

Spring 3-19-2013

A Tale Of Two Agile Implementations: A Cross-Case Exploratory Analysis

Nancy Russo

Northern Illinois University, nrusso@niu.edu

Siamk Shams

Infoglow Ltd, shams@infoglow.com

Guy Fitzgerald

g.fitzgerald@lboro.ac.uk

Follow this and additional works at: <http://aisel.aisnet.org/ukais2013>

Recommended Citation

Russo, Nancy; Shams, Siamk; and Fitzgerald, Guy, "A Tale Of Two Agile Implementations: A Cross-Case Exploratory Analysis" (2013). *UK Academy for Information Systems Conference Proceedings 2013*. 32.
<http://aisel.aisnet.org/ukais2013/32>

This material is brought to you by the UK Academy for Information Systems at AIS Electronic Library (AISeL). It has been accepted for inclusion in UK Academy for Information Systems Conference Proceedings 2013 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A TALE OF TWO AGILE IMPLEMENTATIONS: A CROSS-CASE EXPLORATORY ANALYSIS

Nancy L. Russo

Northern Illinois University, DeKalb, IL USA

Email: nrusso@niu.edu

Siamak Shams

Infoglow for Agile Transformation, London UK

Email: shams@infoglow.com

Guy Fitzgerald

Loughborough University, Leicestershire UK

Email: g.fitzgerald@lboro.ac.uk

Abstract

There is a growing trend towards the adoption of agile methods for information systems development in organisations due to the pressures for products and services to be faster to market. This paper examines the introduction and use of agile methods in two significantly different organizations; a bank and an international law firm. The adoption of agile methods in one organization was deemed a success whereas in the other it was not, although the agile approach did salvage an on-going project, but was not considered for further use. The cases were investigated via action research and the analysis adopted the competing values framework as the theoretical lens. This framework takes a cultural perspective to organisational effectiveness and requires the mapping of organisations into a four-quadrant model along the dimensions of organisational focus (internal versus external) and flexibility (change versus stability). Differences in the two cases and the organisational environments were examined for the purpose of identifying factors related to the successful and sustained adoption of agile methods. One organisation clearly emphasized stability over change and an internal focus, whereas the other was almost the complete opposite. The findings demonstrate the organisational cultural differences of the two organisations and their importance in the successful adoption of agile methods. Organisations should be aware of these differences and be wary of adopting agile development methods in otherwise historically and culturally non-agile organisations..

Keywords: Agile Methods, Case Study, Competing Values

1.0 Introduction

The use of agile methods has been growing as organisations observe the potential of agile methods to respond more quickly to market forces. In particular, a more dynamic and complex environment and the growth in web and mobile application have pushed software developers to seek alternative to traditional approaches. Avison & Fitzgerald (2002) describe the situation facing many organisations as “increasingly

competitive, more customer-focused, and operating in a more international context. Such a business environment is characterised by continuous change, and the information systems in an organisation need to be created and amended speedily to support this change. Unfortunately, information systems development in most organisations is unable to react quickly enough, and the business and the systems development cycles are substantially out of step” (p. 433). Levine (2005) identifies three causal factors that make internet speed software development different from traditional approaches, including: a desperate rush-to-market, a new and unique software market environment, and a lack of experience developing software under the new internet conditions and environments. Agile methods fit this need due to their focus on (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change over following a plan (Agile Alliance, 2002).

Although proponents of agile methods claim that such methods allow development teams to deliver better systems more quickly, not all reports of agile implementations contain positive outcomes. Both the practitioner and academic literature indicate that agile is not always easy nor without problems. For example, Dallas (2012), a President of a software development company, suggests that management is being seduced into the adoption of agile without fully understanding the issues and that it can simply be a faster route to disaster. Whilst, McAvoy and Butler (2009) report on a study of an unsuccessful agile development project and the teams failure to make changes in behaviours, attitudes and opinions that agile necessitates. Some research has shown that agile methods are difficult to implement in organisations that have a history of using approaches that are more planned and controlled (Nerur, Mahapatra, & Mangalaraj, 2005), (Cohn & Ford, 2003). This research seeks to take these issues further and investigate the wider issues related to the adoption of agile development methods by companies that have different organisational cultures and the effect that this has on success or otherwise. A serious implication being that the adoption of agile may be seriously undermined irrespective of the implementation efforts of the agile team, even before it starts. This has been recognised by (Beck, 1999) when he states that, "the biggest barrier to the success of an XP project is [business] culture". Grossman, et al. (2004) concur and suggest the major challenge is "to adapt and

reconcile the corporate and the agile culture processes and methodologies without seriously compromising either". However, there is a lack of research in this context with most agile research being anecdotal, according to (Chow & Dac-Buu, 2008). Thus, this study attempts to address this and reports on the experiences of two very different organisations, with different organisational cultures, in their deployment of agile methods in software development and their differential outcomes, using an organisation based cultural analytical framework.

Next we provide an overview of agile software development methods and factors related to their deployment and use. This is followed by a discussion of the research approach and the case study organisation contexts. To support the analysis of the differences in outcome between the two situations, the organisational culture of each case study firm is mapped into the Competing Values Framework. The paper concludes with a discussion of the implications of the results.

2.0 Background

2.1 Defining Agile Software Development

Boehm & Turner (2003) have described the characteristic of agility in terms of software development practices as:

“...the development group concentrates only on the functions needed at first hand, delivering them fast, collecting feedback and reacting to received information...This is the case when software development is incremental (small software releases, with rapid cycles), cooperative (customer and developers working constantly together with close communication), straightforward (the method itself is easy to learn and to modify, well documented), and adaptive (able to make last moment changes).”

Lindvall, et al. (2002) offer a version of the above, describing the basic characteristics of agility in software development as:

- Iterative software delivery, referring to short software development life cycle allowing for feedback and adjustments to the final product. Iterative software delivery ensures that the final product will remain relevant to the market and the customer by allowing timely changes to the product requirements.

- Incremental software development, which is the constant process of prioritisation of the required functionality through the iterative delivery of the product. Incremental development allows that product owners to react to conditions caused by the market and customer changes.
- Self-organizing teams, which means that a team has autonomy to organize itself to complete the work items. Self-organizing teams are composed of the skills required to deliver the product functionality.
- Emergent, which is the acknowledgement of the changes that will influence a software development project through its lifetime. The emergence refers to technology, requirements (customer and market) and other unforeseen influential facts that may be discovered throughout the project lifecycle.

Agility, for the purpose of this paper is considered to be composed of the following characteristics provided by Levine (2005), Boehm & Turner (2003), and Lindvall, et al.(2002): small and nimbleness of operations, adaptability to react to changes, iterative software delivery, incremental software development, self-organizing teams, emergent requirements and close cooperation of customers and software development teams.

Examples of agile approaches (most from Agile Alliance founding members) include adaptive Software Development (ASD) (Highsmith, 2000), Crystal (Cockburn, 2002a), Dynamic Systems Development Method (DSDM) (Stapleton, 1997), Extreme Programming (XP) (Auer & Miller, 2001; Beck, 1999), Feature Driven Development (FDD) (Coad et al, 1997) and Scrum (Schwaber & Beedle, 2002). For a detailed comparison of agile methods, see Abrahamsson, et al. (2002).

2.2 Factors Related to Successful Adoption and Use of Agile Methods

According to Turk et al (2003), due to the limitations of agile methods as defined by their mission statement (Agile Alliance, 2001), these methods may not be suitable for implementation for all software development projects. Some of the identified limitations of agile methods are (Turk, et al., 2002):

- Limited support for distributed development environments,
- Limited support for subcontracting,
- Limited support for building reusable artefacts,
- Limited support for development involving large teams,
- Limited support for developing safety-critical software,
- Limited support for developing large or complex software (doesn't scale).

In addition, agile projects may be more difficult to manage for those more comfortable with a “command and control” style. Due to the integrative and iterative

approach, clear milestones between linear phases such as analysis, design, and coding aren't available and it can be difficult for an outsider to determine if the project is on track. It is believed by some that agile development requires highly skilled and motivated individuals, which may not always be possible.

Factors related to the successful adoption and implementation of agile methods can be categorized as strategic, structural, technical, and cultural (Cram, 2012) or as related to people, processes, technology, and organisational factors (Nerur, Mahapatra, & Mangalaraj, 2005). A number of studies have looked at the relationship between organisational culture and the adoption and successful use of agile development methods. Chan and Thong (2008) describe a number of cultural factors that affect the adoption of agile methodologies: teamwork, individual ability, motivation, management support, communication, leadership, management style, management of software development knowledge, reward systems, and customer relationships. Adaptability, sociability, and task-orientation were identified as characteristics of organisational culture that are compatible with agile methods (Cram, 2012). Factors such as training, management involvement, access to external resources, and organisation size were found by Livermore (2008) to have an impact on the implementation of agile methods. The opinions of referent groups (subjective norms) and training were two factors that were demonstrated to be related to use of agile methods in a survey conducted by Vijayasarathy & Turk (2012). This study also found that perceived benefits had an impact on use of agile methods only when developers faced hindrances to adoption of the methods. Senapathi & Srinivasan (2012) observed positive relationships between sociological factors (experience, attitude, competency) and the use of agile methods and between organisational factors (top management support and the existence of a method champion) and the use of agile methods and productivity and quality improvements.

Lee and Xia (2010) found a complicated relationship between factors under the control of the organisation (team autonomy and diversity), agility (efficiency and scope of design), and performance (on-time, on-budget, and functionality). They concluded that organisations must make a tradeoff, based on their ultimate objectives. Team autonomy, but not team diversity, is related to efficiency. Team diversity has a negative impact on efficiency. Efficiency contributes to the likelihood of a project

being on-time and on-budget. On the other hand, scope of design (extensiveness) contributes to achieving the desired functionality, another measure of success. Both autonomy and diversity must be present to most effectively achieve extensiveness.

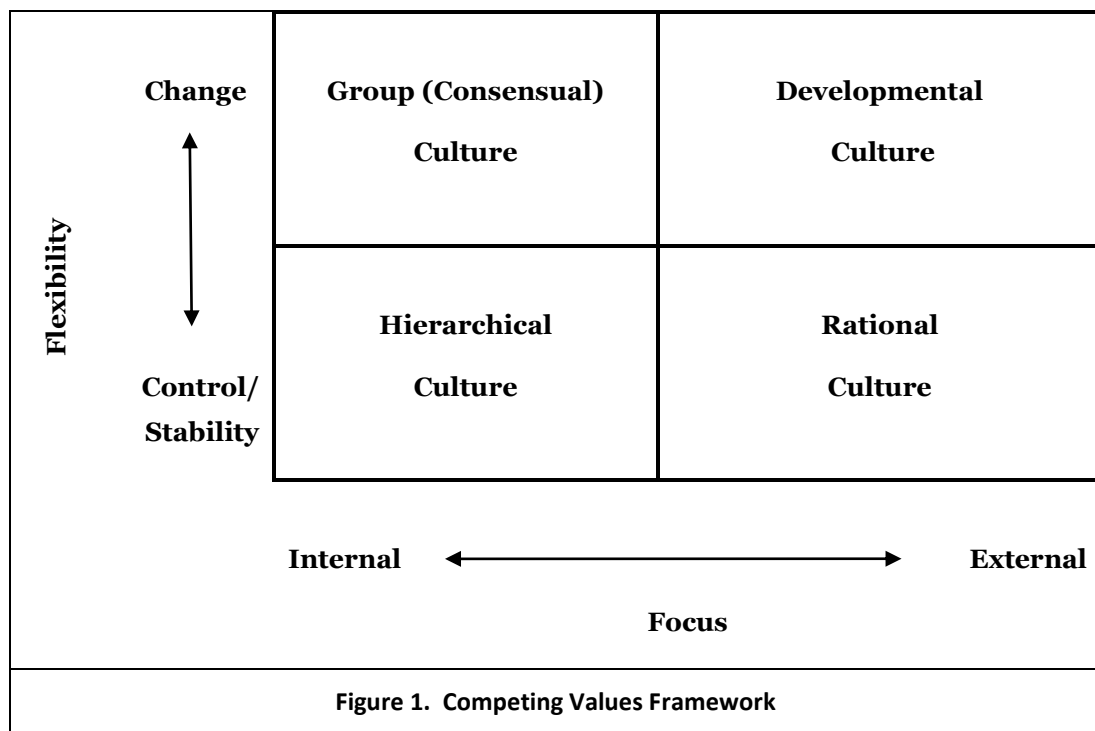
Chow & Dac-Buu (2008) looked at three measures of success: time/cost, scope, and quality. They found that team capability and delivery strategy were the factors most closely related to time/cost. Agile technologies, customer involvement, and delivery strategy had the greatest impact on scope. Agile techniques, team environment and project management were most closely related to quality. Misra, Kumar, & Kumar (2009) also looked at project success, but from a different perspective. In addition to time and cost (ROI), they included improved process and the ability to meet changing requirements. Misra et al. (2009) found that customer characteristics (satisfaction, collaboration, and commitment), short decision time, compatible corporate culture, team based control, personal characteristics of developers (attitude, willingness to learn), societal culture (communicative, dynamic, progressive), and training (less formal, with continuous learning) to be related to project success.

As this sampling of research on the factors related to successful adoption and use of agile methods shows, there is little agreement on exactly what does lead to success. Therefore, rather than attempt to develop yet another set of success factors, here we will instead seek simply to understand and explain the experiences of the case study organisations. To do this, a broader view of organisational culture will be addressed by using the Competing Values Framework. This framework provides a set of concepts that help us "understand how certain organisational cultures enable or constrain organisational changes" (Ngwenyama & Nielsen, 2003, p. 102), thus providing a means of comparing the cultures in the two case study organisations.

2.3 Competing Values Framework

The Competing Values Model of organisational culture is based on two primary dimensions: change versus stability and internal focus versus external focus. "Change emphasizes flexibility and spontaneity, whereas stability focuses on control, continuity, and order. An internal focus underlies integration and maintenance of the socio-technical system, whereas external focus emphasizes competition and

interaction with the organisation environment” (Iivari & Iivari 2011, p. 512 ; Denison & Spreitzer, 1991). The Competing Values Framework (Quinn & Rohrbaugh, 1983) allows us to map organisations into a four-quadrant model along the dimensions of organisational focus (internal versus external) and flexibility (change versus stability) as shown in Figure 1.



Other aspects of the organisational culture that can be used to locate an organisation in the model are the organisational orientation, organisational objectives, organisational structure, base of power, decision making, leadership style, compliance, evaluation of members, and orientation to change (Quinn & McGrath, 1985). These are shown in Table 1 below.

Aspect	Hierarchical	Rational	Consensual	Developmental
Organisational orientation	Stability and control	Productivity and efficiency	Cohesion and morale	Flexibility, adaptability and readiness
Organisational objectives	Execution of regulations	Pursuit of objectives	Group maintenance	Growth and development
Organisational structure	Routine tasks and technology; formal rules and policies	Complex tasks; responsibilities based on expertise	Complex tasks; collaborative work groups	Complex tasks; collaborative work groups
Base of power	Knowledge of organisational rules & procedures	Competence	Ability to cultivate relationships	Values
Decision making	Top-down pronouncements	Goal-centred, systematic and analytical	Participatory, deliberative	Organic, intuitive
Leadership style	Dominance, conservative, cautious	Rational achiever, goal oriented	Team builder, concerned, supportive	Idealistic, risk oriented, empowering
Compliance	Monitoring and control	Contractual agreement	Commitment to process	Commitment to values
Evaluation of members	Adherence to rules	Level of productivity	Quality of relationships	Intensity of effort
Orientation to change	Resistant (maintaining status quo)	Open to goal driven change	Open to change	Change is embraced as part of growth

Table 1: Competing Values in Organisational Culture (Ngwenyama & Nielsen, 2003; Quinn & McGrath, 1985)

3.0 Research Methodology

This study used an Action Research approach in two case studies. Action research is an iterative inquiry process that balances problem solving actions implemented in a collaborative context, with data-driven collaborative analysis to understand underlying causes enabling future predictions about personal and organisational changes (Reason & Bradbury, 2001). Action research moves beyond reflective knowledge created by sampling variables to active theory building, data collecting, and occurring in the midst of an emergent structure. Rather than using large samples and following a rigid protocol to examine a limited number of variables, case study methods are involved in in-depth, longitudinal examination of a single instance or event, a case. Case study provides a systematic way of investigating events, collecting data, analysing information, and reporting the results. As a result, the researcher may gain a sharpened understanding of why the instance or events happen as they do. Case studies may also help identify the areas of importance for future research. Case studies lend themselves to both generating and testing hypotheses (Flyvbjerg, 2006).

An alternative view of case study is that it is a research strategy that relies on multiple sources of evidence and benefits from the prior development of theoretical propositions. This view also suggests that qualitative research can be based on any mix of quantitative and qualitative evidence (Yin, 2002). The case study approach as adopted by this research is closely aligned with the above view of case study as an implementation technique for action research.

Utilizing action research with a case study strategy emphasises the interventionist nature of the researcher's role (Avison et al., 1999). In the case of this research, one of the authors ("the researcher") was a permanent employee of the companies under study throughout the duration of the case studies. The researcher was immersed in the environment and in direct contact with the phenomenon under study, allowing the researcher to introduce changes and examine the causes and effects relating to the cases.

4. Case Study Organisations

4.1 Egg Bank

Prudential, one of the largest UK insurers and investment institutes, in October 1998 launched a branchless deposit-taking and mortgage account telephone banking service called "Egg". Egg was the brainchild of founders Paul Gratton and Richard Duvall who were well known in the financial market sector for the introduction of innovative products to banks and financial institutes. Telephone banking proved to be a huge success, gaining 30,000 applicants in its first week and only 20,000 short of its six-month target. However, the unanticipated number of applicants contacting Egg bank in the short time after its launch caused difficulties for Prudential and Egg call centres in coping with the extra demand. In order to alleviate the pressure on the call centres, Egg IT proposed and launched an internet channel, the primary goal of which was to provide short term relief for the bank's call centres, at minimum cost and with expeditious effects. However, a month after the launch of the internet banking, the internet channel proved so successful that Egg bank stopped taking applications by phone, in favour of becoming the UK's first, pure online bank.

Egg bank's infrastructure was built on a variation of the standard three tiered web application architecture which included: client UI delivered by a web server and displayed through web browsers, security and middleware layers for secure interaction and application of bank's business logic, and the underlying customer and banking databases. The bank's software architecture was primarily built on proprietary components. At the time of establishment of Egg bank's online service channel, the selected tools did not offer connectivity interfaces, nor did they offer any industry standard methods and mechanisms of information interchange. The connectivity mechanisms were a bespoke set of software components built in-house by Egg developers. Egg's pragmatic approach to selection of hardware and software was based primarily on the speed of introduction of quality and cost effective software to the online community. This philosophy had caused Egg's hardware and software infrastructures to become a heterogeneous collection of operating systems, application software and tools. One of the medium to long term aims of the changes introduced by Egg was to standardise the hardware, operating system and software application tools across the board. However, the disparity in tools and platforms also extended to software development practices.

Egg bank, according to its founders, has had a long and established history of business agility. Business agility was necessary, partly because the bank was a small competitor within a market in which institutions had access to much larger operational and revenue reserves, and partly because of the vision of its founding members of creating a particular type of bank or institution which best reflected those ideals. In the initial stages of the establishment of Egg bank, 1998-2001 the management of the bank had focused on business agility and innovations in financial products. The practices of delivering software products in support of business agility were motivated by a single goal, namely that of being first to market, with a view to becoming and remaining, a leading player in the online banking arena. The speed of delivery goal dictated the choice of tools and the methods of software development and delivery, which in the course of this study, were found to be disparate and inconsistent. During interviews with project and release managers it was suggested that prior to introduction of agile methods, the delivery of projects were more important than methodical approaches to software development. By 2001, having met its initial goal of becoming a recognised player in the credit market, executives of the bank had

announced that the IT and software development practices could no longer support the agility required by the bank's business functions. Over a 3-year period, many software projects had been mothballed or abandoned, mainly as a result of difficulties with software production and support. Some of the reasons given for the latter included the following: production delays, budget overruns, over complicated architecture, lack of standard practices, lack of progress monitoring, lack of reuse, and growing operational distance between the software development and business teams. In an interview with a delivery manager in 2002 it was asserted that: "*we have not had one accurate estimate in producing software, and worse still we get to know about the delays and overruns a year or two away which is too late*". Some software development and support managers also attributed the difficulty with software provision to the lack of a cohesive approach and method to software development and support. Considering the scale of the bank's operation and capitalisation, the financial losses that occurred due to problems with delivery and support of the software, which were reported by a senior manager to run into millions in the 3-year period, were substantial. Furthermore, unlike most of the Egg bank's competitors who possessed multiple channels of interaction with their customers, online software products continued to be the only manifestation of the bank's business service provision to its customers. Ongoing problems with on-line software production threatened the bank's only interaction channel with its customers.

In light of the growing difficulties with software provisioning and IT failures to support the business goals, the bank's executive embarked on a wholesale program of change to include the company's structure, personnel, software development methodologies and reprioritisation of business activities to bring IT in line with the program of change while simultaneously changing the business to accommodate the pivotal role envisaged for IT within the organisation. The change program that started in 2003, which cost over £12m, encompassed all aspects of the company at all levels. For IT, the changes were structural and enabling, while for the business, the changes were introduced to increase synergy across the company functions, with the greatest emphasis placed on bringing IT and business operations closer. The change program, which was led by the technology director, ranged from the refurbishment of the new building to the introduction of new software development methodologies. Egg bank's

business units and the highest levels of executives championed the introduction of agile methods to business, and subsequently, to IT functions.

Egg bank opted to implement agile methods for its software development in order to remedy the growing dichotomy of methods employed by the business and technology. Egg bank had used a number of consultancy firms such as the design company Ideo (<http://www.ideo.com>) to embark on a £12m change program that included addressing the way software was produced (Watson 2003). Egg also employed some leading consultancies and individuals in the field of agile software development methods to advise the bank's IT management and staff on the implementation of agile processes and practices for development of the bank's new generation of software. However, a noteworthy observation throughout this research was that the main thrust of the advice received by the bank on agile methods, had a positive bias towards Extreme Programming (XP).

The agile method implemented at Egg bank was a variant of Extreme Programming (XP), augmented by techniques and practices from other agile methods to offer extended coverage. The changes to software development included a change of application development platforms centred on rapid software development tools. The Agile Egg Framework (AEF) is a hybrid method based on XP, while combining many of the best practices from methods such as Scrum, DSDM and others.

Egg used the Agile Egg Framework for its software development efforts and immediately saw benefits. According to Carol Moseley, the then technical change manager at the bank, the software for the Egg Card site had been built by three developers in just three months. This task would have taken many more months, had the legacy technologies been used. *"It's noticeable that we're able to develop things more quickly"* she remarked. Egg continued to use the agile approach for the remainder of the time the researcher was involved with the bank.

4.2 International Law Firm

The second case study examined the selection, introduction and implementation of agile methods for internal applications development at an international law firm based

in London, during March 2006 to September 2006. (Unfortunately the name of the firm must remain anonymous due to privacy and security concerns.)

The International Law Firm (ILF) provides legal advice to businesses and governments throughout Europe, the Middle East, Asia and the United States. The firm is privately owned by law and equity partners. At ILF the IT function was mainly defined by infrastructure and corporate applications support for the following: email, file server, web and remote access. ILF's approach to software applications had traditionally been to purchase Commercial-Off-The-Shelf (COTS) packages. Consequently, ILF's IT development teams were predominantly concerned with customisation and integration of COTS packages, rather than the production and support of software. ILF's board and executive level was made up of lawyers, law professionals and financiers. The IT department did not have representation on the executive board of ILF. At the highest level of IT management, a non-executive quasi partner, reported to the finance department, which in turn, represented the IT department to the board. Funding decisions for IT expenditure were made by a steering committee of partners which convened once every quarter.

ILF's competitive position vis-a-vis other law firms did not extend to or require innovative IT products. Their competitive edge was however, dependent on the provision of successful advice and its effective representation in disputes. The organisation's IT requirements were not a reaction to client's or market demands, rather, they arose as a result of "anticipated" regulatory legislations, or to facilitate business objectives through better and more intelligent access to client and case data.

At the time of this research, ILFs' developers or testers were mainly contract staff. This reflected the company's stance on software provisioning, which as mentioned earlier, focused on COTS packages rather than in-house software development. The numbers of fixed term contract staff, namely the applications developers and testers, as shown on the reporting hierarchy was dependent on the immediate requirements of the team. In broad terms, software projects at ILF fell into two main categories, which were business or IT initiated. Business initiated projects were intended to satisfy national and international voluntary or legislative legal requirements. The historical approach to satisfying business initiated software requirements was to

evaluate and purchase COTS packages. However, it was noted that most COTS packages purchased by ILF, were developed by software houses in conjunction with other law firms, and then packaged as COTS, or produced as commercial packages and marketed by law firms. This approach resulted in a closed-shop perception of software development in regard to law firms.

At ILF some of the IT initiated projects included, initiatives to extend the functionality of COTS packages, synchronisation of databases and the provision of reliable information regarding clients and matters (client's cases). In general, development initiatives were fixed cost, short term projects carried out by contract staff. The "integration Project" studied as part of this research, was the largest internal project undertaken by ILFs' software development teams, both in of its duration and cost.

Until the commissioning of the "Integration Platform" or the "Unification Project", studied as part of this research, most application development efforts carried out by ILFs' IT development were confined to short term customisation of COTS packages. Accordingly, due to the short term nature of the software development activities and the lack of in-house built and maintained software, ILF did not need to retain permanent teams of software developers because the customisation work was predominantly carried out by agency staff on short term contracts. The COTS applications and packages used at ILF were supplied with proprietary databases. The COTS package investigated as part of this research did not provide connectivity such as Application Programming Interact (API) or alternative mechanisms for integration with other systems and applications. Apart from the closed nature of the COTS packages and the data held within, restriction clauses included in software licenses prohibited ILF from maintaining the underlying databases.

It should be noted that ILF's web sites, which may have benefited the most from agile methods, were maintained by external companies. Furthermore, senior IT management personnel had stated that the nature of ILF business did not require its web applications to be any more than brochure-ware, or at best, a remote document access point.

ILF's software development teams were built on two layers of permanent staff members including analysts, architects and managers on one hand, and short term contract software developers and testers on the other. At the time of this research, all developers and testers were on short term contractors. The observed problems associated with the two layer system included a lack of continuity, lack of ownership, risk avoidance by developers and a tendency to follow plans without due consideration of the consequences. Another observation regarding the software development department was that it was divided into areas of technical expertise, each with their own structure, budget and mode of operation. The gated software development approach was embodied in the organisation of the teams and their deliverables. For example, a typical project life cycle included lengthy requirements gathering, as key components of all applications had to be verified by product sponsors who met once a quarter, systems and software analysis, followed by architectural design and development. The time lapse between the cycle of requirements identification and product delivery was a common complaint among stakeholders. Furthermore, the activities of the software lifecycle were conducted and completed in isolation, with little or no contact between stakeholders.

In 2004 ILF commissioned a survey of its IT software provisioning strategy to meet the firm's short and long term needs. As a result of the survey and the recommendations therein, ILF embarked on a program of unification of clients' and matters' (client cases) data collection and storage. An important component of the unification project was to devise a system to allow current and new COTS applications to communicate through a single standard communication protocol for storage and accessing the data. The change program initially focused on the production of an "Integration Platform". The Integration Platform entailed the design of an elaborate central database and an equally elaborate transport mechanism (EAI – Enterprise Application Interface) for communication between the components of the system.

The unification project was planned to be delivered in stages. Stage one was set to start Q3 2004 and end Q4 2006. Considering the initial investigation, analysis and design leading to the commissioning of the program, the total length of time envisaged was to be around 2 years, from beginning to end. The Integration Platform

was agreed to be delivered in a “big bang” approach as waterfall was the predominant method used by ILF IT.

In January 2006 it was acknowledged that the timeline was not feasible. While upholding the goals of the original design, a redrafted plan of action for the development of the Integration Platform was proposed and presented to the IT management. The new plan retained the core components of the original program, namely data storage and transport systems. Owing to the lengthy delays already experienced by the project, it was agreed that the new project would need to be developed using agile methods for iterative and incremental delivery of the product.

In Q1 2006, the original designers of the Integration Platform or Unification Project left ILF and the task of converting the original design to a working product was assigned to a newly formed agile team. The new team was charged with implementing the redrafted plan. The team allocated a period of six months from March to September 2006 for the initial development of the Integration Platform's project components. However, the agile team was tasked to complete within the same period, the installation and setup of agile processes and practices, and setup of the agile working environment.

The first things that had to be addressed were misperceptions regarding the nature of agile methods held by ILF's management. ILF's management were under the impression that agile methods could be applied to software development without the alteration and repositioning of support activities such as requirements analysis, design and testing. Previous mis-information received by management on agile methods had positioned agile methods as isolated practices for software developers without the need for wider engagement of IT and business personnel. Furthermore, ILF managers had been advised that by using agile methods, the overall length of time taken to deliver a product could be reduced while operating within the traditional Waterfall method for requirements gathering and analysis. Additional confusion arose as a result of advice regarding the differences in iterative delivery of functionality and delivery of product, as prescribed by Waterfall methods. The confusion resulted from the fact that ILF management had been advised that the length of iterations in agile

methods was indeterminate and could be as long as required. Therefore, 12-month iterations were seen as the norm.

As a first step in making the transition to a truly agile approach, a short program of training and open discussion regarding agile methods and available options was undertaken. The training plan was intended to inform ILF's software development staff about the available options with a view to reaching a consensus on the most appropriate approach for ILF. As a result of the discussions and workshops, a consensus was reached that Scrum, with a variation of XP, would be the most appropriate methods for ILF software development. It was also agreed that the transformation of ILF software development from non-agile to agile would be gradual and slow.

The Introduction of agile methods at ILF included a dual approach: staff and management education through technical and managerial workshops for introduction of the process and practices, and preparation of the development platforms for subsequent agile software development. The managerial workshops were primarily focused on Scrum processes and practices. Some of the practices emphasised by these workshops were the provision of a constant stream of prioritised backlog of work. The workshops also included prioritisation and grouping of backlogs into fixed for product releases. The management workshops also specified the set of Scrum project control artefacts appropriate to ILF.

The development workshops were primarily focused on XP processes and practices for coding, testing and refactoring. However, throughout the workshop sessions software development managers had reservations regarding specific XP practices such as pair programming. As a result, it was agreed that only occasional pair programming may be more suitable for ILF

A team of developers, tester, architect, data and business analyst were brought together for development and delivery the Integration Platform. The project team was modelled on recommendations of self-governing teams put forward by agile methods. According to interviews with IT management and team members, the successful delivery of iterative drops of software was primarily due to the team composition and

its ability to resolve issues internally without calling on external resources. Owing to the rapid delivery of the working software, the users and sponsors of the product were able to reduce the parts of the requirement which were deemed irrelevant. The Integration Platform agile team remained intact from March 2006 to September 2006, delivering several complete cycles of product drops. Within six-months of the introduction of agile methods, and as a consequence of the iterative and incremental delivery of software, the visibility of the Unification Project was raised and the components of this project were demonstrated

However, although tangible results had been shown, in June 2006 ILF's management decided to remove the business analyst (the customer representative) and the tester from the Integration Platform team on the grounds of the excessive cost of maintaining the team as it was. By the end of September 2006, the Integration Project, the first and only agile project at ILF, was comprised of developers only, with occasional access to business analysts and testers. The eventual conclusion of the Unification Project also ended the agile experiment at ILF.

The following section will compare the organisational contexts at the two firms using the Competing Values Framework.

5.0 Results: Organisational Culture and Agile Deployment

In the following table (Table 2) characteristics of the organisational culture of Egg and ILF have been mapped onto each of the aspects of the Competing Values Framework as identified by Ngwenyama & Nielsen, (2003), and Quinn & McGrath, (1985). As an aid to the reader, the identifying characteristics of each aspect for each of the four organisation culture types from Table 1 are repeated here (in italics). The particular descriptions of the case study organisations, Egg and ILF, on each aspect are summarized in the appropriate cell (or in the case of overlap, cells) in the model.

Aspect	Hierarchical	Rational	Consensual	Developmental
Organisational orientation	<p><i>Stability and control</i></p> <p>ILF: The main activities provided by the law firm had changed little in the past century, with the exception of data storage moving to databases.</p>	<p><i>Productivity and efficiency</i></p>	<p><i>Cohesion and morale</i></p>	<p><i>Flexibility, adaptability and readiness</i></p> <p>Egg: Long & established history of business agility, defined as “rapidly creating new products and services to meet the needs of customers, and interacting with customers in ways that suit and adapts to them ... Agility is essential to survival.” (Business Management, 2006)</p>
Organisational objectives	<p><i>Execution of regulations</i></p> <p>ILF: Responding to regulators’ demands rather than to clients’ preferences</p>	<p><i>Pursuit of objectives</i></p>	<p><i>Group maintenance</i></p>	<p><i>Growth and development</i></p> <p>Egg: Rapidly changing ecommerce marketplace, with new competitors entering, Egg sought to offer innovative financial projects at a speed that competitors could not match.</p>
Organisational structure	<p><i>Routine tasks and technology; formal rules and policies</i></p> <p>ILF: IT viewed as peripheral to business. All that was needed was “brochure-ware” on the web and remote document access.</p>	<p><i>Complex tasks; responsibilities based on expertise</i></p>	<p><i>Complex tasks; collaborative work groups</i></p>	<p><i>Complex tasks; collaborative work groups</i></p> <p>Egg: “Cutting edge applications; bringing together all the stakeholders in one place ... teams of IT and business could work in a unified fashion.”</p>

<p>Base of power</p>	<p><i>Knowledge of organisational rules & procedures</i></p> <p>ILF: Highest IT exec was non-executive quasi-partner who reports to finance department; not part of steering committee</p>	<p><i>Competence</i></p> <p>Egg: Old: dispersed and specialised teams with strict protocols for communication & cooperation. New: co-located, unified business and IT teams</p>	<p><i>Relationships</i></p>	<p><i>Values</i></p>
<p>Decision making</p>	<p><i>Top-down pronouncements</i></p> <p>ILF: Decisions on IT made by Steering Committee made up of law firm partners which met once a quarter</p>	<p><i>Goal-centered, systematic and analytical</i></p> <p>Egg: All decisions guided by market.</p>	<p><i>Participatory, deliberative</i></p>	<p><i>Organic, intuitive</i></p>
<p>Leadership style</p>	<p><i>Dominance, conservative, cautious</i></p> <p>ILF: Very conservative approach; averse to risk-taking</p>	<p><i>Rational achiever, goal oriented</i></p> <p>Egg: Old: Meet customer needs at all costs. New: One of the goals of the introduction of new approach was “a new style of life at Egg which included paying attending to work-life balance.”</p>	<p><i>Team builder, concerned, supportive</i></p>	<p><i>Idealistic, risk oriented, empowering</i></p>
<p>Compliance</p>	<p><i>Monitoring and control</i></p> <p>ILF: Focus on systems to satisfy national and international voluntary or legislated legal requirements and provision of reliable</p>	<p><i>Contractual agreement</i></p> <p>Egg: fiduciary regulations that could not be ignored, but focus on creating</p>	<p><i>Commitment to process</i></p>	<p><i>Commitment to values</i></p>

	information regarding clients and their dealings with the law firm	innovative products		
Evaluation of members	<i>Adherence to rules</i> ILF: national and international regulations	Level of productivity	Quality of relationships	<i>Intensity of effort</i> Egg: Speed, innovation
Orientation to change	<i>Resistant (maintaining status quo)</i> ILF: no need to change software even though the various pieces didn't communicate effectively; tried not the change development approach to accommodate agile methods	<i>Open to goal driven change</i>	<i>Open to change</i>	<i>Change is embraced as part of growth</i> Egg: saw survival as dependent on constant innovation

Table 2: Mapping Egg and ILF Organisational Culture to Framework

The organisational culture at Egg Bank could be described as very focused on meeting the needs of their customers, which requires constant attention on the marketplace and changing customer demands. Therefore, Egg would be positioned on the “External” end of the focus dimension. Egg Bank executives describe it as agile and adaptable, stressing the organisation’s focus on consistently providing innovative solutions to their on-line customers. Based on this, Egg would be placed high along the “Change” dimension, placing it squarely in the “Developmental Culture” quadrant. This is consistent with Iivari & Iivari (2011) who place enterprise agility in this quadrant. This fit between the organisational culture of Egg and the characteristics of the agile approach are confirmed by the early success and on-going use of agile methods at Egg.

At ILF, the organisational culture is much more conservative, with many formal rules and policies. As a law firm, adherence to regulations is critical; therefore monitoring and control is important. Decisions tend to be made from the top down, as seen with the Steering Committee for IT decisions, and change is made slowly and with some level of resistance. The culture at ILF would be located on the stability and control end of the flexibility scale, indicating a low tolerance for change. On the focus dimension, ILF represents a more internal than external focus, in that the firm is more concerned about meeting industry regulations than it is driven by customer demands. This would position ILF in the Hierarchical culture quadrant, which represents the cultural classification with the lowest fit with the agile approach. It is therefore not surprising that the agile method was not readily accepted at ILF and has not been used beyond the Integration Project.

Conclusions

From the experiences of both Egg Bank and the International Law Firm we see that organisations can achieve positive results from the use of agile approaches to software development. Egg Bank was able to increase the speed to market with innovative financial products while at the same time reducing the costs of producing the software by 13%. Whereas the use of agile methods were not continued at the International Law Firm, the short-term use of an agile approach supported the turn-around of the Integration Project and the delivery of a working solution in less than six months.

The Competing Values Framework provided a means of comparing the culture of the two organisations in an attempt to highlight factors which impacted the overall success of introducing agile into the organisation. On nearly all components of organisation culture included in the model, the International Law Firm fit into the Hierarchical quadrant, emphasizing stability over change and an internal focus over an external focus. It is not surprising, therefore, that the agile approach to software development was not viewed as a good fit with this organisation. At Egg Bank, on the other hand, the need for nimble, fast responses to a rapidly changing, competitive marketplace made the agile approach a very good fit. In mapping the organisational culture of Egg Bank to the Competing Values Framework, the evaluation of Egg Bank on most components placed it in the Developmental quadrant, where we expect to see

organisations with an external (i.e., customer) focus and a strong propensity for change and adaptability. As Iivari and Iivari (2011) discuss, agile methods are often described as “adaptive and flexible methods responsive to the environmental volatility” (p. 513) which corresponds to the Developmental culture.

This framework, of course, focuses our attention on only a limited view of the contexts into which the agile approach was introduced, at a certain point in time during which data was collected. There are a variety of other factors (some of which are discussed earlier in this paper) that could have an influence on the success of agile methods. In particular the model does not address the means by which a particular development technique is selected and deployed, the level of commitment of business and IT personnel, or the existence of top management support for the approach.

It could be worthwhile for an organisation to conduct a self-assessment to determine its position in the framework before embarking on an adoption of agile methods. A lack of alignment between organisational culture and agile principles reduces the likelihood of a successful implementation of agile methods. However, due to the mutually reinforcing relationship between the use of agile methods and organisational culture (Iivari and Iivari, 2011) it should also be noted that the adoption of agile methods could assist an organisation in an attempt to evolve its culture to be more open and flexible.

References

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). *Agile Software Development Methods: Review and Analysis*. VTT Publications 478.
- Agile Alliance. (2002). *Agile Manifesto*. <http://www.agilealliance.org>. Retrieved 2004
- Auer, K., & Miller, R. (2002). *Extreme Programming Applied: Playing to Win*. Boston: Addison-Wesley.
- Avison, D., & Fitzgerald, G. (2002). *Information Systems Development: Methodologies, Techniques and Tools* (3rd ed.). McGraw-Hill Publishing Co., UK.
- Avison, D., Lau, F., Myers, M., & Nielsen, P. (1999). Action Research. *Communications of the ACM*, 42(1), 94-97.
- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Boston: Addison-Wesley.

- Boehm, B., & Turner, R. (2003). Observations on Balancing Discipline and Agility. *Proceedings of the Conference on Agile Development* (p. 32). Washington, DC: IEEE Computer Society.
- Business Management. (2006). *Creating an agile business [online]*.
<http://www.busmanagement.com/pastissue/article.asp?art=26108&issue=156>.
 Retrieved 2007
- Cao, L., Mohan, K., Peng, X., & Ramesh, B. (2009, August). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4), 332-343.
- Chan, F., & Thong, J. (2008). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 46(4), 803-814.
 doi:10.1016/j.dss.2008.11.009
- Chow, T., & Dac-Buu, C. (2008). A survey study of critical success factors in agile software projects. *The Journal of Systems and Software*, 81(2008), 961-971.
- Coad, P., De Luca, J., & Lefebvre, E. (1997). *Java Modelling in Color with UML*. Prentice Hall.
- Cockburn, A. (2002). *Agile Software Development*. Boston: Addison-Wesley.
- Conboy, K. (2009, September). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3), 329-354.
- Cram, W. A. (2012). Aligning organisational values in systems development projects. *Management Research Review*, 35(8), 709-726.
- Dallas, A. (2012) Using agile to achieve rapid failure, MD+DI Online, 5th July, <http://www.mddionline.com/article/using-agile-software-achieve-rapid-failure>. Accessed 13/02/2013.
- Denison, D., & Spreitzer, G. (1991). Organisational Culture and Organisational Development: A Competing Values Approach. In R. Woodman, & W. Pasmore, *Research in Organisational Change and Development*, vol.5 (pp. 1-21). Greenwich, CT: JAI Press Inc.
- Flyvbjerg, B. (2006). Five Misunderstandings about Case Study Research. *Qualitative Inquiry*, 12(2), 219-245.
- Grossman, F., Bergin, J., Leip, D., Merritt, S., & Gotel, O. (2004) One XP experience: introducing agile (XP) software development into a culture that is willing but not ready. In Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research (pp. 242-254). IBM Press.
- Highsmith, J. (2000). *Adaptive software development: A collaborative approach to managing complex systems*. New York: Dorset House Publishing Company.
- Iivari, J., & Huisman, M. (2007). The relationship between organisational culture and the deployment of system development methodologies. *MIS Quarterly*, 31(1), 35-58.
- Iivari, J., & Iivari, N. (2011). The relationship between organisational culture and the deployment of agile methods. *Information and Software Technology*, 53, 509-520. doi:10.1016/j.infsof.2010.10.008
- Lee, G., & Xia, W. (2010). Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87-114.
- Levine, L. (2005). Reflections on software agility and agile methods: Challenges, dilemmas and the way ahead. In R. Baskerville, L. Mathiassen, J. Pries-Heje, & J. DeGross (Ed.), *Business Agility and Information Technology Diffusion* (pp. 353-365). International Federation for Information Processing.

- Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., . . . Zelkowitz, M. (2002, January). Empirical Findings in Agile Methods. *Lecture Notes in Computer Science*, 2418, 197-207.
- Livermore, J. (2008). Factors that significantly impact the implementation of an agile software development methodology. *Journal of Software*, 3(4), 31-36.
- Lycett, M., Macredie, R., Patel, C., & Paul, R. (2003). Migrating Agile Methods to Standardized Development Practice. *Computer*, 36(6), 79-85.
- McAvoy, J. and Butler, T. (2009) *A Failure to Learn in a Software Development Team: The Unsuccessful Introduction of an Agile Method*, in Information Systems Development, Conference Proceedings (ISD2007), Vol. 1, Edited by Barry, C., Conboy, K., Lang, M., Wojtkowski, G. and Wojtkowski, W., Springer, pp1-13.
- Misra, S., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *The Journal of Systems and Software*, 82(2009), 1869-1890.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, 48(5), 73-78.
- Ngwenyama, O., & Nielsen, P. A. (2003, February). Competing Values in Software Process Improvement: An Assumption Analysis of CMM From an. *IEEE Transactions on Engineering Management*, 50(1), 100-112.
- Patterson, M., West, M., Shackleton, V., Dawson, J., Lawthom, R., Maitlis, S., . . . Wallace, A. (2005). Validating the organisational climate measure: links to managerial practices, productivity, and innovation. *Journal of Organisational Behavior*, 26, 379-408. doi:10.1002/job.312
- Quinn, R., & McGrath, M. (1985). The transformation of organisational culture: a competing values perspective. In P. Frost, L. Moore, M. Louis, C. Lundberg, & E. Martin, *Organisational Culture* (pp. 315-344). Beverly Hills, CA: Sage.
- Quinn, R., & Rohrbaugh, J. (1983). A spatial model of effectiveness criteria: towards a competing values approach to organisational analysis. *Management Science*, 29(3), 363-377.
- Reason, P., & Bradbury, H. (2001). *Handbook of Action Research: Participative Inquiry and Practice*. London: Sage Publications.
- Schwaber, K., & Beedle, M. (2002). *Agile Software Development with SCRUM*. Pearson Technology Group.
- Senapathi, M., & Srinivasan, A. (2012). Understanding post-adoptive agile usage: An exploratory cross-case analysis. *The Journal of Systems and Software*, 85, 1255-1268.
- Stapleton, J. (1997). *DSDM: The method in practice*. Reading, MA: Addison Wesley.
- Turk, D., France, R., & Rumpe, B. (2002). Limitations of Agile Software Processes. *Proceedings of the Third International Conference on eXtreme Programming and Agile Processes in Software Engineering*. Italy.
- Vijayarathy, L., & Turk, D. (2012). Drivers of agile software development use: Dialectic interplay between benefits and hindrances. *Information and Software Technology*, 54(2012), 137-148.
- Watson, J. (2003, November 12). Egg shells out £12m in development restyle. *Computing*. Retrieved 2005, from <http://www.computing.co.uk/computing/analysis/2075839/egg-shells-12m-development-restyle>
- Yin, R. K. (2002). *Case Study Research, Design and Methods* (3rd ed.). Newbury Park: Sage Publications.