Activities in Scrum Master Teams: 
Process Tailoring in Large Enterprise Projects

Julian M. Bass
Robert Gordon University
Garthdee Road
Aberdeen, UK
Email: j.bass@computer.org

Abstract—This paper explores practitioner descriptions of agile method tailoring in large-scale offshore or outsourced enterprise projects. Specifically, tailoring of the scrum master role is investigated. The scrum master acts as a facilitator for software development teams, nurturing adherence to agile practices and removing impediments for team members. But in large projects, scrum masters work together in, often geographically distributed, scrum master teams. Scrum master teams use protocols to perform sprint planning, coordinate development effort and integrate code bases. The study comprises 8 international companies in London, Bangalore and Delhi. Interviews with 46 practitioners were conducted between February 2010 and May 2012. A rigorous grounded theory research method was used to identify that scrum master teams comprise 10 activities: scrum of scrums facilitator, sprint planner, integration coordinator, learning coordinator, heart beat monitor, process anchor, task estimator, impediment remover, spike process coordinator and test coordinator. This systematic description of activities in scrum master teams extends our understanding of practitioner perspectives on agile process tailoring in large enterprises. Grounded theory is an emerging research method in software engineering.

I. INTRODUCTION

This paper addresses the issues of development method tailoring by practitioners in large-scale enterprise software development projects. Managers and development team members are under intense pressure to successfully implement and deploy large-scale projects. Large development programs are challenging and suffer high risks of deadline slippage and cost overruns. For example, software development programs figure prominently in a May 2013 list of UK Government large projects “at risk” of failure [3]. Thus, implementation challenges with large software projects can impede government policy change goals. The term “Agile” is applied to a collection of software development methods including: Dynamic Systems Development Methods (DSDM) [37], Feature Driven Design [9], Crystal [10], Scrum [36], Extreme Programming [8] and more recently Lean Software Development [29].

Research shows that practitioners identify the three most important perceived agile principles as: (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 [12]. These practitioners are in broad agreement with proponents of Agile methods who argue that improved team moral, results in enhanced productivity and that improved responsiveness to customer needs, results in better software quality [1]. While empirical research suggests that agile methods do improve job satisfaction, productivity and customer satisfaction, there can also be challenges with adoption for large development programs [14].

This research offers novel contributions in two specific areas. Firstly, the research presented here contributes to the literature on tailoring agile methods for use in large software development programs, including five CMMI Level 5 certified offshore software development vendors. Here the phrase development program refers to a collection of strategically related, and often large-scale software projects. This contrasts with the term program used to describe a piece of computer software. The overall research question for the study is “how do practitioners describe the tailoring of agile method roles and practices in large-scale software development programs?” In particular, this paper focuses on the research question “how do practitioners describe the enhancement and expansion of the scrum master role to meet the needs of large-scale software development programs?”

Secondly, the research contributes to the emerging use of grounded theory in software engineering. Replicated sampling has been conducted in different research sites. Perspectives on the scrum master role have been triangulated by research participants with different large project stakeholder roles. Qualitative data has been rigorously analyzed using the grounded theory approach. Thus, this research further contributes an application of grounded theory in software engineering research. A motivation for the research is to reduce burnout and high levels of staff turnover by raising awareness of the complex range of scrum master activities. This understanding can be used to target training, clarify work allocation and provide management support for the activities scrum masters actually undertake.

The paper is structured as follows. Firstly, a discussion of related work in agile software development methods, concentrating on scrum and the scrum master role in particular, is presented. Then there is a discussion of the research methods adopted, including the research sites, data collection and data analysis techniques used. Next, is a presentation of the findings showing how, in practice, companies scale agile methods to large software development programs. Finally, there are conclusions and suggestions for future work.
II. RELATED WORK

Extreme Programming focuses on engineering practices such as test-driven development and pair programming [8]. Pair programming is where developers work together in a pilot/co-pilot configuration and has been studied extensively [6] [19] [28].

Scrum contrasts with the engineering focus of extreme programming, by focusing on the orchestration and management of agile development projects [35]. Scrum, which was the most commonly adopted by the company teams investigated in this study, is briefly described here. Some research has questioned the value of practices scrum in terms of their contribution to project success outcomes [15].

Managing requirements in large scale projects is complex and difficult [39]. User requirements for the software are often captured, analyzed and prioritized in the form of user stories. User stories are brief textual, non-technical descriptions that are readily understood by all project stakeholders. A product owner prioritizes user stories, before the start of each sprint, by carefully considering the strategic needs of clients. Stakeholders, including the scrum master and the development team members, work together to create estimates of the work required to implement each user story often using a consensus-based scoring technique. In the planning phase at the start of each sprint, the development team members decompose each user story into the various technical tasks necessary for implementation. Scrum teams, comprising 7-12 developers, are said to be self-organizing, since they create work estimates and individuals often select user stories (from the prioritized list provided by the product owner) for implementation within that particular sprint [11] [21]. Scrum emphasizes incremental software development using a multidisciplinary “feature” team structure [36]. Feature team members holistically develop end-to-end user story functionality [9]. This contrasts with traditional approaches which hierarchically organize team members around specialist architectural components such as user interface, business logic or persistence layer sub-systems.

In the past, agile proponents argued that agile methods must be implemented in their entirety to achieve their full benefits (for example [8] p. 149). However, the findings presented here suggest this is not always possible in large software development programs.

A. Enterprise Agile

As already discussed, the challenges of scaling agile methods to large international projects have received attention from numerous practitioners [27], [26], [4]. The simultaneous use of agile methods and plan-based methods in large enterprises is also receiving interest from researchers [38]. Large team size, complex business contexts and demanding time constraints can converge to cause a range of threats to productivity in agile projects [?] as the co-existence of plan-based and agile methods increases complexity and impedes involvement of business stakeholders [38]. Large scale projects therefore require a more disciplined approach to software development [5]. There is also evidence that large scale projects can exacerbate communication problems [33].

A scrum of scrums approach has been advocated to accommodate large team size [27]. Several scrum teams are formed, each with a scrum master in the usual way, and each scrum team comprises 7-12 developers. Daily coordination meetings are held within each scrum team, and 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 meeting, 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 progress my team has encountered or created for others.” Scrum of scrum meetings with too many participants can lack focus and relevance [31] and communication is improved where scrum of scrum meetings are organised around common interests and participant needs.

A meta study of research papers has been conducted in the related area of global software development [24]. In global software development, geographical distribution is often, though not always, an indicator of large scale. 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 [23]. Collaboration techniques use by scrum teams include: visits and periods of co-located working, unofficial meetings, training activities and distributed documentation support tools [20]. These communication and collaboration techniques help alleviate sociocultural distance within geographically distributed teams [13].

B. Scrum Master Role

Only three roles are defined in the scrum agile process, the: self-organizing development team, scrum master and product owner [36]. The inclusion of scrum master in this short list indicates its central role in the overall software development process. The scrum master is the primary interface between the product owner and the software development team [35]. The scrum master is responsible for facilitating the development, removing any impediments that the team encounters. The scrum master also ensures the team uses the full range of appropriate agile practices. As agile methods scale to larger projects multiple scrum teams, each with their own scrum master, must be coordinated.

This study investigates the scrum master role in large-scale development programs. The contribution of this paper is to systematically articulate the activities undertaken by scrum masters. It is argued that scrum master teams are required to manage the range and complexity of scrum master activities in globalized software development projects.

III. METHOD

Experimental [40] case study [34] and empirical [25] research methods have been advocated for software engineering research. Method selection depends upon the research question or hypothesis being tested. The grounded theory empirical research method is emerging as a rigorous approach to researching complex real work situations [21]. Grounded theory has been used to investigate activities in co-located self-organizing teams in small and medium sized companies.
TABLE I. PARTICIPATING COMPANIES, INDUSTRY SECTORS AND INTERVIEWEE JOB TITLES

<table>
<thead>
<tr>
<th>Company</th>
<th>Company Sector</th>
<th>Interviewee Job Titles</th>
<th>Interview Dates</th>
<th>Interviewee Projects and Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td>IT Service Provider</td>
<td>Program Manager, Senior Project Manager, Team Member</td>
<td>January 2010</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>Company B</td>
<td>Internet</td>
<td>Engineer, Manager, Product Manager</td>
<td>January 2010</td>
<td>Web Mail</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>April 2011</td>
<td>Web Calendar</td>
</tr>
<tr>
<td>Company C</td>
<td>Software Service Provider</td>
<td>Development Manager, Team Member</td>
<td>January 2010</td>
<td>Rail Booking</td>
</tr>
<tr>
<td>Company D</td>
<td>(Offshore Provider to</td>
<td>Project Manager, Product Owner, Scrum Master (3), QA Lead, Team Member</td>
<td>January 2010</td>
<td>Marketing Campaign Management, Customer Relationship Management</td>
</tr>
<tr>
<td>Company E</td>
<td>Company E)</td>
<td></td>
<td>February 2010</td>
<td>Banking, Marketing Campaign Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>Company F</td>
<td>Industrial Products</td>
<td>Program Manager, Project Manager, Director of Engineering</td>
<td>April 2011</td>
<td>Healthcare Instruments</td>
</tr>
<tr>
<td>Company G</td>
<td>IT Service Provider</td>
<td>Engagement Manager</td>
<td>April 2011</td>
<td>Media Entertainment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Company H</td>
<td>IT Service Provider</td>
<td>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), Team Member (9)</td>
<td>May 2012</td>
<td>Airline Customer Service, Flight Booking</td>
</tr>
</tbody>
</table>

[5] [22]. While Bass [7] investigates large-scale development programs but specifically focuses on product owner teams. This research uses a qualitative, grounded theory, approach to investigate software engineering practice comprising 8 international companies and semi-structured interviews with 46 practitioners.

A. Research Sites

The companies, which were selected from a population of large enterprises, are engaged in (typically both) outsourced and off-shore software development projects. The selected companies have head offices in Germany, India and USA although the research sites were exclusively in UK and India due to researcher travel budget constraints. The two largest companies have turnover of 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 Table 1. The companies investigated were involved in either off-shoring (companies B and F) or outsourcing (companies A, C, D, E, G and H). Off-shoring is typically motivated by a desire to access and cultivate specialist technical skills from around the world. Both off-shoring and outsourcing are thought to offer lower cost skills than in-house onshore staff members. All the projects shown in Table 1 are intended for revenue generation. None of the projects are internal IT infrastructure projects.

Company B, for example, is a well known Internet business. Company B retains an in-house development capability in California but has built a development team in India (as well as elsewhere) to reduce costs while also attracting a wide range of specialist skills. Company F, in contrast, has interests broadly in the industrial products space. Company F has headquarters in Europe but also has research and development centers in India and in other territories. These global enterprises allocate work, according to the concentration of expertise, to specialist groups within the enterprise. This helps to avoid duplication of competencies within the organization. The selected IT Services companies (companies A, C, D, G and H) are all well-known vendors in the world-wide software and/or IT service outsourcing sector.

The process used to select companies for the research study comprised two phases: (1) snowball sampling technique ([32] pg. 237; [30] Pg. 37) followed by (2) intensity sampling ([32] pg. 234). Initially, former co-workers and other professional contacts provided access to the first study participants. Those participants then provided access to development teams in other companies. This early, more exploratory, phase of the study focused on replicated interviews from a broad range of Companies (companies A, B, C, F and G). Snowball sampling gave access to a range of project teams and stakeholders with different perspectives. Company C, for example, exclusively use agile software development methods. Companies E and G, On the other hand, provided some agile skeptics with negative experiences to report.

Later in the study, intensity sampling was used to obtain greater richness and depth in the study by targeting a larger number of interview participants with different responsibilities in the same company or software development program. The interviews at Company H, and Companies D and E provided triangulated perspectives from developers, QAs (quality assurance, testers), project management, program management and corporate-level executives. These differing perspectives implement the intensity sampling technique ([32] pg. 234).

To summarize, research sites were selected to provide replication using snowball sampling in the early phase of the study. Then intensity sampling was used in the later phase to enhance both depth and richness; hence increasing data reliability through participant triangulation.

Using both snowball and intensity sampling is a combination sampling approach which provides methodological triangulation to the sample selection. Combination sampling was used to provide a snapshot of participant perspectives but
also the underlying motivation for these software development practices. It is difficult to assess practitioners motivation for selecting software development practices using other research methods such as surveys for example.

B. Data Collection

Documentary sources were used to support the study. Access to certain commercially confidential corporate agile development method process guidelines was obtained. These corporate guidelines outline: agile practices, roles, policies and recommended techniques. Some documentation produced for specific software development programs, such as design and architecture documents, has also been investigated. Publicly available and web hosted marketing materials were also reviewed. These included white papers, technical reports, case studies and descriptions of vendor capabilities designed to inform potential customers. On-site visits enabled first-hand observation of working practices and work place environments. Some secure work environments were visited. Coordination meetings (stand-up meetings) were observed (at Companies C, D and H) for both co-located and distributed scrum teams. This enabled investigation of arrangements for distributed scrum coordination meetings using both video- and audio-conferencing technologies. Various informal, sometimes off-site, discussions with executives, project management and development team members were conducted.

The primary data used in the study were from face-to-face recorded interviews. Interviews were conducted with 46 practitioner interviewees, with recordings professionally transcribed and reviewed. An open-ended interview guide approach was used to conduct and structure the interviews. An example of the semi-structured interview guide used in this research is presented in Appendix 1. Probing questions were used to encourage interviewees to provide more detail. The interviews were open-ended because respondents were given opportunities to raise any topics, issues and concerns they wished outside the scope of scripted interview questions. Interviews were typically conducted in small meeting rooms exclusively booked for interview purposes on the company premises.

C. Data Analysis

Initially, the audio interviews and corresponding verbatim transcripts were carefully reviewed to ensure consistency. The transcript text was then imported into a qualitative data analysis software tool, in this case Nvivo V9 [2].

Grounded theory is an approach for inductively generating novel theoretical ideas from data. The new theories arise from the data and are thus said to be grounded. The grounded theory analysis began with identification of concepts within the interview data [18]. The interview concepts were coded and then compared within and between interviewees. The constant comparison analysis technique is used to explore homogeneity and heterogeneity within the interview data. These interview concepts were then iteratively grouped and refined into selected categories. Thus, interview concepts were combined to create categories 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 [17]. The categories become saturated as data collection progresses [16]. This categorization forms the basis of the grounded theory which is presented below.

IV. FINDINGS

The main contribution of this paper is to develop a grounded theory from practitioner descriptions of scrum master role tailoring. The research identifies the emergence of scrum master teams to scale-up agile methods to large international projects. We identify 10 scrum master activities in this scrum master team context. Enterprise software development projects are characterized by, large work volumes, short deadlines and entrenched organizational structures. These constraints lead to tailored agile approaches.

First, let us consider use of traditional waterfall methods. Using traditional waterfall methods “the development cycle used to be pretty huge, you know, huge requirement document, huge design phase - three or six [month] development cycle and so on (Product Manager, Company B). But time taken to conduct the waterfall development process undermines customer satisfaction “there was always a huge difference [between] what our customers [were] demanding and what we are giving them” (Product Manager, Company B). This is seen as an inevitable part of the development process “I think that’s an inherent issue in the waterfall model” (Product Manager, Company B).

In summary, in a

“typical waterfall model, where you had requirements being collected, analyzed and then, you had a design phase… and then a development phase started and then a QA phase and then you realize everything is broken and things are not adhering to contracts, so you [the client and software service provider] fight with each other” (Development Manager, Company C)

Company E performs (non-agile) customization of its own software product on client sites. As we will see later, Company E also uses scrum with an out-sourced development partner internally for its own product development. Consider a customization project at a major international bank in the UK. There are challenges with governance of projects

“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).

This project manager who has 30+ years of IT industry experience, illustrates how in a large enterprise context it is difficult to hold project teams to account in terms of scope, budgets and deadlines.
Company F adopted agile methods because “they wanted to bring the best out of the team. And [improve] the quality of the delivery” (Scrum Master, Company F). They were concerned that

“in a traditional waterfall model, it’s like you go for a requirement analysis and then you go for design and then you go for coding. And by the time you actually go for delivery, the entire [need] has changed” (Scrum Master, Company F).

The focus of this study is on the tailoring of agile methods in large-scale enterprise software projects. For example, “two years back we have adopted scrum as our methodology” (Scrum Master, Company D). Similarly “what we’re doing is following a scrum, stand-up meetings, retrospective meetings, following backlogs closely and tracking things like that” (Product Manager, Company B). Adoption of agile methods at Company F “was actually a customer-driven process, rather than an internal project driven process by us” (Scrum Master, Company F). Clients are demanding substantial agile development programs, from offshore service providers. For example, “we are seeing that big programs are coming so, you know 100 people are suddenly required. [which means] 10, 12 scrum masters [are needed]” (Corporate Lead Architect, Company H). There has been skepticism about agile adoption “initially, frankly speaking, when I attended the scrum master training, it looked like a fad…But once I got involved and I saw…why it is better than waterfall” (Scrum Master, Company H). Similarly “when I moved into agile for the very first time, I was very skeptical about it. So that should be a very frank opinion, right?…[But] once you’re getting involved, then you will know all the benefits of it” (Scrum Master, Company F). For their own product development, in Company E “we follow fairly rigidly the Agile process. We have stand-up meetings, we have project kick offs, we have feature discussions, we regularly talk to each other and product owner plays a part” (Engineering Director, Company E). Although “we definitely don’t do extreme programing” (Engineering Director, Company E).

The scrum method advocates iterations of between 2 and 4 weeks “we call it a sprint; that’s like two weeks” (Scrum Master, Company D). The sprint is a dedicated period of development activity.

A customer demonstration is conducted at the end of each sprint, “after [the sprint] we used to do the ‘show and tell’ [customer demonstration] after the work” (Project Manager, Company H). The purpose of the customer demonstration is to obtain client feedback “when we are getting feedback [from the client], we have to change the product” (Scrum Master, Company H).

The retrospective in scrum is used to reflect on the sprint and identify lessons learned “we have a sprint retrospective after the end of every sprint” (Scrum Master, Company H). During the retrospective “we just wanted to be sure what has been wrong, what was good [and] what can be improved” (Scrum Master, Company H).

Co-located scrum teams are always considered desirable. Sometimes in large projects this is not possible “I worked with Europe, US and India together. So definitely it’s a challenge” (Engagement Manager, Company G). One project at Company H “was based on two locations, the android application was being dealt with from Bangalore office and rest of the iOS, iPhone thing was being done from here [Delhi]. So there were two scrum masters involved” (Scrum Master, Company H). Similarly, “we were in a situation where we had the product people in a different time zone. Then we had the testing people in another different time zone” (Engagement Manager, Company G).

Agile methods advocate multidisciplinary teams “a group of, let’s say, five to six developers, a test analyst, and a scrum master going into a team” (Test Analyst, Company H). Again “we have a scrum master and we have four or five people team in one Agile team” (Project Manager, Company H). A “team size is eight, we have one scrum master, one or two C++ developers, depending upon the task and two or three Java developers, one test analyst and one business analyst” (Senior Developer, Company H). What does a scrum master actually do? The scrum master activities identified in this research are described in turn below.

A. Scrum of Scrum Facilitator

Development team size increases in order to deliver larger software systems within required deadlines. For the purpose of this study, large is defined as at least 25 developers engaged for a duration of three months or more. However, the scrum method recommends small development teams. It follows that large development teams need to be divided into smaller teams for compliance with scrum. For example, “the entire [client] team is divided into two sub-teams. One is called the red team; the other is called the blue team. The red team has their own scrum master; the blue team has their own Scrum Master” (Developer, Company D). A similar approach is followed in even larger projects, for example “say I have a big project, 100 members’ project, where there are scrum teams with 10 members so we have 10 scrum masters” (Engagement Manager, Company G).

The ‘scrum of scrums’ approach is used to coordinate the activities of scrum teams working together on a large project. The term ‘scrum of scrums’ specifically refers to a coordination meeting comprising the scrum masters and product owners from each team working together on related projects, such as “all the scrum masters and all the proxy product owners meet” (Project Manager, Company H).

This study found there was frequency and attendance tailoring of the scrum of scrums coordination meetings. Team did not always find it attractive to conduct the scrum of scrums daily. Two projects at Company H, convened their coordination meetings less often. Their scrum of scrums “meet weekly” (Project Manager, Company H) or “That [scrum of scrums] happens every alternating week” (Scrum Master, Company H). While Company B selected team members from each scrum team to “become part of the sprint stand-ups at least a couple of days in a week” (Product Manager, Company B). Thus the frequency of coordination meetings is adjusted to ensure stakeholders feel good use is being made of their time. As one product manager states, “through that route we are able to closely monitor the progress of…other teams on a need basis” (Product Manager, Company B).
Another tailoring approach is to reduce the number of the attendees at coordination meetings. In some projects many stakeholders work needs to be coordinated due the high level of coupling between requirements within the project. At company H “we are having the internal scrum of scrums meetings, [comprising] scrum masters, onsite designers, and BAs [business analysts] as well” (Scrum Master, Company H). However, coordination between loosely couple aspects of a project is ineffective “there were 25 people on the same [scrum of scrums] status call. That was not making sense. So, we tailor [such] that [only] important stakeholders will be involved on a daily scrum [of scrums] call. So, we have a daily scrum call which is basically useful for everyone” (Scrum Master, Company H).

Scrum masters also have to facilitate unusual combinations of scrum team colocation and geographical distribution. Some clients can find it attractive to adopt a mix of onshore and offshore development activity. The project team deployment reflects the technical skills available at different locations. On one project in Company H the client operated scrum teams both onshore and offshore, such that “we do scrum of scrums offshore, and we do scrum of scrums onshore as well” (Technical Analyst, Company H). Requirements are managed as user stories by a staff hierarchy “we have a product manager [onshore] who sends the backlog to the [onshore] scrum master. The [onshore] scrum master sends us across [the backlog]” (Scrum Master, Company H). And “here in offshore we have two teams. So we have basically two leads at offshore who will be coordinating with the scrum master at onshore” (Scrum Master, Company H). Thus, tailoring here results in the two offshore scrum masters receiving requirements from an onshore scrum master. The onshore scrum master is coordinating onshore and offshore teams. The procedure is intended to reduce interference between the software under development by the different teams. However, colocated scrum of scrum teams were typical in this study.

To summarise, scrum masters facilitate scaling by supporting a scrum of scrums approach. In scrum of scrums, scrum master ensure stakeholders are fully utilized by adjusting the composition and frequency of scrum of scrum meetings. ...XXX ...

### 5. B. Sprint Planner

Sprint planning is primarily conducted by a product owner who prioritises requirements for the next iteration. Thus, the product owner liaises with stakeholders to identify and select the most important requirements for inclusion in the sprint. The scrum master provides technical support to the product owner, who may business skills but not technical expertise. For example, at Company D “my product owner is completely a business analyst. He might not be aware of some technical aspects of the project” (Scrum Master, Company D). The scrum master plays an important role in reviewing the product backlog for technical dependencies. The technical dependencies might suggest an implementation sequence for user stories, such as “which [user story] model we should do first” (Scrum Master, Company D). Technical dependencies between tasks are identified “say task six is dependent on task five, so it cannot start before the other task can finish. So we specify exact dates for those tasks” (Product Manager, Company B). Thus, while product owners prioritise using business need as the main criterion, scrum masters look at prioritisation from a technical dependency perspective.

Scrum masters also coordinate the decomposition of requirements, usually in the form of user stories, to technical tasks. In turn, these tasks are decomposed in smaller technical activities, for example “we need to break any tasks into sub-tasks” (Product Manager, Company B).

Precise planning and estimation ensures that all user stories included in an iteration are fully implemented and tested. However, sometimes some user stories are incomplete at the end of an iteration. For example, if “some part of [the sprint] is left, it goes again into the product backlog” (Scrum Master, Company H). This may be because a team is inexperienced in the business domain or is using a new technology. Placing any incomplete user stories into the product backlog allows a re-prioritisation of both new and incomplete user stories.

In a scrum of scrums context, sprint planning is required to avoid interference between the development activities of different teams. An important tactic is to ensure different teams are given distinctive responsibilities, for example “we have a strategy team. Then we have another team which is infrastructure” (Scrum Master, Company H). The product owner assigns user stories to the corresponding scrum team, by asking “is this [user story] a strategic initiative or is it related to infrastructure? Then, the [user] story is assigned to the [appropriate] scrum master” (Scrum Master, Company H). The work of the scrum teams is differentiated in this situation, such that the teams are “working on different areas. So they will not be hitting each other’s area actually” (Project Manager, Company H). In another case, one team is focused on fixing bugs and issues with the existing code base, for example “there is one support team which actually works on different problem requests and change requests” (Project Manager, Company H). Scrum masters check the assignment of user stories to minimise interference between different teams.

Scrum masters coordinate failures of the sprint planning process, such that “If there is any impediment because of [task conflicts with another] team, so the scrum master tries to resolve this by interacting with the other team scrum master” (Project Manager, Company H). Again, “there are a number of teams. So sometimes it happens that one of the teams is working on a particular piece of code and the other team is doing some other piece of code which impacts this one” (Technical Analyst, Company H). Where possible these task conflicts are avoided by sprint planning. When tasks conflicts occur the scrum of scrums coordination meeting is used to surface the issues. Scrums masters then cooperate to minimise the interference.

In summary, scrum masters contribute to sprint planning by providing technical support to product owners, ensuring user stories are correctly prioritized and assigned to teams ...XXX ...

### 5. C. Integration Coordinator

The presence of multiple scrum teams (whether co-located or geographically distributed) leads to the problem of integrating the code bases under development by the different
teams. Code integration is a challenging and important issue, for example “integration is one, the big, issue in these type of environments because most of the team members work on the same type of files” (Project Manager, Company H). Similarly, “as a scrum master, the challenges were…code synchronizing problems, task updating problems” (Scrum Master, Company D).

Scrum masters must adopt and disseminate an integration coordination strategy. One approach is to merge code bases for each release, for example:

“They follow their sprint cycle, we follow our sprint cycle. Finally, they match and we take their code base merge into our one and give it to QA as a single release and then it goes into production” (Product Manager, Company B).

Another approach is to prevent updates ahead of customer demonstrations, for example:

“suppose team A has a ‘show and tell’ [customer demo] tomorrow, say Wednesday. On Monday there would be a code freeze for that particular branch. On that branch, no other team members from team B and team C, would be allowed to check in any code into that particular branch. So for 2 days, because there would be show and tell preparation, for 2 days the branch would be blocked by that particular team” (Project Manager, Company H).

Scrum masters agree and implement a code integration process. Code from multiple teams is integrated into a single code base prior to release.

...XXX...

D. Learning Coordinator

Scrum masters facilitate training for team members, “for any newcomers we generally have orientation trainings. Apart from that, we try do weekly or fortnightly technical overview meetings by every one of the team members, in every group, on every project” (Product Manager, Company B). Staff joining agile teams need to undergo some induction process. Company H has “a complete training plan for everybody who comes into the team. So it is like 15 to 20 days plan” (Project Manager, Company H). Thus, the scrum master coordinates induction of new members to the team and identifies technical training needs. The feature team concept used in agile methods means that team members have a wide set of skills to acquire. Traditionally organized software development teams tend to have more tightly focused job descriptions.

Traditionally, in the “normal waterfall method, if a tester is doing testing, he cannot be an analyst. He is always a tester” (Engagement Manager, Company G). Further “in ideal waterfall model, we have architects, separate designers, separate developers, separate testing team” (Engagement Manager, Company G). When work tasks are assigned by managers they “tend to give people monotonous tasks because once they’re expert in that area, [they] keep on giving that person the similar work” (Product Manager, Company B).

Agile methods can enable team members who want to learn new skills and techniques. Team members “may want to try out different things. Because they want to learn different technologies and areas” (Product Manager, Company B). Again “sometimes what happens, they [team members] want to brush up their knowledge so they take up something else” (Engagement Manager, Company G). Scrum masters can facilitate team member learning “for a particular area of work they can skill up so in the next sprint we can give them a different responsibility” (Engagement Manager, Company G). The scrum master can help “scale up skills of the whole team to a different level where everybody’s able to do almost everything” (Product Manager, Company B).

Novice team members need some time to learn how to estimate accurately “after a couple of sprints, they settle into a rhythm and then they’re able to estimate properly” (Product Manager, Company B).

Where scrum masters facilitate team member learning there are three main benefits. First, this has the benefit of reducing dependence on particular individuals “instead of individuals getting bound to a particular area and that person leaves we have a huge gap within the team” (Product Manager, Company B). Second, productivity is improved “if you’re not giving them [team members] that freedom then obviously productivity will keep on going down” (Product Manager, Company B). Third, “everybody feels that they are equally responsible. It [the team] delivers something in [the agreed] timeframe” (Engagement Manager, Company G). This shared accountability ensures that “if Person A cannot do something it’s Person B’s responsibility to make sure that we are delivering something” (Engagement Manager, Company G).

The idea of learning by trying new activities extends to the scrum master role itself. So, “sometimes the scrum master can become the scrum team member and somebody from scrum team [can] skill up into the scrum master’s role for certain features” (Engagement Manager, Company G).

E. Heart Beat Monitor

A primary benefit of scrum is the ability of managers to track and monitor progress against development effort estimates. This is achieved in scrum by assigning story points, that reflect the size and complexity of requirements represented by user stories, and counting down story points through burn-down charts as work is completed. The team communicates its activities using a daily coordination meeting, “15 minutes stand-up we do every day” (Project Manager, Company H). The coordination meets are used to report to managers (notably product owners) the current status of development work during the sprint.

The coordination meetings are often conducted standing up, to minimize the meeting duration. Similarly, “the day starts with a stand up, which happens at a pre-defined time, which the team agrees upon on. . .I would say it kind of kick-starts the day for everybody.” (Development Manager, Company C).

The agenda of the stand-up is always fixed “in a stand-up, you are supposed to say what I did yesterday, what I’m going to do today, [and] what are my impending blocking issues” (Scrum Master, Company F). In “our stand-up meeting everybody is giving their status ‘I worked on this, I worked on that.’ I should know what others are doing” (Senior Software Developer, Company H).
Some developers, initially at least, “are really not very happy to tell their status in a stand-up. [And yet] it is very convenient for them to speak because they know everybody is speaking. So they find the daily stand-ups very, very useful,” (Project Manager, Company H).

Scrum masters coordinate team members as they perform detailed estimates of the work required to implement user stories. So, team members “have to come with effort estimates and say that, ‘Yeah, I can comfortably take [that task] up, I understand the dependencies and I can finish it in three days or five days.’ No manager goes back and tells the engineers that [their estimate is too long]” (Product Manager, Company B).

Since the team are contributing to task estimation they feel a greater sense of commitment to their estimates. So “by choice people are committing to something, generally you are able to meet whatever you have signed up for” (Product Manager, Company B).

Scrum practitioners keep track of user stories that are completed and tested. These completed tasks should be monitored on a burn down chart. So “every day we have stand-up during which we burn down the number of hours which has been estimated initially” (Scrum Master, Company H). When the development activity “is done then the story is done from offshore perspective. And there is a particular status which is called a sprint complete” (Scrum Master, Company H). Then “we move that story, which has a sprint complete status, [to the client, who] has to decide whether this has to go to production or not” (Scrum Master, Company H).

Sometimes project team members complain of agile methods being used as a means of micro-managing team member effort. For example “I think there has to be a different way of measuring productivity or delivery. Do I really need to track an individuals activity on a daily basis, or is it enough that the team is able to deliver x number of stories for a given sprint?” (Scrum Master, Company F).

There are problems with comparing agile team performance. For instance we can compare Team A and Team B, “Team A might have delivered let’s say 15 stories, and Team B has delivered only five stories”. (Scrum Master, Company F). Superficially it appears that Team A is more productive than Team B. From the “outside it looks like Team A is highly productive. And quite a lot of time, the people who are in Team B will suffer” (Scrum Master, Company F). However, if the user stories from Team A “had, let’s say, forty five acceptance test cases [but Team B’s user stories] had around 70 acceptance test cases, Team B’s user stories are highly complex” (Scrum Master, Company F).

F. Process Anchor

During the initial adoption phase, everyone in the team is learning together. So in Company F, “scrum masters in some teams were [external consultancy staff members, coincidentally from Company C]. And then we also had one or two team members from [Company C] helping us with respect to make sure that we adopt [agile methods] in the right way…They were the ones…passing on the knowledge to us.” (Scrum Master, Company F).

A “basic purpose of a scrum master is [to ensure] that the team follows agile principles and practices” (Scrum Master, Company H).

“It’s very difficult to find a highly cooperative team, until you get an excellent leader or a coach. So the key actually is the mentor, [who] has to remove all the insecurity feelings from every individual, [and] has to ensure that it is not about individuals, it is about team” (Scrum Master, Company F).

The scrum master also ensures that the processes and practices selected are appropriate for the technical and commercial context. A negative report on the use of agile methods came from a program manager at Company E. The team “developed the requirements and the program and tested [an increment of software and] then move to the next one” (Program Manager, Company E). The problem is that in large enterprise programs teams must carefully check for dependencies between requirements. So

“this struck us as a bit dangerous. Because, you could get quite a long way down and find something in [an increment], 10 [increments] on, that affected an earlier [increment] that needed reworking” (Program Manager, Company E).

So on this project, dependencies had not been thoroughly identified resulting in unnecessary rework.

Scrum masters accommodate agile method tailoring due to geographically distributed work allocation. This includes unusual scrum coordination meeting arrangements, such as “we had to do the scrum calls twice [a day] instead of once. One India versus the US, the other one India versus Europe.” (Engagement Manager, Company G). An onshore client is typical in offshore development projects “onshore at the client site [we have a] proxy product owner representing our team. [The proxy product owner] is interacting with [the client’s] project stream lead and technical product owner. Then we have the actual [client] product owner, so there is a hierarchy. Scrum master and team is in India; and project stream lead, technical product owner and proxy product owner are [onshore]” (Project Manager, Company H).

G. Task Estimator

Requirements gathering and prioritization in scrum is done by a product owner. A “product owner is making a story…after [story prioritization] then the product backlog is finalized” (Scrum Master, Company H). The development team is self-organizing because “we sit together as a team, we give points to each and every story, [estimating] how much time will it take” (Developer, Company D). The “scrum master sits with the team to do the estimation of all the user stories” (Project Manager, Company H). But “it’s the scrum master and the product owner’s responsibility to make sure the backlog is prioritized. We as team members don’t really have the access to the backlog” (Developer, Company D). The scrum master contributes to advanced planning for each sprint. One element of sprint planning is conducted at the start of the each sprint “there is a kick off meeting every week because there are two teams” (Engineering Director, Company E). The product owner has primary responsibility for developing requirements which
are usually in the form of user stories. The scrum master helps the product owner prioritize user stories “the product owner and scrum master involved in the next sprint planning” (Scrum Master, Company H). At the start of each sprint, the scrum team is given user stories in the form of a sprint backlog “we have a sprint planning session in which we have a backlog” (Scrum Master, Company H).

An important aspect of self-organizing teams in scrum is that team members are empowered, where ever possible, to select their own work tasks for a sprint. So, “we need to give freedom to developers is decide which task they want to do in the sprint planning meeting” (Product Manager, Company B). Scrum masters are “actually grooming your team members to be able to better decide upon what part of the implementation to take” (Development Manager, Company C).

Secondly, scrum masters need to enable work task estimation by team members “tell me how time we’ll need to finish this and [we’ll] commit to that” (Product Manager, Company B). The “estimates would either be too high or too low, but after a couple of sprints… they’re able to estimate properly” (Product Manager, Company B).

When the [scrum] team is developing the second sprint, the QA, BA, scrum master and product owner are discussing [details of] the third sprint” (Scrum Master, Company H). User stories “come to us and we check whether we have the capacity to do those stories or not. We do the planning, we do the estimates” (Scrum Master, Company H).

H. Impediment Remover

The scrum masters in this study spend time investigating and removing impediments for team members. The coordination meetings are used to surface problems “stand-ups are usually 15 minutes. In the 15 minutes, we don’t really talk about any issues [that arise]. So once the stand-up is complete, then we discuss any issues or any clarifications” (Scrum Master, Company D). Scrum masters have to “focus on the blocking issues [and] try to facilitate how to get rid of them” (Scrum Master, Company F).

So, “my main role [as scrum master] is to make sure that the team members have enough information that they can work on” (Scrum Master, Company D). The “scrum master takes [issues and queries] offline after the [standup] meeting” (Project Manager, Company H). Similarly, “if we have any problem, we have a protocol to follow, we send a mail to our scrum master” (Senior Developer, Company H), the scrum master is then expected to find a solution to the issue.

I. Spike Process Coordinator

The “spike” concept provides a way to handle unexpected complexity encountered during a sprint. The development effort required for each user story is estimated during sprint planning. Spikes occur when software implementation risks exceeding that previously estimated development effort. Spikes are problematic because they compromise a team’s ability to deliver their promised productivity.

Some user stories are inherently more complex than others “if we have a complex story we’ll be first preparing a design document.” (Scrum Master, Company H). For these complex stories “there will be a task for the design document creation that could range for seven/eight hours, maybe more than that depending upon the complexity” (Scrum Master, Company H).

A development team that invokes the spike process too frequently to disguise poor development effort estimation. Thus, there is a danger the spike process can be misused by teams “I am not a spike person; I just find that it is a cop out from commitment that I will deliver this” (Engineering Director, Company E). But in general scrum master need a process to be able to highlight unexpected user story complexity. Scrum masters can then negotiate additional time for working that user story.

J. Test Coordinator

In traditional methods testing and quality assurance was conducted toward the end of the project. So,

“in a waterfall model, you will actually budget for a QA cycle towards the end of the development cycle and then you saw some budget for… over here we are taking the budget and we are spreading it across your development cycle” (Development Manager, Company C).

Scrum masters coordinate the quality of software produced by the team. So, “When you think you’ve got every logical point accomplished on a story, you would actually run test cases to make sure they are all fine” (Development Manager, Company C). The “unit testing and the integration testing, that will always be done in-house” (Program Manager, Company E). In contrast, user acceptance testing is often done by a separate quality assurance team in large enterprise projects. In addition, automated “smoke” tests are used as part of a continuous integration pipeline. Teams

“have a selection of tests called smoke tests which are run on the pipeline… You run those smoke tests to figure out if there are parts of the systems that are broken because of your changes” (Development Manager, Company C).

Some teams “actually put the acceptance criteria on the [user] story [card] itself” (Scrum Master, Company F). Initially

“when you have put acceptance test case obviously that is going to fail. Because we don’t have a code yet behind it. Then we start to fill in the code. To make the acceptance test pass. So it was actually a test-driven development” (Scrum Master, Company F).

V. DISCUSSION

VI. LIMITATIONS

Three tests have been identified for establishing the quality of descriptive empirical social research: construct validity, external validity and reliability [41]. Construct validity is ensured by using multiple sources of evidence and ensuring a chain of evidence. Specific perspectives on the scrum master role have been triangulated through interviews with both development team members and product owners as well as with scrum masters themselves. In fact, interviewees have also included
corporate-level executives that can offer perspectives on the scrum master role. A chain of evidence has been ensured by including verbatim quotations from interviewees in the findings section of the paper. The grounded theory approach was to analyze the transcript data and create the activity categories. However, transcript quotations maintain the chain of evidence to original sources; the interviewees themselves. The construct validity has also been tested by obtaining feedback on an early draft of this paper from Company H.

External validity can be achieved through study replication. Multiple sources of evidence have been achieved by conducting replicated studies at eight companies with large studies at Companies H and E (along with their offshore service provider Company D). 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 achieved by minimizing errors and biases in the study. The study uses an open-ended semi-structured interview guide. The interview guide acts as a script guiding data collection and ensuring consistency between interviews. An example open-ended semi-structured interview guide is presented in [10].

VII. CONCLUSIONS AND FURTHER WORK

This paper explores large-scale enterprise software development programs using practitioner descriptions of agile method tailoring. Specifically tailoring of the scrum master role has been systematically investigated using the Grounded Theory research method. Grounded Theory is emerging as a technique in software engineering research for exploring complex real-world settings. Activities performed by scrum masters, in the context of scrum master teams have been described. These 10 activities help scale agile methods to large programs: scrum of scrum facilitator, sprint planner, integration coordinator, learning coordinator, heart beat monitor, process anchor, task estimator, impediment remover, spike process coordinator and test coordinator. The scrum of scrum facilitator coordinates work with the other scrum services in the program. The sprint planner supports the user story triage that occurs prior to development work starting in each sprint. The integration coordinator merges code bases developed by cooperating teams working in parallel. The learning coordinator ensures project team members have opportunities for continuing professional development. The heat beat monitor ensures that the team maintains a productive sense of progress during each sprint. The process anchor nurtures adherence to agile methods. The task estimator supports workload planning and ensures user stories are appropriately assigned to the team. The impediment remover obtains information to ensure developers can make progress with their work. The spike process coordinator evaluates and manages any unexpectedly complex user stories that are encountered within a sprint. The test coordinator ensures the software produced meets quality requirements.

These scrum master activities contribute governance to agile development programs in the CMMI Level 5 certified companies investigated. Further, these activities are performed by scrum masters in the context of scrum master teams. Thus, scrum master team members can specialize by assigning these activities within the group. Alternatively, aspects of the scrum master team activities can be delegated to colleagues. The scrum master team offers important resources for tailoring agile methods in large scale development programs. Managers that understand the complex range of activities undertaken by scrum masters can mitigate burnout and high staff turnover by offering targeted training and support resources.

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

VIII. ACKNOWLEDGMENTS

I am grateful to the companies and interviewees who participated in this research. Thanks also to the students of the Executive MBA at the Indian Institute of Management, Bangalore who facilitated access to several participating companies. The International Institute for IT, Bangalore provided hospitality for several research visits. The research benefited in part from travel funding from the UK Deputy High Commission, Bangalore, Science and Innovation Network, and the Institute for Innovation, Design & Sustainability (IDEAS) at Robert Gordon University, UK. Accommodation and sustenance was provided by Company H during the data collection visit to Delhi, India.

REFERENCES


APPENDIX I

Interview Guide, Agile Method Tailoring [Company H], May 2012

Background Notes

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

The purpose here is to try to understand how agile methods are used in [Company H], so that we can try to learn in the future. I want to ask you the following questions and tape record your answers. I do plan to publish interview extracts but I will make names, projects and clients anonymous. Can I switch on the recorder?

Agile Processes

• What agile methods and practices are you using?
• Would you describe agile methods as being successful for you? In what ways?
• What challenges have you encountered with agile methods?

Scaling to Enterprise Projects

• Describe any software tools or technologies you use to support agile methods?
• Have you adapted agile methods because of the geographical distribution of the team?
• Have you adapted agile methods because the client organisation was geographically distributed?
• Have you adapted agile methods because of a particularly large team?
• Have you used agile methods in a context with demanding regulatory compliance? What adaptations did you make?
• Have you used agile methods in a particularly complex domain context? What adaptations did you make?
• Have you used agile methods on a particularly technically complex project? What adaptations did you make?
• Have you used agile methods with an especially complex range of stakeholder relationships?
• What adaptations did you make?
• Have you adapted agile methods for use on a strategically important enterprise architecture programme?
Future Perspectives

• What future trends do you foresee in your use of agile methods?
• If there was one thing you could change about the way agile methods are used at [Company H] what would it be?
• What advice would you give to improve agile productivity?
• What advice would you give to improve agile product quality?
• What advice would you give to improve transitioning to offshore agile?

Any other comments

Now, I want to check if there is anything else you would like to say?

• Do you have any further comments in relation to agile methods?

About Your Project(s)

Now I want to ask some questions about you and your project. These details will be kept confidential.

• What project are you working on currently? How many projects?
• How is the project team structured (for management purposes)?
• How is the project team organised geographically?
• What is the project domain? What is the project purpose?
• How large is the project in terms of team size? In terms of value?
• When did the project start?
• How much longer will the project run for?

About You?

• Your name?
• Your age?
• What is your role (product owner, developer, architect or scrum master)?
• How long have you been in your current role?
• How long have you been working in your current organisation?
• How long have you been working in the software industry?
• What formal qualifications do you have, if any?