



OpenAIR@RGU

The Open Access Institutional Repository at Robert Gordon University

<http://openair.rgu.ac.uk>

This is an author produced version of a paper published in

ICGSE 2013: 2013 IEEE 8th international Conference on Global Software Engineering (ISBN 9780769550572)

This version may not include final proof corrections and does not include published layout or pagination.

Citation Details

Citation for the version of the work held in 'OpenAIR@RGU':

BASS, J. M., 2013. Agile method tailoring in distributed enterprises: product owner teams. Available from *OpenAIR@RGU*. [online]. Available from: <http://openair.rgu.ac.uk>

Citation for the publisher's version:

BASS, J. M., 2013. Agile method tailoring in distributed enterprises: product owner teams. In: ICGSE 2013: 2013 IEEE 8th international Conference on Global Software Engineering. 26-29 August 2013. Los Alamitos, CA: IEEE. Pp. 154-163.

Copyright

Items in 'OpenAIR@RGU', Robert Gordon University Open Access Institutional Repository, are protected by copyright and intellectual property law. If you believe that any material held in 'OpenAIR@RGU' infringes copyright, please contact openair-help@rgu.ac.uk with details. The item will be removed from the repository while the claim is investigated.

Agile Method Tailoring in Distributed Enterprises: Product Owner Teams

Julian M. Bass

*School of Computing Science and Digital Media
Robert Gordon University
Aberdeen, UK*

j.m.bass@rgu.ac.uk

Abstract—This paper explores practitioner descriptions of agile method tailoring in large-scale offshore or outsourced enterprise projects. Specifically, tailoring of the product owner role is discussed. The product owner identifies and prioritizes customer requirements. But in globalized projects, the product owner must reconcile large numbers competing business interests and generate prioritized requirements for many development teams. The study comprises 8 international companies in London, Bangalore and Delhi. Interviews with 46 practitioners were conducted between February 2010 and May 2012. A grounded theory approach was used to identify that product owner teams comprise nine roles: Groom, Prioritizer, Release Master, Technical Architect, Governor, Communicator, Traveler, Intermediary and Risk Assessor. These product owner roles arbitrate between conflicting customer requirements, approve release schedules, make architectural design decisions, provide technical governance and disseminate information across teams. Understanding these roles will help agile coaches guide large scale agile teams.

Index Terms—agile software development, enterprise systems, product owner.

I. INTRODUCTION

Problems with large scale software development projects can damage the reputation of well-known companies and challenge the ability of Governments to implement policy. In the UK for example, Santander UK plc had problems with an IT integration project during a take-over of a smaller bank [1]. The adverse publicity this incident attracted, was not restricted to technology news outlets [2]. Further, Government decisions to change policies in relation to taxation or social security systems often imply large scale software implementations. The ability of Governments to implement large scale software projects has been questioned. In the UK, concern has been expressed regarding the Government's ability to implement a new social security payment system. For example, Liam Byrne MP during questions to the treasury said “the Universal Credit is late and over budget; there is widespread unease surrounding the implementation of the £2 billion [US \$ 3.2 billion approx.] scheme's IT system” [3]. Thus project teams on large scale projects are under considerable pressure to deliver successful outcomes.

There is increasing practitioner interest in the adaptation of agile software methods to large, often geographically distributed, enterprise software development projects [4] [5] [6].

According to practitioners the three most important perceived agile principles are: (1) achievement of customer satisfaction through early and continuous delivery of valuable software, (2) business representative and development team members working together frequently throughout the project, and (3) face-to-face conversations are the most efficient way to convey information to, and within, the development team [7]. This is in broad agreement with proponents of Agile methods who argue that they improve team moral, resulting in enhanced productivity and improved responsiveness to customer needs, resulting in better software quality [8]. Empirical research suggests that while agile methods do improve job satisfaction, productivity and customer satisfaction there can be challenges with adoption for large and complex projects [9].

Efforts by the software engineering research community to understand agile software engineering in practice has tended to conflate investigation of smaller companies and larger companies as if the pressures they face were the same, for example see [7] and [10]. The research presented here contributes to the literature on tailoring agile methods for use in large geographically distributed enterprise software projects, including five CMMI Level 5 certified offshore software development vendors. The overall research question for the study is “how do practitioners describe the tailoring of agile method roles and practices in large-scale geographically distributed enterprise software projects.” In particular, this paper focuses on the research question “how do practitioners describe enhancement and expansion of the product owner role to meet the needs of large-scale geographically distributed enterprise software projects.”

The paper is structured as follows. An overview of agile software development methods is provided, concentrating first on scrum, which was the most widely used technique in the surveyed companies, and then the product owner role. Next is a discussion of the research methods adopted, including the research sites, data collection and data analysis techniques used. The findings are provided showing how, in practice,

companies scale agile methods to large enterprise projects. The findings address four aspects of agile method scaling: large team size, geographical distribution, technical complexity and domain complexity. Next, there is a discussion of the findings. Finally, conclusions and suggestions for future work are presented.

II. AGILE METHODS

There are a range of software development methods that can be regarded as being agile including: Dynamic Systems Development Methods (DSDM) [11], Feature Driven Design [12], Crystal [13], Scrum [14], Extreme Programming (XP) [15] and more recently Lean Software Development [16].

XP has been associated with engineering practices such as test-driven development and pair programming [15]. Pair programming, where developers work together in a manner analogous to a pilot/co-pilot configuration, has been studied extensively [17] [18] [19].

In contrast with the engineering focus of XP, scrum has tended to focus on the orchestration and management of agile development [20]. Scrum, most commonly adopted by the company teams investigated in this study, is briefly outlined here. Scrum proposes short, focused periods of development called sprints. Typically lasting between two and four weeks. Software requirements are captured, analyzed and prioritized in the form of brief textual, non-technical descriptions called user stories. The user stories are prioritized, before the start of each sprint, by a product owner who represents the strategic needs of the client. Stakeholders, including the development team members and the product owner, work together to create work estimates for each user story using a consensus-based scoring technique. The development team members decompose each user story into the constituent technical tasks necessary for implementation at the start of each sprint. Project team members communicate using a daily coordination meeting, the eponymous scrum. The scrum is traditionally conducted standing up, in a conscious effort to minimize the duration of the meeting. Team members are required to answer three questions: (1) “what have you done since the last meeting?”, (2) “what will you do between now and the next meeting?” and (3) “what impediments that prevent your progress have you encountered or created for others?”

Scrum teams are self-organizing, since team members collaborate to develop work estimates and can select user stories for implementation within the current sprint [21] [22]. Scrum emphasizes incremental software development using a “feature” team structure [14]. Feature team members tackle holistic development of end-to-end user story functionality [12]. This contrasts with traditional approaches that tend to hierarchically decompose systems into specialist architectural components such as user interface, business logic or persistence layer elements.

An earlier study involving the author [23] focused on software development teams located in small and medium sized enterprises (companies with less than 250 employees and

a turnover \leq € 50 m [24]). That study [23] convincingly illustrated a very different business context and market pressures from the large enterprise environment. Some agile proponents argue that agile methods must be holistically implemented in their entirety in order to achieve full benefits (for example [15] p. 149). However, the findings presented here suggest this is not always desirable in large projects.

A. Enterprise Agile

The challenges of scaling agile methods to large international projects have received attention from practitioners [4] [5] and [6]. A project case study suggests success is possible using scrum in a large-scale project setting [25]. A scrum of scrums approach has been advocated to scale to a large team size [4]. Several scrum teams are formed, each with a scrum master in the usual way. Here, each scrum team comprises 7-12 developers. Daily coordination meetings are held within each scrum team in the usual way. However, in addition the scrum masters attend a coordination meeting across the teams (the scrum of scrums). The scrum of scrums is used to tactically manage and coordinate the progress of iterations through the various scrum teams. During the scrum of scrums each scrum master will report: (1) “what my team has done since the last meeting?”, (2) “what my team will do between now and the next meeting?” and (3) “what impediments that prevent your progress has my team encountered or created for others?”

In the related area of global software development, where geographical distribution is often - though not always an indicator of large scale - a meta study of research papers has been conducted [26]. The meta-study suggests the most researched agile practices are (1) continuous integration, (2) stand-up meetings, (3) pair programming, (4) retrospectives, (5) scrum of scrums, and (6) test-driven development [26]. To mitigate geographical distribution, multiple modes of communication support are available such as phone, web camera, teleconference, video conference, web conference, net meeting, email, shared mailing lists, instant messaging and short messaging service. A variety of collaboration techniques are available to scrum teams including: visits and periods of co-located working, unofficial meetings, training activities and distributed documentation support tools to help alleviate sociocultural distance [27].

B. Product Ownership

Scrum only defines three roles in its agile processes: the team, the scrum master and product owner [14]. The inclusion of product ownership in this short list indicates the central role product ownership plays in the overall software development processes. The product owner is central to the communication between customer and development teams [28]. The product owner is responsible for developing and maintaining the product backlog; the list of user stories that define the requirements for the project. In XP, the product ownership role is conducted by the on-site customer; a client representative

that is available to the team on a full-time basis [15]. As agile methods scale to larger projects it becomes harder to image a senior executive being available to development teams full-time.

III. METHOD

This qualitative study of software engineering practice comprised 8 international companies and interviews with 46 practitioners.

A. Research Sites

The companies were selected from a population of large enterprises engaged in (typically both) on-shore and off-shore software development projects. The companies chosen had head office in Germany, India and USA. The turnover of the two largest companies are almost €8 billion and over US \$1.5 billion. The interviews were conducted in Bangalore, India (January 2010 and April 2011), London, UK (February 2010) and Delhi, India (May 2012). Altogether, there were 46 practitioner interviews at 8 international companies, as shown in Appendix 1. The companies investigated were involved in either off-shoring (companies B and F) or out-sourcing (companies A, C, D, E, G and H). Off-shoring is typically motivated by a desire to access and cultivate world-wide talent pools. Both off-shoring and outsourcing offer lower cost resource pools. For example, Company B is a household name in the Internet business sector. Company B retains an in-house development capability in California but has built-up a large development team in India (as well as other territories) to reduce costs while attracting a range of specialist skills. While Company F, with broad interests in the industrial products space has headquarters in Europe but also has research as well as development centers in India and elsewhere. Work is allocated according to the concentration of expertise into specialist groups within the enterprise (to avoid duplication of competencies throughout the organization). While the IT Services companies (companies A, C, D, G and H) are well-known vendors in the world-wide outsourcing sector. Selection of the companies and research study participants was through a snowball sampling technique ([29] pg. 176; [30] Pg. 37). Initially, professional contacts provided access to the first study participants. Those participants then were able to enable access to other development teams and companies. Early phases of the study focused on participant breadth, at Companies A, B, C, F and G, getting access to a range of project teams and stakeholders with different perspectives. Participants ranged from Company C, one of whose defining characteristics is their adherence to agile methods. On the other hand, agile skeptics with negative experiences to report were found at Companies E and G. The later phases of the study focused on depth by targeting participants with a range of stakeholder roles in the same company or project. Here the interviews at Company H, and Companies D and E provided developer, QA, project management and corporate-level perspectives. This in-depth phase of the study is an implementation of intensity sampling

([29] pg. 171). Thus, selecting research participants that provide a wide range of perspectives using snowball sampling in an early phase and then using intensity sampling in a subsequent phase is a combination sampling approach which provides an element of methodological triangulation to the sample selection. Combination participant sampling provides insights into both the current status of the research problem and the motivation that underlies these practices. These motivations for the use of practices are difficult to obtain using large scale survey methods, for example.

B. Data Collection

A range of documentary sources were used to inform the study. Commercially confidential corporate agile development method process guidelines were studied. These guidelines outline corporate agile practice roles, policies and recommendations. Some project documentation has also been investigated, such as design and architecture documents. Marketing materials such as publicly available and web hosted white papers, technical reports, case studies and descriptions of vendor capabilities designed to inform potential customers, were also reviewed.

Site visits enabled observations of working areas and working practices. Secure development team work environments were visited. Coordination meetings (stand-up meetings) were observed in real-time for both co-located and distributed scrum teams. The work environment arrangements for distributed scrum coordination meetings using both video- and audio-conferencing technologies were investigated. A range of informal, sometimes off-site, discussions with executives, project management and development team members were conducted.

Face-to-face recorded interviews were conducted with the 46 practitioner interviewees. Recordings were transcribed. An open-ended interview guide approach was used to structure interviews, an example interview guide is included in Appendix 2. Respondents were given opportunities to raise any topics, issues and concerns of their choosing outside the scope of scripted interview questions. Interviews were typically conducted on company sites using small meeting rooms exclusively booked for interview purposes.

C. Data Analysis

Both the audio interviews and associated written transcripts were initially carefully reviewed. The transcript text was imported into a qualitative data analysis software tool, in this case Nvivo V9 [31]. Initially key points were identified within the interview data. The key points were coded and compared within and between interviewees. An iterative data analysis approach was used refine coding categories, as shown in Figure 1. Thus, the key points were combined to create concepts which were then themselves coded, listed and compared within and between interviewees. Concept categorization was used to organize the large volume of data into a taxonomy. The categorization forms the basis of the grounded theory [32] [33].

IV. FINDINGS

Early proponents of agile methods advocated small, self-organizing and co-located teams. However, in enterprise settings, large work volumes, short deadlines and entrenched organizational structures result in tailored agile approaches. The contribution of this paper is practitioner descriptions of product owner role tailoring. The research identifies the emergence of product owner teams to allow agile methods to scale-up to large international projects. Nine functions of the product owner role are identified.

First, let's consider the absence of a product owner. These issues can be illustrated by drawing on Company E which conducts (non-agile) customization projects on client sites and uses scrum with an out-sourced development partner for its own product development. For example, "there are client representatives at [major international bank, UK] they are a little ill-defined what they are actually doing. But there isn't a stakeholder that comes and makes any real decisions on the project" (Project Manager, Company E, February 2010). This project manager continues "there's a lot more syndication that actually goes on in coming to a conclusion" (Project Manager, Company E, February 2010). Here this project manager with 30+ years of IT industry experience uses the word syndication as a euphemism for equivocation. This illustrates that, from a development team perspective, it is difficult to get decisions made about requirements and their priorities in the absence of a product owner. At the same bank

"their governance is actually done on [a] biweekly basis and then, there's another level of governance which is done on a four- to six-week basis... they have traffic light system of reporting on a project, but they have a very peculiar set of rules to actually how those traffic lights can go to amber or, very rarely, can go to red, which is very surprising... I would have all the projects at red at the moment but they're either green or possibly only just about going amber" (Project Manager, Company E, February 2010).

So, from an enterprise perspective it is difficult to hold project teams to account in terms of scope, budgets and deadlines.

Enterprises tend to expand the number of job titles beyond the roles defined in standard agile literature. For example "There is a team in India of 20 people with a manager. There is a product manager as well as there is project manager and product owner and there is me overseeing" (Engineering Director, Company E, February 2010). Product owner job functions need to be clarified because of the lack of standardization between job titles and product owner activities. Thus, product owner functions may be conducted by staff members with various job titles.

What does a product owner actually do? From one perspective "product [scope], the design, the requirements, the discussion with the business side from the customer, that is all done through product owner" (Engineering Director, Company E, February 2010). The nine roles identified in the research are described below.

A. Groom

Product grooming "is a list of... requirements or features... So it is a product owner's responsibility to make sure product backlog should be continuously evolving (Program Head, Company H, May 2012). Simply put, "I'd be ready with my sprint backlog. What are the things we are going to do? Maybe a few additional features and some backlog from the previous [sprint] where we have received feedback from the client" (Company G, Engagement Manager, April 2011). Product owners reconcile conflicting priorities

"there is a product [management organization in California]. So different regions like APAC and Europe and North America come with some requirements, this is what they want to get done. And they come to the product manager... who keeps on collecting the requirements, prioritizes them. There's a six-monthly road map discussion... where they have a high-level look at the things we are trying to achieve this year. They act as a channel between the development team and the market so, for us, they are the customers." (Engineering Manager, Company B, Jan. 2010).

The product owner needs to interact with customers to gather the requirements "our Product Owner, luckily, is having a very rich experience of interaction with customers" (Scrum Master, Company D, Jan. 2010). However, simply compiling a list of requirements is not sufficient.

B. Prioritizer

The product owner is also responsible for prioritizing requirements in the product or sprint backlogs. For example, "we keep re-triaging [the backlog] because we have too many mandatories" (Engineering Director, Company E, February 2010). And also "if [requirements]... are not sequenced properly as for the priority, legality. So whatever we are delivering will not give that priority value to the end user or the business user" (Program Head, Company H, May 2012). Thus, it is the product owner's responsibility to identify and reconcile the needs of different parts of the client organization. Product owners become experienced in assessing and prioritizing contrasting the needs of different segments of the customer-base. Thus, they must have sufficient stature and seniority to perform that conflict resolution function.

In some cases, such as enterprise products simultaneously aimed at a large number of end-user territories, project teams can be more directly involved in systematic recording of

customer requirements “we call it a requirement template approach ... so this particular template recognizes what is the basic requirement which all of [the client territories] will agree. [each client territory in turn] will say 'okay this is less priority for me' or there'll be some variation. So against each [requirement template] there is some line item that can be done, okay we want this piece” (Practice Head, Company A, Jan. 2010). This particular project involved a simultaneous product launch in 17 different countries.

C. Release Master

A release master manages and approves release schedules. For example, “a release plan is prepared. It is sent for approval to the product owner... once it's approved by the product owner then we stick to that” (Architect, Company D, Jan. 2010). Backlog grooming, requirements prioritization and release planning are standard scrum practices. What other product owner activities have practitioners described? Six other product owner roles are identified, as follows.

D. Technical Architect

To design, implement and disseminate a reference architecture a technical role for the product owner is required. The product owner coordinates technical and architectural policies between the scrum teams. This will include sufficient documentation and illustrative source code to ensure project teams can understand and follow the guidelines. The product owner must disseminate the reference architecture to teams. The project team may legitimately find anomalies and ambiguities that the product owner should provide guidance on.

A significant amount of up-front work should be done by the product owner prior to scrum teams working on increments. These architectural design decisions can often be performed using broad statements of functional requirements. However, non-functional requirements have more bearing on the development of a reference architecture.

This is a technical thought leadership and coordination role. Architectural styles are developed and disseminated by the product owner in compliance with corporate architecture guidelines “so we have a [corporate-wide] council, architecture council” (Architect, Company D, Jan. 2010). The absence of a product owner does not mean these decisions are not made. On the contrary, the decisions are made by teams in an uncoordinated manner. This results in divergent architectural styles in different parts of the program.

There are tensions between performing the architecture design within the self-organizing scrum teams or as a separate function. For example “in ideal waterfall model, as I said earlier we have separate architects, separate designers, separate developers, then testing team's separate. In a scrum team they're all working together and there is no differentiation” (Engagement Manager, Company G, April 2011). The problem is that “I have observed that people who are working in waterfall model were not taking responsibility [for delivering a

quality product on time]” (Engagement Manager, Company G, April 2011).

The self-organizing teams must also learn about the reference architecture “this requires every individual to put a lot of effort into understanding the R&D and understanding the architecture... this responsibility is shared to the team not just lying with only Product Owner and Scrum Master” (Scrum Master, Company D, Jan. 2010).

E. Governor

A large corporate e-Commerce website is a corporation's “public face, in 26 languages, across the world, right? That website needs to have technical governance on how changes are made.” (Lead Architect, Company H, May 2012). To achieve that technical governance “you need technical product ownership, especially in large programs right? If you are, [working on] something like [MajorAirline.com], right?” (Lead Architect, Company H, May 2012).

The product owner provides a technical governance framework to project teams working on a program. The product owner will liaise with governance structures such a technical oversight committees and corporate architecture groups. The product owner will ensure selection of common tools and technologies for the project. This role will ensure projects within a program share an appropriate technology infrastructure.

Needless to say the product owner must be aware of corporate governance policies and strategic directions. Any proposals made by the product owner should comply with these governance directives.

F. Communicator

Geographical distribution is not an attractive attribute for a project team, but is a feature of the globalized nature of enterprise software development. At a senior level there is a view in some quarters that geographical distribution itself is not the major problem to solve. “I think we are now very mature. Initially we thought that our biggest challenge would be the geographical distance, some people here, some people there. So that is definitely a challenge but that is I would say the easiest to fix” (Chief Architect, Company H, May 2012).

Geographical distribution within the scrum team membership is discouraged in the companies investigated. Company B avoids geographically distributed scrum teams “most of the scrums we are the only ones, so we do our own things here [offshore]. Product managers and UI designers, they dial in from there [onshore]” (Engineering Manager, Company B, Jan. 2010).

A compromise of adding a remotely located technical specialist to the co-located team is sometimes helpful “We had one engineer working from [California] join us, she used to directly dial in to our sprint planning meetings, retrospection, everything” (Engineering Manager, Company B, Jan. 2010). This is onerous to that individual where the time zone difference is large (since the remote technical specialist usually

has to adopt working patterns to suite the rest of the co-located development team). Sometimes “participation in the stand-up is extremely tough because you do it in a room with a speaker phone and five people talking about different things, it becomes very tough for a [remote] person to understand what’s going on and contribute ... we found it was better to have one-on-one call with her” (Engineering Manager, Company B, Jan. 2010).

Development team working patterns can be adapted to increase the overlap between office hours onshore and offshore. Where the time zone difference is not too great, offshore scrum teams can work into the evening “our normal shift is 11:00[am]-8:00[pm]” (Software Engineer, Company H, May 2012) and from a different project team “we are working 1pm to 10pm” (Project Manager, Company H, May 2012). This is an example where time zone differences between Europe and India are accommodated by shifting work patterns (in India) to increase the number of overlapping working hours. Offshore staff members can sometimes expect to receive some extra payments and free dinners when working this type of pattern. Transportation is often provided to staff what ever shift patterns are being worked. Working after 12 midnight is not popular with respondents because it disrupts weekend social and family life. Whereas early evening working can be accommodated without too much disruption of life outside work.

A more common arrangement is a co-located offshore scrum team working with a remote product owner “I’ve got a product owner... based in New Zealand” (Engineering Director, Company E, Jan. 2010). This reflects a classic onshore/offshore model. Again, mutual meetings timings must be found, which is challenging with a large time zone differences. Video (and less commonly audio) conferencing is used to conduct daily scrum coordination meetings. These are sometimes conducted as stand-up meetings (using video conferencing technologies).

Communication challenges include “practical difficulties making sure people can interact; such as booking video conference rooms at both ends, Internet connectivity limitations on video conferencing; tendency to underestimate investment required in travel and connectivity” (Delivery Manager, Company H, May 2012).

G. Traveller

Product owner teams have staff members onshore for discussions with clients and off-shore for disseminating information to development teams. The proxy product owner will usually spend time (1-3 months, depending on the scale of the project) on the client site at the start of the project becoming familiar with any special features of the client's requirements. The breakdown of product owner teams between onshore and offshore activities needs to be determined “if you are sort of 70 people working from the offshore there are 5 to 6 guys at the onsite all, every day at the same time zone [as the client]” (Practice Head, Company A, January 2010). The Traveler is important for supporting development teams “we have a field person at the customer site who can answer

queries... They’re the ones who interact with the customer directly” (Architect, Company D, Jan. 2010).

Enterprises can also adopt a range of solutions in terms of locating teams “One scrum team is based onshore completely, the other scrum team was sent onshore for a month, had two iterations over there, and then came back and started doing iterations here, and then the third team is a distributed team that has been formed. So everybody is getting the exposure of working directly with the product owners and the business analysis is what we did to mitigate that risk” (Delivery Manager, Company H, May 2012).

So, product owners act as a bridge between onshore and offshore. They can be based with clients or they can travel to clients. Alternatively, and at higher cost, entire teams can travel to clients to co-locate with product owners.

H. Intermediary

The product owner is supplemented with an intermediary from within the development team to mitigate domain complexity, the “proxy product owner [is] an extension of the product owner, the product owner’s availability or understanding of the off-shoring process being limited” (Delivery Manager, Company H, May 2012). The intermediary will need to have extensive experience of the system business domain.

In Company B “we have some kind of shared product ownership very limited that is done by [local Product Manager]. But it is mostly on requirements... we have like four/five different conferences [conference calls] with different stakeholders. So, like product managers and some other folks, engineering folks, and some people who are working on performance, for example.” (Engineering Manager, Company B, Jan. 2010)

I. Risk Assessor

Enterprises routinely conduct risk management to assess technical complexity and potential shortcomings in the development team skills and capabilities. Product owners perform

“risk management, if something is seen as technically very complex, it will come up as part of the risk [assessment] of that particular project. Then, you will have to see how you mitigate that risk; if it requires support from a centre of excellence within [the company] or stronger [staff technical] profiles to be a part of that project then we will do that.” (Delivery Manager, Company H, May 2012).

The risk mitigation might include “assigning people called Technical Analysts, I mean SMEs [subject matter experts] with respect to their technical domain understanding” (Program Head, Company H, May 2012).

Technical Specialists assigned to the scrum team can provide insight into managing complexity. For example, where multiple programming languages are being used the interfaces

between language components require special skills. Similarly interfaces to external systems and credit card payment gateways require access to a technical specialist. Intelligent choice need to be made between sharing a technical specialist between one or more sprint teams. Where the technical complexity is affecting the work of and entire sprint team, then a 100% assignment of a full-time technical specialist makes sense. Where the technical complexity affects some aspect of the work of the sprint team, then access to a part-time technical specialist will be sufficient.

V. DISCUSSION

The findings presented here support previous research that suggests agile methods can be scaled to large projects [34] and used in distributed software development settings [35]. The closest context to our study was presented in [36]. That study explored scrum developers working together yet based in USA, Canada and Russia. These previous studies tended to emphasize tailoring agile method practices rather than roles.

One of the companies studied in [36] centralized and co-located scrum masters, product owners and architects. Companies A, B, D, E and H in the findings presented here had a distributed model of product ownership. Product owners are required to undertake new activities to manage geographically distributed clients and development teams. The product ownership team concept emerges, in part at least, from the need to manage product owner and development team distribution.

The respondent from Company F described a highly regulated healthcare instrument project. The use of agile methods in highly regulated environments has been investigated in [37]. Again, however there was little specific discussion of the product owner role in [37].

This research has focused on practitioner descriptions of activities within the product owner role. This approach was similar to [10] that focused on roles within self-organizing teams. My contribution is to articulate the activities undertaken by product owners. I argue that product owner teams are required to manage the scale and complexity of product owner activities in globalized software development projects.

VI. LIMITATIONS

There are three tests for establishing the quality of descriptive empirical social research: construct validity, external validity and reliability [38].

Construct validity can be ensured through using multiple sources of evidence. This has been achieved through conducting studies at eight companies with large studies at Companies H and E (along with their offshore service provider Company D). Multiple sources of evidence within these large studies has included corporate, program and project level management as well as development team members.

External validity is achieved through replication of studies. Replication here is at eight companies and the range of project stakeholder respondents. However, the findings and conclusions presented here should not be generalized to small

and medium sized companies. Smaller companies work under profoundly different commercial pressures with different quality assurance responsibilities.

Reliability is developed through use of the interview guide. This ensures consistency in the data collection. The interview guide is reproduced in Appendix 2.

VII. CONCLUSIONS AND FURTHER WORK

This paper has explored practitioner descriptions of agile method tailoring in large-scale enterprise software development. Specifically product owner role tailoring has been investigated.

Nine product ownership functions have been described that are used to scale agile methods to large projects: groom, prioritizer, release master, technical architect, Governor, communicator, traveller, intermediary and risk assessor.

The groom gathers requirements from business clients. The prioritizer ensures requirements bring value to the business. The release master manages and approves release plans. Technical architects provide architectural coordination on large projects. This architectural coordination is achieved by using reference architectures to guide and support self-organizing scrum teams. Governors are required to ensure project compliance with corporate guidelines and policies. The communicator bridges onshore and offshore geographical distribution. Travellers spend time onshore at client sites gathering first-hand knowledge of the client's needs. The intermediary acts as an interface to senior executives driving large scale development programs disseminating domain knowledge to teams. The risk assessor evaluates technical complexity.

These product owner activities provide a layer of governance to agile methods in the CMMI Level 5 certified companies investigated. Further, these activities can be performed by members of a product owner team. Thus, product owners can devolve selected activities to technical and remotely located colleagues. The product owner team becomes a key tool in tailoring agile methods for large scale distributed projects.

Further work, in collaboration with Company H, will explore agile methods and CMMi Level 5 certification. That paper considers how agile projects gather and provide the evidence required by certification authorities.

Also, it is puzzling that engineering practices from XP such as pair programming and test driven development are not more widely used. Further investigation would help understand the lack commitment in these large enterprises to such practices.

ACKNOWLEDGMENT

I am grateful to all the companies and interviewees who were generous enough to contribute their time and resources to participate in this research. Thanks also to the current and former students of the Executive MBA at the Indian Institute of Management, Bangalore who helped identify target companies. The research benefited in part from travel funding from the UK

Deputy High Commission, Bangalore, Science and Innovation Network the Institute for Innovation, Design & Sustainability (IDEAS) at Robert Gordon University, UK, and accommodation and sustenance from Company H during the data collection visit to Delhi, India.

REFERENCES

- [1] Computer Weekly, ‘Santander migration glitch affects Alliance & Leicester customers’ August 12, 2010, <http://www.computerweekly.com/news/1280093541/Santander-migration-glitch-affects-Alliance-Leicester-customers>
- [2] BBC News, ‘Bank merger account glitches’, August 27, 2010 <http://news.bbc.co.uk/1/hi/programmes/moneybox/8946199.stm>
- [3] House of Commons, Hansard, September 11, 2012, Column 117, <http://www.publications.parliament.uk/pa/cm201213/cmhansrd/cm120911/debtext/120911-0001.htm>
- [4] D. Leffingwell, *Scaling Software Agility: Best Practices for Large Enterprises*, Boston, MA: Addison Wesley, 2007.
- [5] C. Larman and B. Vodde, *Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum: Successful Large, Multisite and Offshore Products with Large-scale Scrum*. Addison Wesley, 2008.
- [6] S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner’s Guide to Agile Software Delivery in the Enterprise*. IBM Press, 2012.
- [7] S. de Cesare, M. Lycett, R. D. Macredie, C. Patel, and R. Paul, ‘Examining Perceptions of Agility in Software Development Practice’, *Communications of the ACM*, vol. 53, no. 6, pp. 126–30, Jun. 2010.
- [8] ‘Agile Alliance’. [Online]. Available: <http://www.agilealliance.org/>. [Accessed: 25-Sep-2011]
- [9] T. Dyba and T. Dingsoyr, ‘What Do We Know about Agile Software Development?’, *Software, IEEE*, vol. 26, no. 5, 2009, pp. 6-9.
- [10] R. Hoda, J. Noble, and S. Marshall, ‘Organizing self-organizing teams’, in *Software Engineering, 2010 ACM/IEEE 32nd International Conference on*, 2010, vol. 1, pp. 285–294.
- [11] J. Stapleton, *DSDM: Dynamic Systems Development Method*. Harlow, England: Addison Wesley, 1997.
- [12] P. Coad, E. LeFebvre, and J. De Luca, *Java Modeling in Color*. Englewood Cliffs, NJ: Prentice Hall, 1999.
- [13] A. Cockburn, *Agile Software Development*. Reading, MA: Addison Wesley, 2001.
- [14] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Upper Saddle River, NJ, USA: Prentice Hall, 2001.
- [15] K. Beck and C. Andres, *Extreme Programming Explained*, 2nd ed. Addison Wesley, 2004.
- [16] X. Wang, K. Conboy, and O. Cawley, ‘Leagile Software Development: An Experience Report Analysis of the Application of Lean Approaches in Agile Development’, *The Journal of Systems and Software*, vol. 85, no. 6, 2012, pp. 1287–1299.
- [17] V. Balijepally, R. Mahapatra, S. Nerur and K. H. Price, ‘Are Two Heads Better Than One for Software Development? The Productivity Paradox of Pair Programming’, *MIS Quarterly*, Vol. 33 No. 1, 2009, pp. 91-118.
- [18] J. E. Hannay, E. Arisholm, H. Engvik, and D. I.K. Sjøberg, ‘Effects of Personality on Pair Programming’, *IEEE Tans. Software Engineering*, Vol. 36, No. 1, 2010, pp. 61-80.
- [19] K. M. Lui, K. C. C. Chan, and J. T. Nosek, ‘The Effect of Pairs in Program Design Tasks’, *IEEE Tans. Software Engineering*, Vol. 34, No. 2, 2008, pp.197-211.
- [20] K. Schwaber, *Agile Project Management With Scrum*. Redmond, WA, USA: Microsoft Press, 2004.
- [21] M. Cohn, *Succeeding with Agile: Software Development Using Scrum*, Upper Saddle River, NJ: Addison-Wesley Professional, 2009.
- [22] R. Hoda, J. Noble and S. Marshall, ‘Developing a Grounded Theory to Explain the Practices of Self-Organizing Agile Teams’, *Empirical Software Engineering Journal (ESE)*, Volume 17, Number 6, Pages 609-639, 2012.
- [23] R. Gittins, J. Bass, and S. Hope, ‘A Comparison of Software Development Process Experiences’, in *Extreme Programming and Agile Processes in Software Engineering*, LNCS Vol. 3092, J. Eckstein and H. Baumeister, Eds. Springer Berlin / Heidelberg, 2004, pp. 231–236.
- [24] ‘What is an SME? - Small and medium sized enterprises (SME) - Enterprise and Industry’. [Online]. Available: http://ec.europa.eu/enterprise/policies/sme/facts-figures-analysis/sme-definition/index_en.htm. [Accessed: 11-Aug-2012].
- [25] M. Paasivaara, S. Durasiewicz, and C. Lassenius, ‘Using scrum in a globally distributed project: a case study’, *Software Process: Improvement and Practice*, vol. 13, no. 6, 2008, pp. 527–544.
- [26] S. Jalali and C. Wohlin, ‘Agile Practices in Global Software Engineering - A Systematic Map’, in *IEEE 5th International Conference on Global Software Engineering (ICGSE)*, 2010, pp. 45–54.
- [27] E. Hossain, M. A. Babar, and H. Paik, ‘Using Scrum in Global Software Development: A Systematic Literature Review’, in *IEEE Fourth International Conference on Global Software Engineering (ICGSE)*, 2009, pp. 175–184.
- [28] R. Hoda, J. Noble and S. Marshall, ‘The Impact of Inadequate Customer Involvement on Self-Organizing Agile Teams’, *Journal of Information and Software Technology (IST) Volume 53, Issue 5, May 2011, Pages 521-534*.
- [29] M. Q. Patton, *Qualitative Evaluation and Research Methods*, 2nd Edition, SAGE Publications, Newbury Park, 1990.
- [30] M. B. Miles and A. M. Huberman, *Qualitative Data Analysis*, SAGE Publications, Newbury Park, 1984.
- [31] NVivo, Help, 2012, http://help-nv9-en.qsrinternational.com/nv9_help.htm
- [32] B. G. Glaser and A. L. Strauss, *Discovery of Grounded Theory: Strategies for Qualitative Research*, Piscataway: Aldine Transaction, 1999.
- [33] J. M. Corbin and A. C. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 3rd ed. Thousand Oaks: SAGE Publications, 2008.
- [34] D. J. Reifer, F. Maurer, and H. Erdogmus, ‘Scaling agile methods’, *Software, IEEE*, vol. 20, no. 4, pp. 12 – 14, Aug. 2003.

APPENDIX 1

TABLE I. PARTICIPATING COMPANIES, INDUSTRY SECTORS AND INTERVIEWEE JOB TITLES

Company	Sector	Interviewee Job Titles	Projects and Programs
Company A, Bangalore	IT Service Provider	Program Manager Senior Project Manager Team Member	Customer Relationship Management
Company B, Bangalore	Internet	Engineering Manager Product Manager (interviewed twice, Jan 2010 and April 2011)	Web Mail Web Calendar
Company C, Bangalore	Software Service Provider	Development Manager	Rail Booking
Company D, Bangalore (Offshore Provider to Company E)	Software Service Provider	Project Manager Product Owner Scrum Master (3) QA Lead Team Member	Marketing Campaign Management Customer Relationship Management
Company E, London	Enterprise CRM	Program Manager Project Manager Director of Engineering	Banking Marketing Campaign Management Customer Relationship Management
Company F, Bangalore	Industrial Products	Scrum Master	Healthcare Instruments
Company G, Bangalore	IT Service Provider	Engagement Manager	Media Entertainment
Company H, Delhi	IT Service Provider	Chief Technology Officer Corporate Lead Architect General Manager Human Resources Delivery/Program Manager (3) Project/Senior Project Manager (3) Scrum Master (2) Technical Analyst/Consultant/Specialist (6) Test Analyst (2) Business Analyst Software/Senior Software Engineer (7)	Airline Customer Service Flight Booking

- [35] B. Ramesh, L. Cao, K. Mohan, and P. Xu, 'Can distributed software development be agile?', *Commun. ACM*, vol. 49, no. 10, pp. 41–46, Oct. 2006.
- [36] J. Sutherland, A. Viktorov, J. Blount, and N. Puntikov, 'Distributed Scrum: Agile Project Management with Outsourced Development Teams', in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, 2007*, p. 274a.
- [37] O. Cawley, X. Wang, and I. Richardson, 'Lean/Agile Software Development Methodologies in Regulated Environments State of the Art', in *Lean Enterprise Software and Systems*, vol. 65, P. Abrahamsson and N. Oza, Eds. Springer Berlin Heidelberg, 2010, pp. 31–36.
- [38] R. K. Yin, *Case Study Research: Design and Methods*, 4th ed. Thousand Oaks, California: Sage Publications, Inc, 2009.

APPENDIX 2

A. Motivation and Purpose of Research

I want to ask you about your experience of geographically distributed agile software development projects. The research involves interviews with people doing a range of different roles and from companies with different development models.

I want to learn more about your views of agile processes. I am particularly interested to know what factors are affected by geographical location and separation. The purpose here is to try to understand the factors that affect project outcomes, successful or otherwise, so that we can try to learn for the future.

B. Ethical Commitments and Informed Consent

I want to ask you the following questions and tape record your answers. I will keep your responses absolutely confidential. Certainly nothing will be shared with any client companies. I do plan to publish interview extracts but I will make names and companies anonymous. Can I switch on the recorder?

C. Your Current Project(s)

- How many projects are you working on currently?
- What is (was) the title of your current (or most recent) project?
- What is the project management structure?
- How is the project organized geographically?
- How many people are in the project team?

D. Agile Practices

- What Agile practices do you advocate for offshore projects?
- What agile practices do you avoid or not recommend?

E. Requirements

- How are requirements decided and prioritized?
- How do user stories evolve over time?
- How do user stories move up or down the backlog?

F. Product Owner/Customer (POC)

- How do you represent the product development team within the client organization?
- How do you represent the client organization within the product development team?

G. Releases and Testing

- How do unit tested code become a release?
- How is user acceptance testing managed?
- How are bugs reported back, prioritized and fixed?

H. Challenges

- What challenges do you face in agile offshore projects?
- How have you tried to address these challenges?
- What challenges remain to be resolved?

I. Learning

- How does learning take place within the team?
- How does learning take place for you personally?

J. Any other comments

- Do you have any further comments in relation to geographically distributed agile development projects?

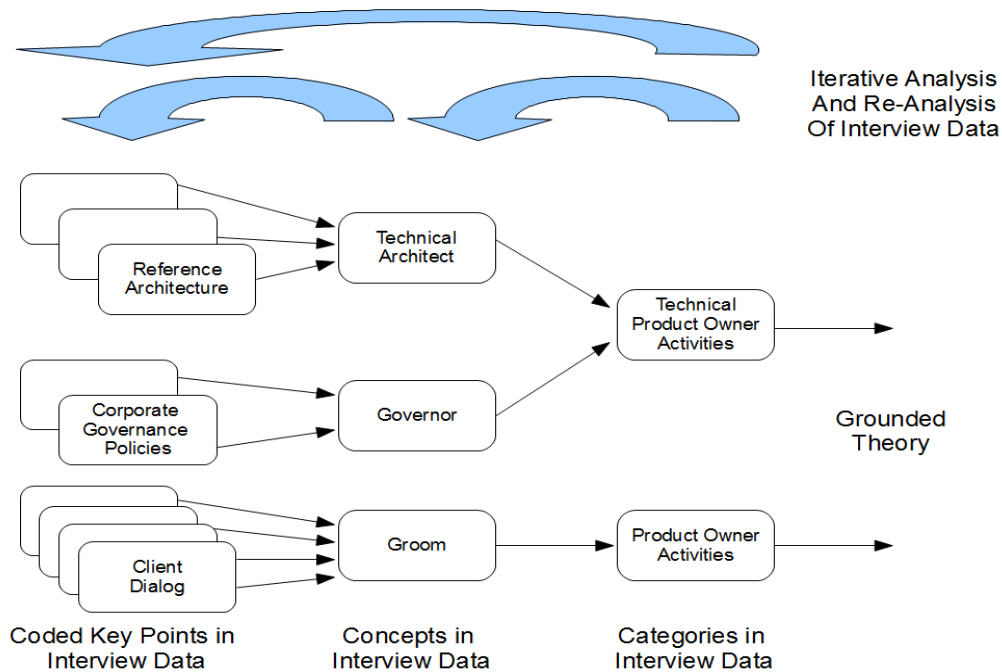


Fig. 1. Iterative analysis leading to grounded theory.