

Distributed Scrum Projects as Business Transactions

Robert T. Hughes, School of Computing, Engineering and Mathematics, University of Brighton, United Kingdom

Abstract

This paper has as its starting point a study of a UK-based software development company (disguised by the name ‘Seagull Software’) which implemented a software project using Scrum for a client (disguised as ‘Koala Travel’) based in Australia. The project required intensive, structured, communication by telephone conferences and email. Although a common assertion is that approaches like Scrum need face-to-face contact, Scrum was adopted for this project. An opportunity arose for a researcher to observe the communication between the stakeholders as a silent listener at the telephone conferences and as the recipient of copies of emails.

Having gathered data, the analysis here examines four well-known business and organisational models: Roger’s diffusion of innovation (DOI) model, Hofstede’s cultural model, the Austin/Levin novelty cost model, and Porter’s Five Forces model. The relevance of each model to the circumstances of the agile project is assessed.

This leads us to examine the software development projects as business transactions between parties whose bargaining power varies according to the market forces explored by business thinkers such as Michael Porter.

1. Introduction

Attention has been drawn to the degree to which research has ‘impact’ by the recent changes to the requirements for research proposals that are submitted to UK research councils. Various objections have been made to what are seen as yet more bureaucratic hurdles. As a co-author of a moderately successful textbook [Hughes and Cotterell, 2009] on software project management, I feel that if the ‘impact’ of research is to be assessed then the consumers of research should be more widely consulted. These include research peers, but also practitioners and even the humble producers of text books that in successive editions are expected to track developments in a particular field.

Like many innovations in IT and software development practice, the source of agile software development has largely been practitioners rather than academic researchers. This makes things more complicated for writers trying to capture the ‘state of the art’ in software engineering, especially as there can be wide gaps between what is promoted as good practice and the reality of practice in business. Examining software developers as *a community of practice* (COP) is therefore instructive. It allows questions of how novel practices (such as agile software development) come to be adopted.

The idea of COPs is an example of an explanatory model. In developing teaching materials from research findings, I believe teachers in general seek out ‘models’ – descriptions or theories that in some way generalise the experiences gathered through research and practice and which can be applied to new situations. This ties in neatly with some of our recent research which has examined the use of cognitive causal maps to support the collaborative identification by stakeholders of what they perceive are the factors that had an influence on

the success/failure of a past project. The map can then be used to suggest measures that can be taken to reduce the risk of failure in future projects.

Most of the subject matter of the discipline of software engineering can in fact be seen as attempting to reduce the risk of failure in software development. Agile development can be seen as essentially a risk reduction strategy. This is illustrated by the ISPL/Euromethod guidelines [Turner and Jenkins 1996] on the adoption of project strategies – see Table 1 for just one element of this. This is itself another model.

Project characteristic	Recommended approach
Uncertainty	Exploratory e.g. the use of prototypes
Complexity/size	Incremental
Deadline pressure	Incremental or prototyping

Table 1 Project Strategies.

In the remainder of this paper we commence by briefly introducing a case study based on the observation of a project using Scrum where the developers and clients were based in the UK and Australia respectively. We will then examine four organisational and/or business models and assess the degree to which they can be usefully applied to the case study. The four models are Roger’s diffusion of innovation (DOI) model, Hofstede’s cultural model, the Austin/Levin novelty cost model, and Porter’s Five Forces model.

2. Seagull software: a study of distributed Scrum software development

The increased use of external bodies to carry out what were previously core business operations has been advocated by business writers such as Evans and Wurster [2000]. An example is the travel sector where, to reduce processing costs and to provide a more convenient service to customers, airlines supply electronic tickets to customers who book on-line. Legal requirements mean that on-line transactions have to be confirmed using an alternative communication channel which is most conveniently email. Rather than generate these emails in-house, travel companies can employ a company like Seagull IT (a real company, but a fictitious name) to create the emails on their behalf. Travel companies already use shared third party IT services, for instance the Amadeus centralized flight reservations system, and Seagull IT obtains some of the information needed in the customer emails from the Amadeus passenger record or PNR.

Seagull IT has developed and manages a batch processing system to carry out these operations. Many travel companies use this service, their specific formatting requirements being recorded in and accessed from a number of customized electronic templates. The business case for using this service is that it is cheaper for the travel company to use the existing assets developed by Seagull IT than to develop their own capability – even if cheaper resources in developing countries are used.

Despite existing Seagull IT software assets, each new client added to the Seagull IT customer base will require some development work to cater for the travel company’s individual needs. Care is required in the design of the automated emails, which appear to the recipient travellers as coming directly from their travel company. The travel company will need to supply graphics, style guides, templates and data feeds to ensure that the look and feel of this communication is consistent with their overall brand. To be successful this process requires considerable interaction with the client. Seagull IT has adopted a Scrum approach to this task.

Communication was a key project challenge. Same time (or ‘synchronous’) communication such as telephone conferences should, where possible, take place during normal office hours for all parties. The overlapping period of one hour at the end of the Australian day and the beginning of the UK day allowed this. If Koala Travel clients made a request at the end of their working day, Seagull IT had a working day to fulfil the request before the client returned to work. However, a Seagull IT developer with query and waiting for the next same-time meeting would lose a whole Australian working day. Using different time (‘asynchronous’) communication, in practice email, was quicker in this case.

Email was a key communication channel. However, a telephone conference initiated the project with a Sprint Planning Meeting. The Product Owner, who was the business lead for Koala Travel, participated in the daily Scrum meetings which were conference calls on Tuesdays and Thursdays. On other week-days, the developers at Seagull IT held their own face-to-face Scrum meetings.

Further details of this case study can be found in Hughes [2011].

3. Matching the case study against models

3.1. Agile software development as an innovation

Communities of practice (COPs) can be seen as existing within organisations where there may, for example, be groups of software developers who work together, and as straddling organisations where there is a distinct professional body of knowledge and expertise associated with a group of specialists (as with say, accountants). COPs can vary widely in the degree of formality and recognition they are afforded. In this case we are interested in the relatively informal COPs associated with agile software development. The lens of COPs can take a broader view of methodologies such as Scrum, XP and DSDM than as simply collections of models and techniques.

Such methodologies can be examined as a particular type of innovation. The factors that influence the diffusion of innovation (DOI) has a very strong research traditions associated particular with the work of Rogers [1983]. Fichman and Kemerer [1993] applied Everett’s DOI to Object Oriented development. Everett identified the attributes of an innovation that had a bearing on ease of acceptance as:

- Relative advantage over existing technologies
- Compatibility with existing values, skills and work practices
- Lack of complexity
- Ease of trialling
- Observability

It appears to us that Scrum rates highly on all of the ease of acceptance factors except perhaps compatibility. The defects in the ‘traditional’ waterfall approach to projects have been acknowledged as far back as McCracken and Jackson [1982] and agile methods are seen as a way of addressing these shortcomings. Scrum is not complex, and may be more simple to grasp than DSDM. Scrum can be trialled by use on a single project. It does not require any special software tools. The outcomes of Scrum can be observed. Some have gathered and published statistics claiming considerable productivity gains for Scrum [e.g. Sutherland et al 2009]. Although the principles of Scrum are straightforward, they require participants to

adopt new roles at work, and for management to take a more ‘hands-off’ attitude to the conduct of a project: some may not take easily to these new behaviours.

While it is may be relatively easy to adopt Scrum, it is also easy to water down the application of its principles. Subsequent contact with Seagull suggests this may be the case with this organisation: a slipping back into a more waterfall mode of working was reported following the departure of the key evangelist of Scrum.

Fichman and Kemerer identified other economic factors that affected the acceptance of innovations . These included sponsorship and expectations. It can be seen for example that the ‘success’ of PRINCE2 is largely owing to its promotion by the UK Government through the Office of Government Commerce. Recently there has been a flurry of excitement in the DSDM agile development community with the publication of the Institute of Government report *System Error* which recommends the adoption of more agile practices on government IT projects. The prospect of official government sponsorship of DSDM, making it somehow the new PRINCE2, is very appealing to that community. An associated factor identified by Fichman and Kemerer is expectation, a belief that the innovation is likely to become widely adopted in the future. A community of early adopters will want to promote the idea that what they are doing now, everyone will be doing tomorrow. (I recall twenty years ago being told that in the future or IS development would be based on formal mathematical methods).

3.2 Hofstede cultural differences

A common observation about attempts to use agile approaches to distributed projects is the challenges of overcoming cultural differences [see for example, Fowler 2006, Filev 2011]. Fowler reiterates the aspiration of agile approaches to support intense and frank communication between co-workers, which can be at odds with more traditional management approaches. He states ‘*We find this to be a big problem in western companies, but the problem is amplified in Asia since Asian cultures reinforce deference to superiors*’. With the Seagull scenario we are unable to comment on this directly, as the Australian and UK cultures are so similar. If anything one would expect Australians to be less deferential than the British. In general, while clearly there is always a possibility for cultural differences where people in different countries have to collaborate, in our view there is a risk that these differences can be exaggerated. Hostede [1984] is probably the most widely referenced authority on cultural differences. Based on the analysis of a global employee questionnaire survey for IBM, Hofstede identified four attitudes which distinguished difference national groups: power distance between staff and managers, the desire to avoid uncertainty, individualism versus collectivism, and masculinity/femininity. Some have tried to interpret emergent national characteristics as having an impact of practical software development. Turner [1997] suggested that Eastern cultural attitudes were more applicable at the project initiation and close-out stages of a project while Western attitudes were better for the delivery phase.

One might expect that a smaller perceived power difference between staff and managers might be compatible with Scrum and here the UK scored better than India. However one might also expect that with agile approaches a greater acceptance of uncertainty, reflected in a willingness to defer design questions for example would be desirable. Here the Indian score was very close to that of the UK. You might also expect that a less individualistic and more collective mindset would support agile, and here the UK is hugely more individualistic than India. The Hofstede ratings are very difficult to apply to real scenarios. For example, in Hoecklin [1995] a situation is reported where Italian and UK managers had to work together.

The Italians perceived the British to be well-organised analytical thinkers, but to be very much preoccupied with rules and procedures. One might expect from this that the Italians have a higher uncertainty tolerance level, yet the relevant scores were Italy 75 for uncertainty avoidance, while for the UK it was 35.

It is our view that many of the perceived differences between ‘on-shore’ and ‘off-shore’ attitudes of a distributed project may have different and more nuanced causes than simple matters of national culture. For example, the very terminology, ‘on-shore’ and ‘off-shore’ implies a power difference. In the Seagull scenario, the Seagull end is regarded as on-shore because we share the same island with Seagull, but the Koala organisation would equally regard themselves as ‘on-shore’. The ‘on-shore’ organisation is regarded as the central, controlling, entity. In many scenarios, it prescribes the project approach. In most projects, there is a clearly defined client who provides the finance, and suppliers who carry out work for client. In Section 3.4 we explore how this business relationship could influence the project environment.

3.3. Austin/ Devin model: novelty cost model

Austin and Devin [2009] present a model that attempts to explain where agile software development would be preferred to more structured approaches. They suggest that a key influence is the perceived value of novelty. In many cases customers may be satisfied with a product that is very standard and therefore relatively cheap to buy while sometimes they may require a unique product. One example might be the case of tools where for many tasks a standard multipurpose implement like a hammer is sufficient but in some cases a specialist tool may be needed, which might even need to be made specially. It is suggested that where there is a demand for novelty then craft-like agile approaches may be needed.

For us this model seems to elucidate the choice between off-the-shelf and bespoke software that a software user might be confronted with. The grounds for selecting a standard ERP solution versus a system built specifically for an organisation’s needs are well explored – see for example Davenport [1998]. However, either an agile or a traditional waterfall development approach could be used to produce a unique product.

In practice, the difference between standard and unique systems is not always clear cut. In the Seagull scenario, Seagull had a basic platform to generate and distribute standard emails to specified recipients in a batch process. It can be seen that this type of system would benefit from a relatively structured approach. When a new client was recruited, templates needed to be developed that reflected the corporate image of that client. The recipient of the email would need to be convinced that it came directly from the client (a sort of authorised ‘phishing’). This process of customisation was one where intense iterative interaction between developers and client was beneficial and where an agile approach seemed appropriate. The concept of customised components being built on a standardised platform was popularised by Ross et al [2006] and has now been recommended in an Institute for Government report on UK government IT development [Stephen et al. 2011].

3.4 Porter five forces model

Michael Porter devised a ‘Five Forces Model’ which explains the likely competitive situation of in a business sector [Porter 1980, 1985]. Building on this now well-established model, Porter subsequently extended his analysis to the impact of the world wide web on business completion [Porter 2001].

Porter identified five forces affecting a business's competitive position:

- The potential for new entrants to enter the market;
- The rivalry between existing firms;
- The pressure of substitutes
- The bargaining power of buyers;
- The bargaining power of suppliers.

One of Porter's insights is that the relative power of a buyer and seller in any situation will vary because of a number of characteristics of the nature of the business in which they operate. For example when you want to buy a computer printer for home use, there are a number suppliers competing for your custom and thus prices tend to advantage the buyer. Once the printer is purchased, the cost of printer cartridges is very high but you have little choice but to buy the printer cartridge that your printer requires. The bargaining power of the buyer and seller has changed because there is now a barrier to you going to a competitor.

Many IT development projects use external resources, so that the project is based on a buyer/seller relationship. The ISO 12207 standard on 'Information technology: software lifecycle processes' assumes such a relationship. It follows that the competitive positions as identified by Porter will have a significant impact on the relationship between client and provider and on the project management of the project. For example, one example of a factor which increases the power of a buyer is where there is a small number of buyers for a product or service. A supplier may for example supply technical equipment to telecommunications companies. Because there is a relatively small number of these then getting a sale is important and prices will be squeezed. Where software is being supplied a large organisation can mandate the processes used by the supplier by insisting on compliance with standards such as CMMI, ISO9000 and PRINCE2. On the other hand, where suppliers have a large number of small customers they can afford to avoid those clients who make them 'jump through hoops'. What is remarkable in the Seagull scenario is that here a small supplier was able to persuade a large client organisation to use an agile approach in a situation where at first sight it did not seem suited.

In the Seagull scenario, the balance between the supplier and the buyer, Koala Travel, appeared to be fairly even. Koala was a large travel related industry with a global reach, while Seagull was a small specialist firm. This would tend to give Koala the upper hand. Seagull's business was based in the internet. While this gave it access to customers right across the globe, it also meant that potential rivals could emerge from any corner of the globe, including from many low-wage economies.

However, Seagull had some advantages. It currently has few (if any) competitors. Over the course of acquiring customers, it has built up reusable assets in terms of server systems, specialist software and developer expertise which means that new clients can be added at modest cost. New entrants to this market would have to invest heavily to develop these types of asset. Once contractually engaged, the client would not be able to switch to a new service provider without the costs of redeveloping the emailing templates for the new supplier.

Using Scrum in the way it was required intensive effort from client staff. While this was to Koala's advantage as it allowed the new facility to be developed quickly, it seems unlikely that the level of time commitment could be maintained with projects with longer durations. Koala also imposed, in addition to Scrum, a conventional acceptance testing process which appeared to be managed in a traditional manner.

5. Concluding remarks

In this paper we attempted to demonstrate how an agile software development project which had an overseas client could be examined from the viewpoints of some examples of cultural and business models. In general, it can be said that the Hofstede cultural analysis was found to be less helpful than might have been expected, while the Austin-Devin did not seem a good fit with the particular software development environment under examination. Porter Five Forces model provide many interesting insights, but we would be very hesitant to use this as a predictive tool.

There appears to be a major obstacle to agile development where the developers and clients are in different organisations. In these cases, a legal contract is likely to be involved, typically laying down requirements to be met, and probably a fixed cost. It was put to us that ideally agile developers would like a fixed price contract with unfixed requirements, or a time and materials contract. This is asking a lot, particularly where the supplier is on a different continent.

This does not mean that agile approaches cannot be used, but that there is a risk that they will be watered down. A key question for the future is to what extent can agile methods be compromised without their benefits being lost. Will PINO (PRINCE2 in name only) be followed by DINO (DSDM in name only) or SINO (Scrum in name only)?

References

- Austin, R.D and Devin, L (2009), Weighing the Benefits and Costs of Flexibility in Making Software, *Information Systems Research* 20(3) 462-477
- Davenport, T.H. (1998), Putting the enterprise into the enterprise system, *Harvard Business Review*, 76(4) 10-18
- Evans, P and T. Wurster (2000), *Blown to bits: how the new economics of information transforms strategy*, Harvard Business Press.
- Fichman, R and Kemerer, C.F (1993), Adoption of software engineering process innovations: the case of object orientation, *Sloan Management Review*, 34 7-22
- Filev, A (2011), Adopting and benefiting from agile processes in offshore software development. Online at <[http://msdn.micorsoft.com/en-us/library/bb245671\(printer\).aspx](http://msdn.micorsoft.com/en-us/library/bb245671(printer).aspx)> Accessed 4th January 2011
- Fowler M (2006), Using an Agile Software Process with Offshore Development, online at: <<http://martin.fowler.com/articles/agileOffshore.html>>
- Hoecklin, L (1995), *Managing Cultural Differences: Strategies for Competitive Advantage*, Addison-Wesley
- Hofstede, G (1984), *Culture's Consequences: International Differences in Work-related Value.*, Sage
- Hughes, B and Cotterell, M (2009) *Software Project Management*, McGraw Hill
- Hughes, R.T. (2011), Observing Dispersed Agile Project Processes, Proceedings of the European Conference on Research Methods in Business, Normandy Business School, Caen 20-21st June
- McCracken, D.D., and Jackson, M.A. (1982), Life cycle concept considered harmful, *ACM Software Engineering Notes* 7(2) 29-32
- Porter M.E. (1980), *Competitive Strategy*, The Free Press.

- Porter M.E. (1985), *Competitive Advantage*, The Free Press.
- Porter M.E. (2001), *Strategy and the Internet*, *Harvard Business Review*, 79(3) 62-78
- Rogers, E.M (1983), *Diffusion of innovations*, Free Press
- Ross, J.R., P.Weill, and D.C.Robertson (2006), *Enterprise architecture as a strategy: creating a foundation for business execution*, Harvard Business School Press
- Stephen, J., Page, J., Myers, J., Brown, A., Magee, I. (2011), *System error: Fixing the flaws in government IT*, Institute for Government IT. Online at <http://www.instituteforgovernment.org.uk/publications/23/> accessed 25th May 2011
- Sutherland, J., Schoonheim, G., Rijk, M. (2009), *Fully Distributed Scrum: Replicating Local Productivity and Quality with Offshore Teams*, 42nd Hawaii International Conference on System Sciences, Waikoloa, Hawaii, January.
- Turner, J.R. (1997), *Cultural Profile for Project Management*, *The Handbook of Project-based Management*, McGraw-Hill
- Turner, P. and Jenkins, T. (1996), *Euromethod and Beyond: Open Frameworks for European Information Systems*, International Thomson Computer Press.
- Wenger, E (1998), *Communities of practice: learning, meaning and identity*, Cambridge University Press