrahamsson[12] on the use and usefulness of XP and Scrum in European embedded software development organisations no one has sought to identify whether the existing paradigms, in their implementation, do, in actuality, follow the precepts as laid down by the proponents of these self-same methodologies.

Specifically, this refers to whether software development organisations who self-report as using either Scrum or XP are really implementing the method as intended or indeed whether (as is hypothesized by the author) the organisations are adopting an *ad hoc* approach to implementation, a ‘pick-and-mix’ approach, in effect, where those precepts that suit the organisation are adopted and those that would require a fundamental shift in the organisation’s culture are sidelined.

Basically, the author wished to understand the level to which management/developers follow or are allowed to follow the process. An integral part of the research was to identify whether those managers and developers who admit to working in Agile software development organisations actually believed their organisations to be Agile (the view from the coal-face, one might say).

Consequently, the purpose of the research was to firstly clarify which methods were being used for software development and then to identify the perceptions of those that worked in these self-reported Agile organisations. In addition, the author wished to establish empirically whether the precepts of the practiced Agile methodology were being inculcated into the actual development process as has been advocated by the pioneers of the methods.

It was believed that the answers to these research questions would add significantly to the body of knowledge regarding Agile implementation in Irish software development organisations.

A. Research Method

The research effort in this case was centered on conducting a quantitative study that would be descriptive in nature. Leedy & Ormrod [39] describe this type of research as “identifying the characteristics or exploring possible correlations among two or more phenomena.” The authors also state “descriptive research examines a situation as it is” [35].

The chosen method that was used was that of the survey, which, according to Leedy & Ormrod [39], "involves acquiring information about one or more groups of people by asking them questions and tabulating their answers" Leedy & Ormrod [39] indicate that "the ultimate goal is to learn about a large population by surveying a sample of that population." Due to time constraints and logistics it was decided to use an online survey. However, with a view to gaining a deeper insight interviews were also conducted when it was felt a response required clarification or more detail was sought.

In an ideal scenario, it would have been preferable to obtain a totally random selection of employees in Irish software development companies to answer the research questions. However, there was a concern that if the response rate was low (which is one of the main drawbacks of this research method, what Leedy & Ormrod [39] refer to as "low return rate") then the research may have been over before it began. Consequently, it was decided to adopt a degree of 'selective sampling'. This is what Nardi [40] refers to as "purposive sampling." This involved including specific pre-defined groups in the sampling frame with a view to increasing the likelihood of collecting "data on organisations that had some prior knowledge of Agile practices, as opposed to taking a completely random sample, which may have resulted in confused responses. [1]"

In addition to personal contacts it was decided to 'acquire' a list of /access to software development companies from software groups such as AgileIreland, Information Technology Association Galway (ITAG), the Irish Software Association (ISA), the Irish Software Innovation Network (ISIN) training companies, blogs etc. and also from colleagues/past students of the NUI Galway MScSED course who have contact with the software development industry. A very brief description of the various software support groups is listed in Section IV. C, which deals with the survey participants.

B. *Research vehicles*

Whilst the primary focus of the research was on organisations that use either Scrum or XP it was decided to use the research to gather as much data as possible.

With this in mind, it was decided to stratify the research into two focal domains. First there was the Agile management domain which would be comprised of those involved in the management of Scrum/XP development projects. Specifically roles such as Software Development Manager, Project Manager, Test Manager, Scrum Master, Release Manager etc. were expected to feature in this category.

The second domain was that of those who were more 'hands-on' in the software development process. It was hoped to get information from those who would be involved in the software development team.

In order to make the research manageable it was decided to create two surveys which would be made available to the participants who would then be free to choose the one most applicable to their role in software development. Clear and unambiguous instructions accompanied links to the surveys both informing would-be participants of the research and also guiding them to choose the most appropriate vehicle for them to access. Having identified the research questions the survey was created by the author and prior to release the survey was validated by a cross-functional set of academics and software industry representatives. When it was felt that the surveys as developed were capable of generating clear, unambiguous data they were hosted online by SurveyMonkey.

For ease of reference the first grouping was labeled the Management survey and the second domain was referred to as the Developer/Team survey.

To a large extent both surveys contained many of the same questions although the actual detail of the team working practices were omitted from the Management survey. The next section looks at the research participants.

C. *Participants*

As previously stated purposive sampling was used to target would-be participants due to the requirement to ensure that the data obtained would come from organisations that were conversant with Agile software development. Research into the Irish software development community suggested the following be included:

1) *AgileIreland*

AgileIreland is an Irish web community. According to its mission statement AgileIreland is "a community site for anyone interested in agile and lean methods of software development throughout the 32 counties of Ireland." Unfortunately, there is no information on the site about the number of members who might potentially access the surveys. The AgileIreland discussion boards were used to launch the online survey on their site.

2) *Information Technology Association Galway (ITAG)*

A large number of local software development companies are members of ITAG, which is the Information Technology Association, Galway. According to their web Home page ITAG "was established in 2000 by a group of forward looking IT professionals representing both multinational and indigenous IT companies. Our goal is to foster the continued growth of a strong IT cluster in the Galway region, through networking and training events, joint initiatives, and regional and national advocacy." The author contacted ITAG with a view to having the online survey link circulated within the software development industry in the West of Ireland.

3) *The Irish Software Association (ISA)*

The Irish Software Association (ISA) is a part of IBEC (the Irish Business & Employers' Confederation). Their web presence states that their "membership base is comprised of over 160 companies actively involved in every area of the software sector in Ireland." They have a strong Internet base and host the ISA LinkedIn pages. The LinkedIn membership is quoted at 1,416 and it was to this potential audience that the author publicized the online survey.

Obviously, with annual vacations etc., it was not realistic that the author's online survey would be accessed by the entire membership, but the author was hopeful that a fraction would be sufficiently cooperative to acquiesce to completing the surveys.

4) *Personal Contacts/Tutors/Past MScSED students*

It was likely that much of the data collected from the survey would be obtained from the author's personal contacts, as it is an established fact that people with whom one has an existing connection are likely to be more responsive to requests for information.

Based on all of the various channels described above the author was confident that little more could be done to solicit responses to the research questions. However, whilst the potential audience for the research was estimated to be in the region of 2,000 individuals the author was well aware that only a small fraction would take the time and trouble to respond.

With a view to capturing a fully representative view, cross-functional participants, including both Agile team members and software development management in organisations that self-reported as using Scrum and XP, were targeted. In this way it was hoped that the findings would be representative of the actual state of play of software development in Irish industry.

The complete breakdown of Scrum management participants is shown in Table I.

TABLE I. MANAGEMENT SURVEY PARTICIPANTS

	Organisation Size		
Role	***1 to 50***	***51 to 500***	***500+***
S/W Dev. Mgr.	3	3	4
Project Mgr.	2	2	4
Q.A. Mgr.			1
Test Mgr.		1	

Similarly, the breakdown of Scrum team participants is shown in Table II.

TABLE II. SCRUM TEAM SURVEY PARTICIPANTS

	Organisation Size		
Role	***1 to 50***	***51 to 500***	***500+***
Designer	1	1	1
Senior Developer	2	3	1
Developer	2	4	3
Test Engineer	2	3	2

Given the fact that the survey was online, it was not possible to compute a response rate, *per se*. However, given the purposive sampling involved and the profile of a number of the organisations represented (e.g., Cisco, Globoforce, Avaya, CSG International, etc.) it was felt that a sufficiently representative number of respondents to the Scrum questions had contributed to make the results relevant.

Unfortunately, the level of XP response was extremely disappointing. In total, only three responses who reported using XP were received. However, the data that was collected will, nevertheless, be included though one would have to refrain from making any pronouncement based on limited statistical data.

V. RESULTS

As explained previously, at the commencement of the research activity the author sought to classify the respondents to enable a proper context to be applied to the research questions. Participants had access to both a Management and a Developer/Team survey. Clear instructions were provided and the respondents were then invited to select the survey that best described their role in the Agile organisation within which they worked.

A. *Management Results*

In order to provide a degree of clarity this section commences with an introduction to the profile of the respondents surveyed. This includes organisational size and sector in addition to the respondent's role in the organisation. The subsequent section presents the Scrum results and finally the XP results are presented.

1) *Respondent Characteristics*

It can be stated that those who completed the Management targeted survey encompassed organisations of varying size, i.e., 28% from organisations with fewer than 100 software personnel, 41% from organisations with between 101 and 500 personnel involved in software development and 31% coming from organisations with more than 501 personnel involved in software development activities. All organisations were based in Ireland with a preponderance coming from the west coast of Ireland. This is shown graphically in Figure 3.

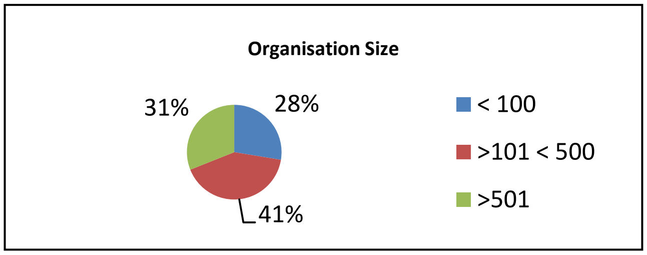

Figure 3. Organisation size of management survey respondents.

The next classifier was the nature of the respondents' organisation. As part of the survey validation process it was decided to include the organisation sector to see if there were any discernible patterns in the data. The overwhelming category represented is in the sector of telecommunications/unified communications with a 55% representation. 21% represents software products ranging from reward & recognition software to security. The overall breakdown is shown in Figure 4, below.

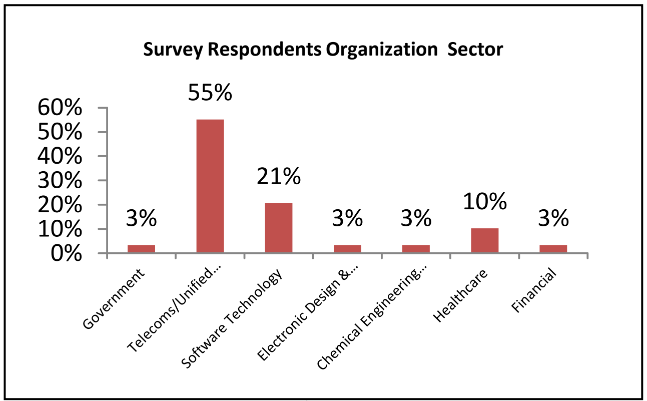

Figure 4. Organisation sector of management survey respondents.

The final part of the respondents' classification focused on their role within the software development process. Note that of the two surveys available to respondents this particular survey was targeted only at the organisational /managerial level of the software development activity. Consequently, the author would expect to only see responses from personnel involved in the management and deployment of the software development activity in the organisation in which they operate. In a software development organisation this would most likely encompass a broad spectrum from Software Development Manager to Project Manager, Scrum Manager, QA Manager, Test Manager, etc. as shown in Figure 5.

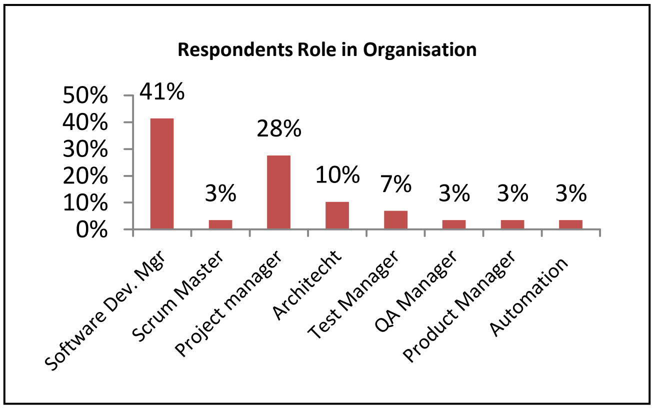

Figure 5. Management survey respondents roles in organisation.

The range of roles selected by the respondents validates that the author did, indeed, capture views of those individuals involved in the management of the organisations' software development activities.

In terms of the type of Agile method that predominates in the organisations that admit to having an Agile software development process the overwhelming method used is Scrum with 83% of all respondents selecting it. XP features in a minor sense (17%) and other techniques such as Crystal Clear do not feature whatsoever in this survey although they were offered in the survey as a possible option. The results are displayed in Figure 6.

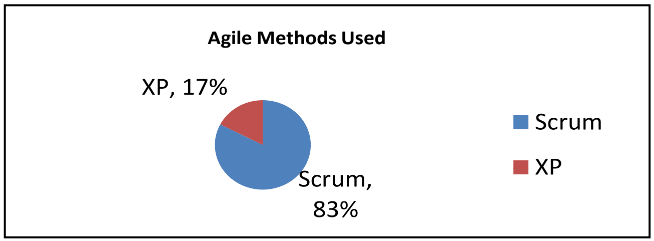

Figure 6. Agile methods used in respondents organisation.

a) Scrum Management Respondents Results

Given the self-described adoption of Agile one might expect that the survey respondents would perceive their organisation to, in fact, be operating in a fully Agile manner. This formed the basis for the next question which asked whether the respondent considered his/her organisation to be truly agile in the context of handling and managing software churn. The results are presented in Figure 7.

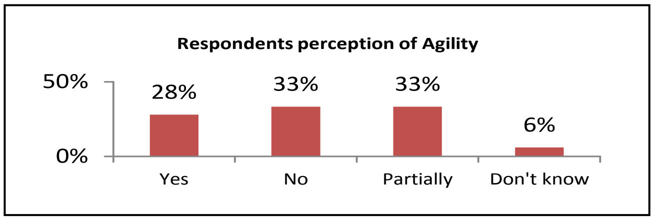

Figure 7. Management respondents perception of organisation's agility.

A number of respondents confirmed these views in subsequent interviews admitting that whilst the perception by many (senior management, Customers, etc.) is that they are committed to Agile, the organisation for which they work is, in fact, only paying "lip service" (Respondent #17) to being Agile. There appears to be a degree of frustration that respondents are not being enabled to implement the techniques correctly and this is due in part to pressure to "keep the product gates open."(Respondent #4)

In addition to assessing agility, it was felt to be important to determine if survey respondents were satisfied with their existing Agile software development process. The overall answer to this question from the Management perspective is displayed in Figure 8.

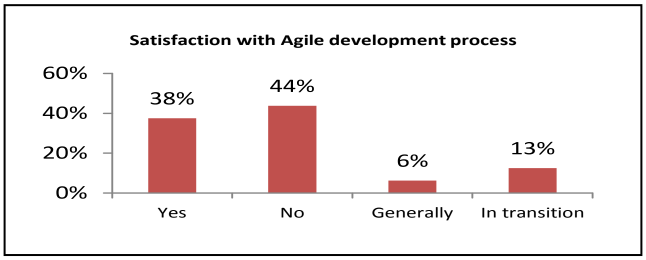

Figure 8. Management survey respondents satisfaction with Agile process.

To conclude this section the author sought to clarify the level of Customer involvement in the development process as perceived by the respondent who is in an Agile management role as opposed to an operational role. The question asked whether the respondent's organisation actively encouraged Customer involvement in the software development process. The answer, displayed below as Figure 9 was an overwhelming Yes at 86%.

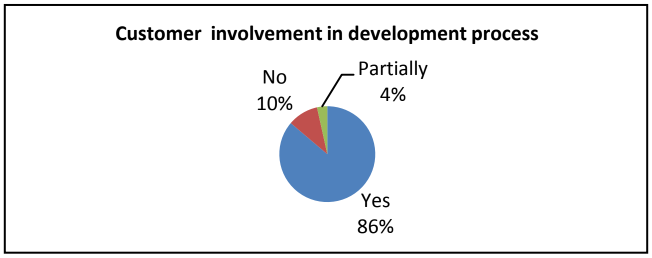

Figure 9. Management respondents perception of customer involvement in software development process.

Of those that responded No it is interesting to note that they represent organisations who claim (self-describe) to be Agile. The respondent who acknowledged that Customers were "partially encouraged" (Respondent #8) was also employed in a self-described Agile organisation.

With a view to affirming the actuality of Customer involvement the respondents were asked whether software was released/demonstrated early to Customers. It would be logical to expect that Customer involvement and attendance at Sprints to see working software would yield similar results. The responses are displayed in Figure 10 below.

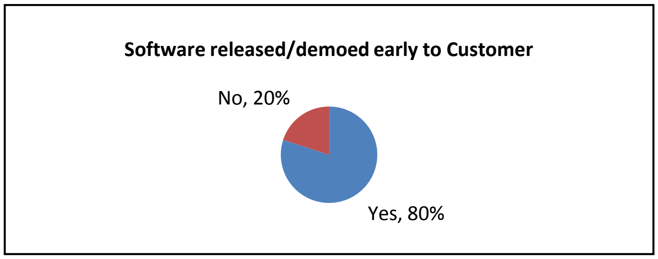

Figure 10. Management respondents knowledge of early software release.

The final question related to how management believed their Customers perceived their organisations' Agile development process.. Somewhat surprisingly there was not an overwhelming majority as can be seen in Figure 11.

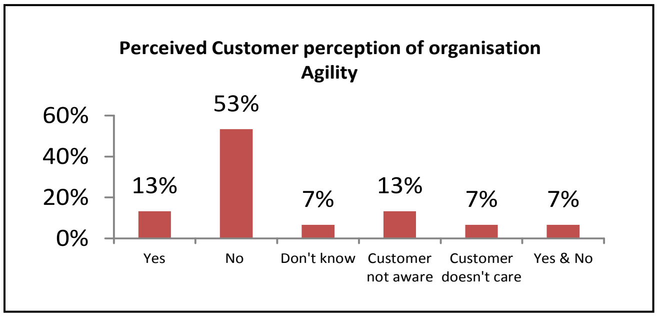

Figure 11. Management respondents view of customer perception of organisational Agility.

B. Developer/Team Results

In light of the management perception of the software development process portrayed in Section V.A the author hoped that the views of the development team might shed further insights into the software development process. More than half of the surveys collected (53%) represented the views of developers to many of the same questions but in addition the respondents were asked about the details of their software development processes.

As with the Management survey, initial questions focused on obtaining a profile of the respondents.

1) Respondent Characteristics

In terms of the size of the organisations in which the respondents work the analysis as shown in Figure 12, describes 43% as working in companies of less than 100 people, 33% between 101 and 500 and 24% in large organisations of greater than 500 people.

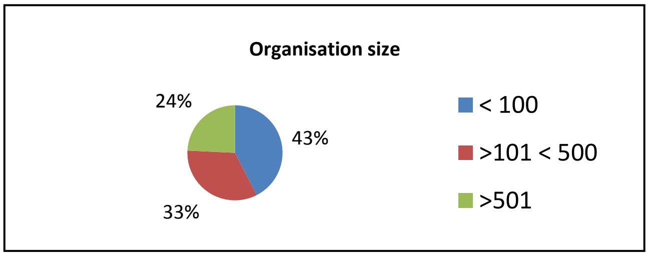

Figure 12. Developer survey organisation size.

As with the Management survey the respondents were representative of a diverse range of business sectors although as before, the telecoms/unified communications sector dominated at 46%. The software technology sector, as before, encompassed security and reward recognition software in addition to some unspecified software products. This is displayed below as Figure 13.

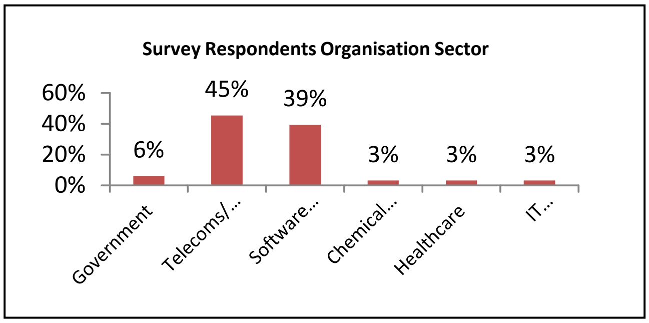

Figure 13. Developer survey respondents organisation sector.

Finally, the role of the respondents in their respective organisations was categorized as shown in Figure 14. It must be noted that the Part 2 survey was targeted exclusively at non-management personnel. It was hoped that the respondents profile would be that of the 'worker at the coalface' of software development (a software development team member) thus enabling the author to obtain the views of those individuals actively involved in the software development process. As shown in Figure 14 the survey did, indeed, capture a good cross section of these individuals.

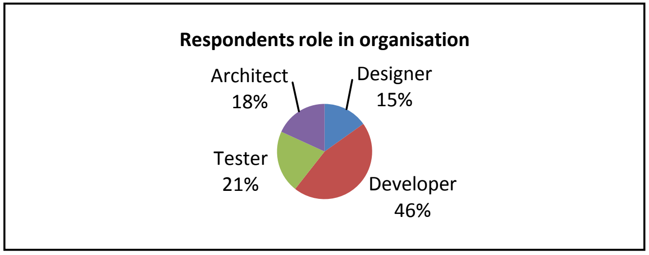

Figure 14. Developer survey respondents roles in organisation.

As before, when questioned about the Agile method used in their organisation to develop software the results were overwhelmingly in favour of Scrum. This is shown in Figure 15 below.

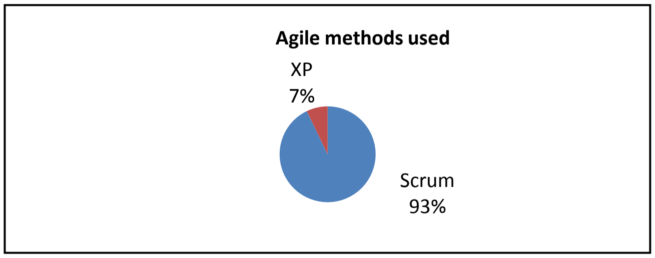

Figure 15. Agile methods used in respondents organisation.

a) *Scrum Team Respondents Results*

In light of those who stated that their organisation was Agile the author sought to verify that, in the context of their role, the survey respondents had the perception that their organisation was truly Agile when dealing with software churn. Figure 16 presents the overall picture.

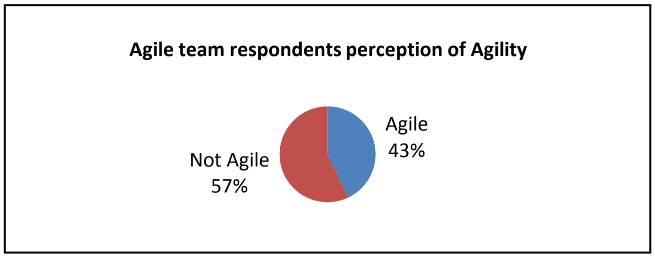

Figure 16. Developer respondents perception of organisation's agility.

As with the Management section of the survey the author wished to ascertain the level of satisfaction with the current Agile process being used. This is shown as Figure 17 below.

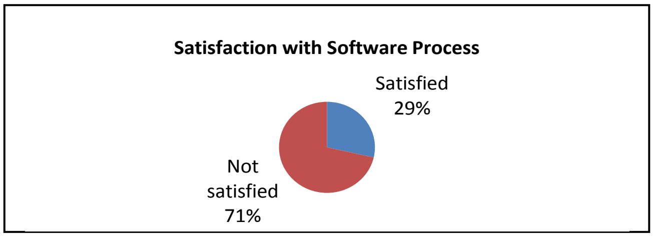

Figure 17. Developer survey respondents satisfaction with Agile process.

b) *Scrum Team Process Results*

The data collected in the next section specifically answers the questions on the implementation of those key aspects of the Scrum methodology that were detailed in Section III of this paper.

Of those organisations that use Scrum, Figure 18 displays the breakdown of perceptions regarding Customer involvement in the software development process.

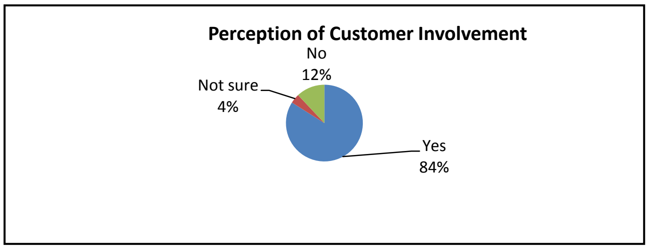

Figure 18. Developer respondents perception of Customer involvement in software development process.

100% of respondents admitted that the role of Product Owner was actively incorporated in their Scrum team.

This was followed by asking how often the Product Owner consulted with the Customer. The results are displayed in Figure 19 below:

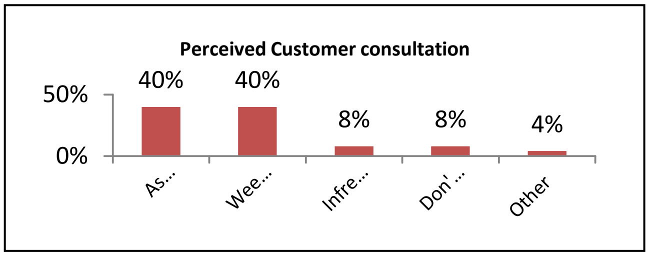

Figure 19. Team respondents perception of Customer consultation.

With regard to whether or not the software development process in the respondents' organisation facilitated the use of self-organised teams the respondents surveyed returned the breakdown as displayed in Figure 20.

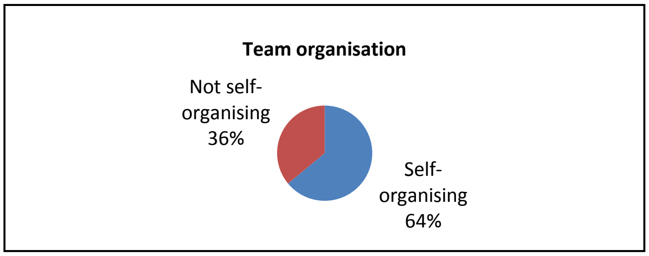

Figure 20. Developer survey respondents team organisation status.

One of the key tenets of Scrum is that working software should be released regularly and/or demonstrated to Customers to solicit feedback. Respondents were asked if software was released early to Customers according to this principle. The results are shown in Figure 21 below.

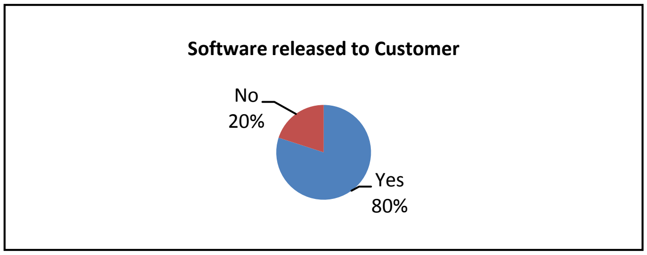

Figure 21. Software released for early feedback.

The final but nevertheless important question for Scrum teams centers around who sets the priorities for the team. As can be seen from the pie-chart in Figure 22 the results are somewhat varied.

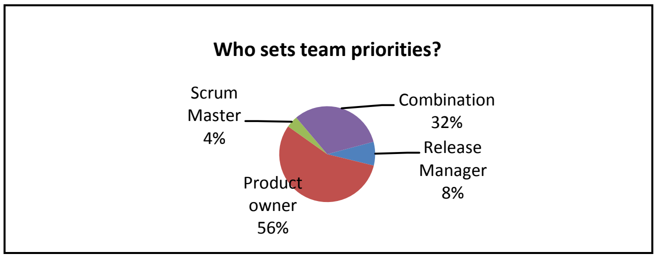

Figure 22. Setting of team priorities.

c) *XP Respondents Results*

The responses to the XP questions are not statistically valid given that the sample size was only three responses. However, in the interest of completeness, the results will be presented.

The first question to be asked of those respondents who described their software development process as Agile using XP concerned whether the Customer (or their designated, capable representative) was on-site during the development process. The results of this are displayed in Figure 23 below:

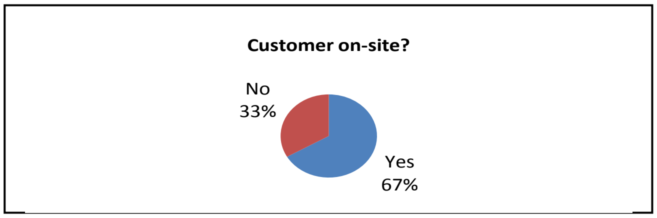

Figure 23. Presence of on-site Customer.

In response to the question on open-plan seating it transpired that all respondents claimed the team was situated in an open workspace.

Next, the issue of pair-programming was examined. The results are shown in Figure 24.

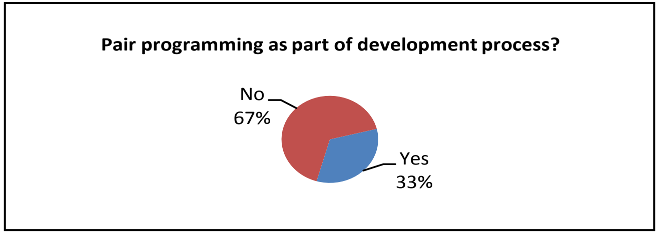

Figure 24. Utilization of pair programming.

Similar results were obtained for the team organisation in that as shown in Figure 25, 66% of respondents described their XP team as non-self-organizing.

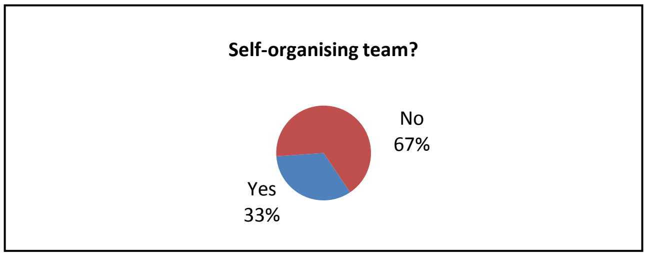

Figure 25. Utilization of self-organizing team.

The next section attempts to 'make sense' of the collected data with a view to answering the initial hypotheses and adding some general observations about the research process itself.

VI. DATA ANALYSIS/FINDINGS

As previously stated, the purpose of this research was to initially clarify which methods were being used for Irish software development. Following from this the author surveyed software professionals at both management and developer level in Irish companies to examine the perception of the agility of the software development processes being used and also to identify whether the precepts of the chosen Agile method were being inculcated into the actual development process as has been advocated.

The Management survey encompassed a broad spectrum of software development process management with Software Development managers being the largest sector at 42%. However, other senior/management roles from Project manager, Test manager, Scrum manager and Architect were also represented. Consequently, the author believes that a sufficiently wide management perspective was attained.

It can be observed from Figure 7 that those that claim to be fully Agile do not always perceive their own development process in this light. In fact, only 27% of the Management surveyed felt that their process was, indeed, Agile, 33% answered partially Agile but worryingly 33% of these respondents did not perceive their process as Agile. This is quite a surprising but nevertheless important result.

Perhaps the previous statistic accounts for the lack of satisfaction reported by Management respondents with the existing Agile processes in which those surveyed operate. 44% of Management respondents (Figure 8) reported a lack of satisfaction with the way in which Agile was being implemented. In terms of Customer perception, the author acknowledges that this should really be a question asked of Customers of the organisations surveyed. However, due to the logistics involved the author opted, instead, to ask the managers in the software development organisations if they were of the opinion that their Customers perceive their software development process as Agile. Whilst this is clearly a subjective question it was nevertheless included in order to identify whether, if, as the author contended an ad hoc approach to Agile software development was being used, the Customer was aware of this situation. Management were thought to be sufficiently close to such important project deliverables as on-time delivery, on-budget, and content as to be able to make a judgement call on their Customer's perceptions of the software development process. The findings of this question are that the majority of respondents (53%) do not. This is shown in the bar chart of Figure 11.

It was hoped that what is referred to, as the Developer/Team survey would represent the views and opinions of those individuals who are actually engaged in the 'hands on' activity of software development. This, indeed, proved to be the case as an analysis of the respondents' profiles showed that 46% were engaged in software development with test, design and development equally represented in the remaining 54%.

For the Developer survey it was interesting to note that 57% of those surveyed would not perceive their process as Agile. (cf. Figure16). One explanation for this listed by a respondent was "we are not yet fully Agile." Conceivably others who did not perceive their process as Agile had similar misgivings.

Possibly the most salient finding of the research dealt with the respondents' perception of the implementation of Agile precepts in their software development organisation. The author believes this to be one of the key elements of the research as it may explain the high level of dissatisfaction (44% according to Figure 8) reported by management with the Agile software development processes. Incidentally the level of dissatisfaction at the operational/developer level was significantly higher with 71% of respondents claiming to be dissatisfied with their software development process (cf. Figure 17).

One would have to have serious misgivings about the morale of these organisations. In any organisation were staff are felt to be dissatisfied with the way in which the development process is being conducted it would not be surprising to find a knock-on effect of software development ineffectiveness. The respondents surveyed were asked about the success or otherwise of projects they had been involved in in the past two years. The results of this are shown in Figure 26.

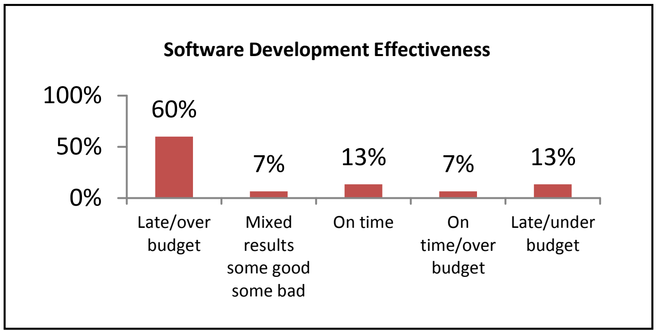

Figure 26. Software Development Effectiveness

With regard to the full implementation of Agile software development methods, specifically Scrum in those organisations represented by the respondents who completed surveys the author has used the data collected by the survey research to generate a table. The table, presented below as Table III, shows the adoption, or lack thereof, of the various key aspects advocated by the proponents of the methodology and explained in Section IIIA.

TABLE III. AGILE PRECEPTS ADOPTION

Suirvey #	Customer involved?	Who sets priorities?	Self org. team?	s/w rel early?
1	Yes	Release Mgr.	Yes	Yes
2	Yes	Release/Scrum	Yes	Yes
3	Yes	Product Owner	No	Yes
4	Yes	Product Owner	Yes	Yes
5	Yes	Product Owner	Yes	Yes
6	Yes	Release Mgr.	Yes	Yes
7	Yes	Prod. /Rel Mgr.	Yes	Yes
8	No	Product Owner	No	Yes
9	Yes	Product Owner	No	Yes
10	Yes	Product Owner	Yes	Yes
11	Yes	Product Owner	No	Yes
12	Yes	Product Owner	No	Yes
13	Yes	Product Owner	No	Yes
16	Yes	Product Owner	Yes	Yes
17	Yes	Product Owner	Yes	Yes
18	No	Mysterious pro	No	Yes
22	Yes	Scrum Mgr/Pro	Yes	Yes
23	Yes	Scrum Mgr/Pro	No	Yes
24	No	Scrum Mgr/Pro	Yes	Yes
25	Not sure	Scrum Mgr/Pro	Yes	Yes
26	Yes	Product Owner	Yes	Yes
27	Yes	Product Owner	Yes	Yes
28	Yes	Release/Scrum	No	Yes
30	Yes	Product Owner	Yes	Yes
33	Yes	Product Owner	Yes	Yes

As can be seen from Table III only 28% of the respondents are working in organisations that adhere to all of the Agile Scrum guidelines.

It should be noted that due to the limited amount of data collected this has not been done for the software development organisations that claimed to use XP.

VII. DISCUSSION

Firstly, this research confirms previous international findings that Scrum is the predominant Agile methodology in use in software development. In this sample of Irish software development industry this finding was found to hold true in both the Management and Team surveys with 83% of those in Managerial positions and 93% of Scrum team members reporting it as their organisations' Agile method of choice.

In terms of the actual research it was found in terms of the Scrum precepts, and, notwithstanding that the XP dataset was small, the XP precepts, that the actual implementation of the Agile methodologies was not as rigorous as had been hoped. Rather, the author's contention that organisations adopt an *ad hoc* approach to implementing Agile has been borne out.

Table V demonstrates the argument for this contention. Out of 33 surveys from team members who self-described as using Scrum only 18% were following all of the guidelines of Scrum as described by its proponents. This discounts the possibility that a limited number of 'rogue' organisations were not using a number of the Scrum guidelines. Rather, the data shows that 82% of those organisations surveyed were falling short in at least one regard.

There are many possible reasons for this. Firstly it is conceivable that organisations who have recently transitioned from a traditional approach to software development are experiencing difficulty in 'letting go' of the formal chain of command that frequently accompanied the more traditional plan-based methodologies, e.g., Waterfall. This would account for 30% of the anomalies in setting team priorities.

This same rationale would also account for the non-self-organizing teams. In order to transition to an Agile environment often the organisational culture will have to be changed to facilitate autonomous teams who are responsible for achieving team goals and managing their own workload. This would account for another 27%.

Secondly, when it comes to Customer involvement, this is a difficult arena where it is necessary to foster a trusting partnership with the Customer. It can be truly daunting to open up a software development organisation to the Customer and expose the organization's internal workings.

Based on the research it would appear that organisations who lay claim to being Agile are taking on board those guidelines which are relatively easy to implement. A case in point is the appointment of a Product Owner. As the Product Owner is often referred to as "the single wringable neck [2]" it is relatively easy to change the Traditional model role of Project Manager into the Agile Product Owner.

This theory is also borne out in the, albeit, limited data available on XP in that all of the survey respondents acknowledged an open plan workspace, which requires little organisational commitment but eschewed Pair Programming, which would require a paradigm shift in the software development operation.

VIII. LIMITATIONS & FUTURE WORK

It needs be stressed that survey research "captures a fleeting moment in time" [41]. It is completely possible that the response to a particular question might be totally different in the future as circumstances alter. Once this was taken on board, however, it was felt that a survey would be a perfectly acceptable way to discover information about this research topic. De Vaus [42] states "Survey research is widely regarded as being inherently quantitative and positivistic and is contrasted to qualitative methods that involve participant observation, unstructured interviewing, case studies, focus groups, etc. Quantitative survey research is sometimes portrayed as being sterile and unimaginative but well suited to providing certain types of factual, descriptive information – the hard evidence."

"If survey research has a drawback it would seem to be that the results are dependent on the participants' willingness to participate in addition to their ability to correctly answer the questions asked" [1]. Leedy & Ormrod [39] refer to the fact that the method relies on "self-report" data. The authors caution that "people are telling us what they believe to be true or, perhaps, what they think we want to hear."

Perhaps the greatest limitation of this research is its relatively small sample size. In total, the survey respondents numbered 45 individuals (cf. Table I and Table II). The margin of error on such a small sample is 14% but the author believes that due to the combination of quantitative and qualitative techniques employed the results are nevertheless indicative of the actual state of the Agile software development processes in Ireland. It was hoped that more data could have been obtained but given the short timeframe – the research was effectively conducted during the Summer of 2011 as part of a Masters dissertation in Software Engineering (MScSED) – this proved not to be the case.

Future work in this domain is ongoing specifically in the realm of Agile Scrum teams.

IX. CONCLUSIONS

The goal of this research was to add to the existing body of knowledge regarding Agile implementation in a sample of Irish software development organisations.

Conboy [43] declares "there is no consensus as to what constitutes an agile method." Undoubtedly, this research would agree with that statement.

The research set out to ascertain whether Agile practices are being implemented rigorously. The results would seem to indicate that, as hypothesized, this is not the case.

One would wonder if the lack of satisfaction with the respondents Agile processes could, in part, result from such an *ad hoc* approach. As Addison & Vallabh [44] advocate to control software projects it is important to "develop and adhere to a software development plan."

As was explained in Section III of this paper there is good rationale underpinning all of the Agile precepts and consequently there needs to be a similar well-reasoned rationale for excluding these self-same guidelines.

REFERENCES

[1] T. O'Connell, "The Scrum Product Owner – Customer Collaboration & Prioritizing Requirements," in Proc. ICSEA13, Venice, Italy, pp. 368-372, 2013.

[2] K. Schwaber, "Scrum Development Process," in Proc. OOPSLA'95 Workshop on Business Object Design and Implementation, Austin, Texas, USA, pp. 117 -134, 1995.

[3] K.Beck, Extreme Programming Explained: Embracing Change, Boston, MA. Addison Wesley, 1999.

[4] A. Cockburn, Agile Software Development, Boston, MA. Addison Wesley, 2001.

[5] VersionOne "7th Annual State of Agile Development Survey," retrieved 2014.01.20 from http://www.versionone.com/state-of-agile-survey-results/

[6] M. Fowler and J. Highsmith, "The Agile Manifesto," Software Development, Vol. 9, No. 8, pp. 28-32, Aug. 2001.

[7] L. Buglione, “Light Maturity Models (LMM); an Agile Application,” Proc. of the 12th International Conference on Product Focused Software Development and Process Improvement, pp. 56-57, 2011.

[8] S. Ambler and M. Lions, Disciplined Agile Delivery: A Practioner’s Guide to Agile Software Delivery in the Enterprise, Boston, MA. Pearson Education, p. 2, 2012.

[9] J. Dooley, Software Development and Professional Practice, New York, NY: Springer, p. 29, 2011.

[10] W. Royce., “Managing the Development of Large Software Systems,” Proceedings of IEEE WESCON Vol.26, No. 8, pp. 1-9, Aug. 1970.

[11] J. Sutherland, “Agile Principles and Values,” MSDN, retrieved 2014.02.04 from http://msdn.microsoft.com/en-us/library/dd997578.aspx

[12] O. Salo and P. Abrahamsson, “Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum,” in Proc. IET Software, Vol. 2, Issue 1, pp. 58–64, 2008.

[13] S. Millett, J. Blankenship, and M. Bussa, Pro Agile. NET Development with SCRUM, New York, NY: Apress, p.13, 2011.

[14] P. Deemer, G. Benefield, C. Larman, and B. Vodde, “Scrum Primer v 1.2,” Retrieved 2013.08.17 from http://goodagile.com/scrumprimer/scrumprimer.pdf.

[15] T. Barari, “Tips for First Time Scrum Masters,” Scrum Alliance, 2009. Retrieved 2013.08.16 from http://www.scrumalliance.org/community/articles/2009/may/tips-for-first-time-scrummasters.

[16] B. Schatz and I. Abdelschafi, “Primavera Gets Agile: A Successful Transition to Agile Development,” IEEE Software, Vol. 22, no. 3, pp. 36-42, May/June 2005.

[17] J. Highsmith, Adaptive Software Development Ecosystems, Boston, MA: Pearson Education Inc, pp. 244-245, 2002.

[18] T. Stober and U. Hansmann, Agile Software Development: Best Practices for Large Software Development Projects, Berlin Heidelberg: Springer-Verlag, 2010.

[19] R. Pichler, Agile Product Management with Scrum: Creating Products that Customers Love, Boston, MA: Pearson Education Inc, 2010.

[20] D. Rico, H. Sayani, & S. Sone, The Business Value of Agile Software, Fort Lauderdale, FL: J.Ross Pub., p. 8, 2009.

[21] H. Beyer User Centered Agile methods, San Rafael, CA: Morgan & Claypool, p. 4, 2010.

[22] J.R. Hauser & D. Clausing. (1988) "The House of Quality," Harvard Business Review, pp. 63 -73, May-June 1988.

[23] Standish Group International, CHAOS Summary for 2010 retrieved 2013.08.17 from http://insyght.com.au/special/2010CHAOSSummary.pdf.

[24] C.G.Cobb. Making Sense of Agile Project Management: Balancing Control and Agility, Hoboken, NJ, USA: Wiley & Sons, Inc., 2011, p. 114.

[25] J.L. Cooke, Agile Productivity Unleashed, IT Governance Publishing, Cambridge, UK, p. 109, 2010.

[26] J. Tata and S. Prasad, “Team Self-management, Organisational Structure, and Judgments of Team Effectiveness,” Journal of Managerial Issues, Vol. 16, No. 2, pp. 248-265, 2004.

[27] S. Koch, “Agile Principles and Open Source Software Development: A Theoretical and Empirical Discussion,” Proc. of 5th International Conference on Extreme Programming & Agile Engineering, pp. 85- 93, 2004.

[28] D. Rawsthorne, Patterns that make Scrum work: Understanding and Scaling Scrum, LeanPub.com, p. 16, 2013.

[29] C. Khalil, V. Fernandez and T. Houy “Can Agile Collaboration Practices Enhance knowledge Creation between Cross-Functional Teams?,**”** Proc. of the First International Conference on Digital Enterprise Design & Management (DED&M), pp. 123-133, 2013.

[30] L. Williams & R. Kessler, Pair Programming Illuminated, Pearson, Boston MA. p. 177, 2003.

[31] H. Kniberg & M. Skarin, Kanban and Scrum: Making the Most of Both, C4 Media Inc. USA, p.9-10, 2010.

[32] A. Martin, R. Biddle & J. Noble, “The XP Customer Team,” in Proc. of the Agile Conf., pp. 57-64, 2009.

[33] J. Shore: The Art of Agile Development: The XP Team, Sebastopol, CA, O’Reilly Media, 2007.

[34] N.B. Moe, T. Dingsøyr, T. Dybå, “Understanding Self-organizing Teams in Agile Software Development,” Proc. 19th Australian Conference on Software Engineering, IEEE, pp. 76-84, 2008.

[35] R.A. Guzzo and M.W. Dickson, “Teams in organisations: Recent research on performance and effectiveness,” Annual Review of Psychology, No. 47, pp. 307-338, 1996.

[36] L.A. Griffin, “A Customer Experience: Implementing XP,” in XP Universe, Raleigh, NC, USA, pp. 195-200, 2001.

[37] H. Robinson and H. Sharp, “XP Culture: Why the Twelve Practices both are and are not the Most Significant Thing,” in Proc. Agile Development Conference (ADC’03), pp. 12-21, 2003.

[38] J. Recht and M. Nielsen, “Discipline in Extreme Programming,” Retrieved 2014.03.01 from http://braindump.dk/dat8-discipline.pdf . p. 3.

[39] P.D. Leedy and J.E. Ormrod, Practical Research Planning and Design, New Jersey: Prentice Hall, p. 179, 2005.

[40] P.M. Nardi, Doing Survey Research: A Guide to Quantitative Methods, Boston, MA: Pearson Education, p. 119, 2003.

[41] C.A. Mertler, Action Research: Improving Schools and Empowering Educators, California: Sage Publications, California, p. 95, 2006.

[42] D. DeVaus, Surveys in Social Research. New South Wales, Australia: Routledge, p. 5, 2002.

[43] K. Conboy, **“**Agile Methods: The Gap between Theory and Practice,**”** in Proc. 5th International Conference on Extreme Programming & Agile Engineering, p. 316, 2004.

[44] T. Addison and S. Valabh, “Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers,” in Proc. SAICSIT, Port Elizabeth, South Africa, pp. 128-140, 2002.