

Enhancing product and service capability through scaling agility in a global software vendor environment

Ramesh Lal
Auckland University of Technology
Private Bag 92006
New Zealand
Ramesh.lal@aut.ac.nz

Tony Clear
Auckland University of Technology
Private Bag 92006
New Zealand
Tony.clear@aut.ac.nz

ABSTRACT

Agile software development has become increasingly common in software vendor organisations, and their impact on practices and roles is now extending beyond the project level across the entire organisation. In this study, we investigate how a major Australia-based multi-site global software vendor transitioned from a structured to a Scaled Agile approach.

We demonstrate how practices and roles in a distributed software vendor evolved over time across the organisation in an on-going process of their global agile transformation. Through this elaboration, we identify three major agile transitions and the contribution of a scaled agile approach in the building of market driven capabilities. We theorised a relationship between increased dynamic capabilities of the firm and a scaled agile transition. Supporting our view, we noted that agile practices and roles across the organisation contributed as expected to improved internal process capability. More surprisingly, they were also found to increase our vendor's ability both to identify and take advantage of opportunities and to innovate in global product and service development and delivery.

CCS CONCEPTS

• **Software and its engineering** → **Software creation and management** → **Software development process management** → Software development methods → Agile software development •

KEYWORDS

Global Software Engineering Research, Global Software Engineering Practice, Dynamic Capabilities, Scaled Agile Methods, Software Vendors, Product and Service Innovation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICGSE '18, May 27–29, 2018, Gothenburg, Sweden © 2018
Association for Computing Machinery. ACM ISBN 978-1-4503-5717-3/18/05...\$15.00 <https://doi.org/10.1145/3196369.3196378>

ACM Reference format:

R Lal, T. Clear, 2018. Enhancing product and service capability through scaling agility in a global software vendor environment. In *Proceedings of International conference on Global Software Engineering, Gothenburg, Sweden, May 2018 (ICGSE '18)*, 10 pages. <https://doi.org/10.1145/3196369.3196378>

1 INTRODUCTION

Software vendors in a global market-driven environment not only anticipate competition but also require continuous investigation and learning about change happening in global software development practices. Importantly, market-driven changes in the global space are the basis for driving their software process (SP) improvement. If not aligned with market needs, then an ineffective or redundant SP affects their ability to identify, develop and deliver software products. For software vendors, aligning business goals and objectives with software development practices and roles are important drivers for SP improvement (through new method adoption). Continuous SP improvement allows software vendors to create firm-specific in-house dynamic capability [2, 16, 17].

In this paper, we investigate how a major Australia-based multi-site global software vendor developed its capabilities through a multi-year (2003-to date), three-stage transition from a *Structured* process (based on Rational Unified Process –RUP), through a *Hybrid-Agile* method, to a *Scaled Agile* approach using *Disciplined Agile Delivery (DAD)*.

The paper outlines the background to this scaled agile transition, the notion of “dynamic capabilities” [16,17], describes the case study and methods applied, then tabulates the practices and roles in use across three temporal stages of evolution in the organisation's SP. We analyse how these changes can be related to a dynamic capabilities framework, what that contributes to our understanding of “agility” and conclude with recommendations for global software vendors wishing to engage in a scaled agile transition.

2 BACKGROUND AND THEORY

Prior work in SP improvement initiatives has applied normative frameworks such as the Capability Maturity Model [3], or in the case of distributed development, where research into SP has been limited [5] more custom-developed frameworks specific to distributed development have been implemented [14], but little attention to date has been given to SP for agile methods in global software development.

Agile methods have increasingly moved into the mainstream, and are now progressing beyond the team level into the wider

organisation. Consultants have been promoting a variety of different ‘scaled agile’ frameworks e.g. [1,9,10] to help embed agility within global software organisations, but it is early days yet for these enthusiasm-driven and practitioner-led initiatives. Some academic evaluation of these developments has taken place [12, 7], but there has been limited theorisation of these developments to date.

In this study we map the roles and practices identified from the interview data gathered over the course of a longitudinal case study, which originated in the work of the first author conducted in [8]. We apply the ‘lightweight software practice framework’ of [6] to help classify the practices identified, and the cause-effect diagrams of [15] to draw out the relationships between issues encountered, the practices and roles adopted in response, and the evolution of those agile practices and roles.

In an attempt to interpret the contribution of scaled agile practices to the effectiveness of the vendor organisation and its software processes we adopt the ‘dynamic capabilities’ framework [16,17,2,13] as an underlying explanatory theory. In [7] we had indicated the promise of a ‘dynamic capabilities’ framework for understanding scaled agile developments across the enterprise, so here we pick up on that work.

Dynamic capabilities have been defined as:

“the firm’s ability to integrate, build, and reconfigure internal and external competences to address rapidly changing environments. Dynamic capabilities thus reflect an organization’s ability to achieve new and innovative forms of competitive advantage given path dependencies and market position” [13].

Within the dynamic capabilities framework three broad groups of capabilities have been identified [17], adaptive, absorptive and innovative, which reflect respectively the organisation’s ability to take advantage of opportunities; make use of information or demonstrate novelty in product or service development. More specifically while the three are interrelated they are “conceptually distinct” [17], where:

“adaptive capability stresses a firm’s ability to adapt itself in a timely fashion through flexibility of resources and aligning resources and capabilities with environmental changes”

“Absorptive capability highlights the importance of taking in external knowledge, combining it with internal knowledge and absorbing it for internal use”

“Innovative capability effectively links a firm’s inherent innovativeness to marketplace-based advantage in terms of new products and/or markets” [17].

In [7] we have hypothesised “that the ‘dynamic capabilities’ model may be adapted to accommodate the realities of a software vendor embarking on a programme of scaling agile across the enterprise”. We have adopted this framework [17,2,13] as an underlying explanatory theory in an attempt to interpret the contribution of scaled agile practices to the effectiveness of our vendor organisation and its software processes.

We therefore conjecture that:

C1: dynamic capabilities of software vendor organizations would be enhanced as a scaled agile transition progresses.

This conjecture was supplemented by three further conjectures based on insights from our analysis (section 4), namely that:

C2: Responding effectively to time pressures through an agile transition would most directly enhance the firm’s adaptive capabilities [17] – i.e. its ability to respond to opportunities.

C3: Responding effectively to scope pressures through an agile transition would most directly enhance the firm’s adaptive capabilities [17] – i.e. its ability to respond to opportunities.

C4: Responding effectively to quality pressures through an agile transition would most directly enhance the firm’s absorptive capabilities [17] – i.e. its ability to make use of information within its internal processes (practices and roles), and thereby increase its innovative capability in producing products and services for its market.

3 METHODOLOGY

The analysis process is depicted in Figure 1. Through a longitudinal case study, we followed the progression of development practices and roles at multiple levels and sites within our global vendor over the period from 2003 to 2017 (cf. fig.1). Currently this vendor has ten development teams based in three different countries (6 teams in Australia, 3 in USA, 1 in India). They sell a complex assets management product to a global market which includes Australia, USA and Europe. Their development teams are based on the DAD philosophy with 12 software engineering personnel per team. Each team has a leadership group that consists of a product owner, team leader and a team architect. The others in each team are six software engineers and two quality assurance engineers and one technical writer. They have a separate quality assurance (QA) team led by the QA manager. This vendor has two program managers, a portfolio manager, a product management team (consisting of 4 product managers) and two organisational architects and a Director of Software Engineering (SE). They also have a marketing team and two after sales and support teams with one based in Melbourne and the other in USA.

Transcripts of interviews conducted with senior managers, software engineers, and other key roles within the organization were complemented by a set of field observations. The coding steps were based on the doctoral thesis of the first author [6], and two more recent sets of interviews and observations with the case study organisation. The first set of data was collected through a series of interviews held from 2006 to 2008. In this period a total of 38 interviews were held, (8 with the director of SE; 7 with 2 project managers; 2 with 2 principal engineers; 2 with 4 four senior SEs; 2 with 2 SEs; 1 with the QA manager; 2 with 2 two QA engineers; 1 with a technical documenter and. 1 with the marketing manager).

The second set of interviews was done in 2015 with the Director of SE. This was followed by the third set of interviews held in April 2017 (2 with the Director of SE; 1 one each with 2 senior SEs, 1 with the engineering manager, 1 with an SE, 1 with a user experience (UX) engineer/Product Owner; and 1 interview with the QA manager). In between 2008 to 2015 contact was maintained with our case study organisation through ongoing email contact and site visits for updates on their agile approach.

We mapped the evolution of practices and roles related to three key phases, (structured, hybrid-agile and DAD) during which the organisation had adapted its practices and roles in an agile transition process. This mapping applied the lightweight process mapping framework of Kirk and Tempero [6], with use of their lifecycle phase headings (*Define, Make, Deliver*) presented in Tables A1 and B1 below. As a first step we tabulated the practices and roles in use across the three critical phases of our study. Both authors worked closely together to verify and confirm by consensus the plausibility of the mappings and consistency with their earlier joint work reported in [7].

For each major transition identified in our study, we also probed into underlying drivers for change by coding issues and challenges, which were distilled into the resulting impacts, presented in figures 2 and 3. We then theorised the organisation’s progressive improvement of practices across the three phases of scaled agile transition through a dynamic capability model [17]. We mapped the key dynamic capability contributing agile practices in use at each phase. Our findings were compared against our conjectures in section 2 that the issues identified at each transition point would be addressed by improvements in relevant dynamic capabilities.

In producing Tables 1, 2 and 3, we discussed and jointly coded each practice and role combination against the primary dynamic capability to which we considered it most contributed. This avoided multi-coding, but in many cases one capability would have contributed to the building of others, e.g. the ability to respond to opportunity through enhanced adaptive capability would often have added to the organisation’s innovative capability.

Data Sources		Interviews - 1st set	observations, meetings, ongoing email contact, site visits	Interview - 2nd set	Interviews - 3rd set
	Structured [RUP]		Hybrid-Agile	DAD (Scaled Agile)	
1995		2003		2015	2017
Case Study Site Timeline					

Figure 1. Longitudinal Case Study Research Timeline

4 RESULTS AND DISCUSSION

The case study organisation had adopted the ‘agile manifesto’ [4] as a philosophy guiding its behaviour, and driving a progressive organisation wide transition towards a scaled agile framework (DAD) [1]. The developer of DAD, Scott Ambler has facilitated this scaled agile transition through an active coaching programme.

Our tabulation identified some 170 discrete practices, with a progression away from structured practices, beyond the team level towards greater adoption of agile practices at higher levels of the organisation. Complementing the practices, we also tabulated the roles in use. Here we saw some 35 roles as they evolved with the changing practices. Roles progressed from structured and highly specialised, towards what Ambler terms more “generalising specialists”¹ and then towards “cross-functional t-skilled individuals”² with depth in particular areas of

the product or technology, and breadth of skill across both, working within team settings. Extracts from these tables are presented in appendices A and B.

From our data analysis we identified three phases of transition towards a scaled agile approach, classified as firstly RUP, second a Hybrid Agile Method (Hybrid Agile) and thirdly Disciplined Agile delivery (DAD). We analysed two critical junctures covering the transitions from 1) RUP to Hybrid Agile and 2) Hybrid Agile to DAD. This analysis consisted of reviewing the interview transcripts aiming to identify issues that led to the need for changes in software development process, and assigning these identified issues to the transition in question. The two Fishbone diagrams (Figure 1 and Figure 2) (drawing on the approach of [15]) relating to these two crucial market-driven eras (1995- 2002 and 2003 to 2015), identify the critical issues and challenges impacting on the vendor’s ability to deliver their software product.

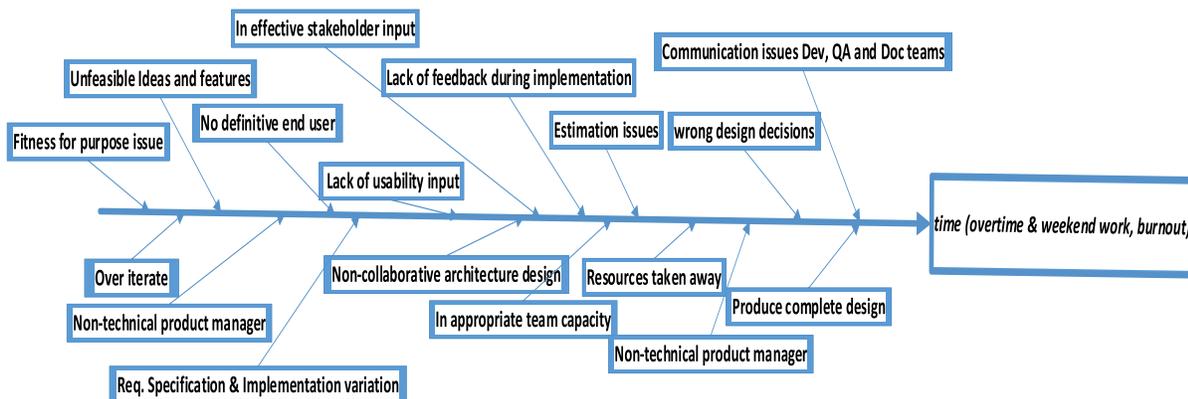


Figure 2. Cause-effect diagram for RUP to Hybrid Agile Transition (1995-2002)

Figure 2 highlights the time pressures facing the organisation. Inappropriate practices and roles inhibited the company’s ability

to respond and deliver in a market driven environment. In the RUP era for this global software vendor, their software

¹ <http://www.agilemodeling.com/essays/generalizingSpecialists.htm>

² <https://dsi-dev.org/wp-content/uploads/2017/11/p1291641.pdf>

engineering team felt frustration due to the development issues they were facing-

“we had a lot of people saying hey this doesn't feel right. Like we're spending a lot of time here ... when we were doing lots of specs” I am spending a lot of time in other activities. Whether it be loading specs or meetings”.

Overly structured practices and roles prevented cross-functional collaboration. Procedures and rules for creating linear design and documentation led to inability to accommodate change, and get effective and timely stakeholder input and feedback. To maintain their good delivery effort (on-time and

quality product), projects took longer to deliver unexpected extra work and required more resources. This was achieved through development teams working overtime (long hours and weekends) usually in the backend of the projects.

“The difference is the stress levels... in some of the days when we have got into really tough situations it has been really stressful... get a quality result out... have done that ... it takes its toll and you get burnt out.

This unsustainable situation led the organisation to transition towards an agile approach.

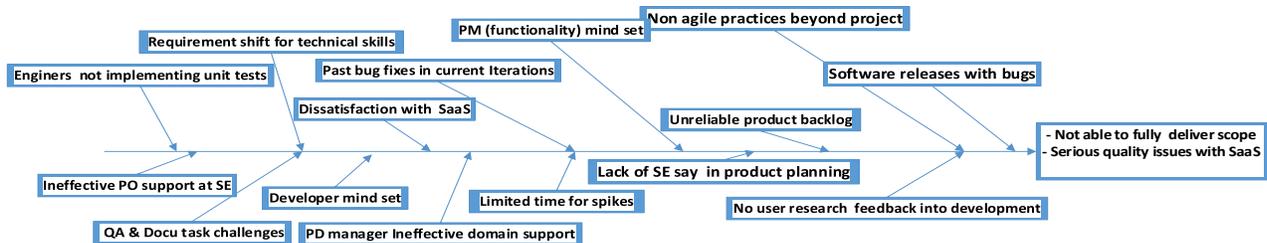


Figure 3. Cause-effect diagram for Hybrid Agile to DAD Transition (2003-2015)

Figure 3 highlights the scope and quality pressures facing the organisation. With this software vendor, DAD adoption was driven by quality & scope issues experienced by their clients with their SaaS offering. They discovered that the customer expectation in the cloud space is quite different from the expectations of the on-premises clients.

“Quality was not to the level that we need it to be. But on the cloud, if it's not performing to the level of what they used to in using Facebook and everything else they're using in the internet, then it's like this is a performance problem. What we noticed that as we got more and more customer, that voice of saying “hey, performance actually is not where it needs to be”

Scaling-up via DAD provided the opportunity to refine their overall development process and be more skilled as a software vendor in developing and delivering the right high quality products and features for the marketplace. They had fully embraced the core principles of the agile manifesto since their agile adoption in 2003 and transition to DAD approach with recommended practices was a logical organization wide extension.

“Do a high quality job and also as the business be able to respond more quickly with more frequent releases, and that just all the benefits of being agile, it helped us to refine and be more skilled at the execution of what we need to do and it's really helped the team be empowered even more”.

The DAD adoption was top- down driven through their Senior Vice President Engineering. The DAD value for the “organisational context to be enterprise aware” was the key motivation for this method. The DAD method emphasises internal culture based on development teams and other functional units being outwardly and market needs focused. Importantly, DAD actually brings the business level planning (business strategy, program and product planning) and software engineering

planning (project and development) together, aligning the two with each other.

“DAD pushed it in an enterprise context. Agile turns developers into the center of the universe, just do development and iterations, and everyone else just deal with whatever that means. There's a whole DAD aspect that goes into roadmap planning, product planning, and strategic planning”

The approach taken here for DAD transition was that the consultants from USA were brought to train and coach the engineering unit at Box Hill. Hence, with this approach the transition went smoothly with significant structural changes adopted within the engineering unit.

“We got Scott Ambler here to train us. we probably wouldn't have come close to succeeding without having somebody stepping us through it in the real world. Because doing the training is one thing, but applying it to your project is another thing.”

The critical junctures highlighted in Figures 2 and 3 above, indicate the classic iron triangle of time, scope and quality pressures facing the organisation and driving change. The drivers for change in each of these transitions, led us to add the three conjectures (C2-C4) relating to time, scope and quality in section 2.

In the set of three tables below, building on the tables in Appendix A and B, we have mapped our three dynamic capabilities (namely adaptive, absorptive and innovative) to practices and roles for each method and period, applying the define, make and deliver phases of the lightweight practice framework of [6]. The tables also include the organizational level and site/location addressing the global and local dimension and operation. In this process we identified each practice or role that we deemed to contribute to a dynamic capability, and elicited the positive and negative impact of each.

TABLE 1. Dynamic Capability Enabling Practices and Roles Assessment (Adaptive)

	Phase/category	Practice	Organisational level	Roles	Site/location	Negative/Positive (-ve/+ve)
Adaptive	RUP- Define	Ideas gathering	Product management	Traditional Product managers	Global- US & Australia	-ve unfeasible implementation ideas +ve high quality (functionality)
		Req. specification	SE- Development teams	Principle & senior engineers	Local	-ve, upfront, lengthy review, implementation delays Restricts, innovation and creativity
	Hybrid - Define	Vision planning	Product management	Technically knowledge Product managers	Global- US	-ve Functionality driven, little consideration on end-user experience. +ve Internal and external collaboration and input to identify feasible reqs. +ve fully justified and prioritised high level req.
		Roadmap planning	Product Management	Steering committee & Product strategy manager	Global- US	+ve collective decision making +ve visibility organisation wide for market releases (for a set period) +ve aligns future business opportunities with SE capacity +ve clearly identify high level req. (mostly 1- 3 months) enables short duration projects -ve SE no input on priority setting, purely market-driven or executive decision
	DAD – Define	Product planning – propose new features	Product management	Product managers	Global- Australia	+ve ability to understand long term needs and requirements +ve reliable and achievable ideas proposed for development +ve solid understand of market needs +ve establish relationship with all external stakeholders +ve fully focussed on PM
		Portfolio planning	Portfolio management	Portfolio manager	Global- Australia	+ve business values features +ve feasible features +ve market-driven priority priority
		Program planning	Program management	Program manager	Global- Australia	+ve swiftly deliver features +ve highest priority, full business support for implementation

Adaptive- taking the opportunity

RUP practices-

RUP was seen to enable adaptive capabilities across our Define practice categories (Appendix A & B). Specific practices of idea gathering and requirements specification had a positive element in leading to high quality functionality, but negatively with upfront lengthy reviews and infeasible ideas reaching development teams causing waste and delays. The positioning of the product managers as effectively global lone rangers making top-down decisions contributed to this ineffectiveness.

Hybrid Agile practices-

The Hybrid method enabled adaptive capabilities again across our Define practice categories (Appendix A & B). Specific practices of vision and roadmap planning, with global collective decision making enabled well thought-out business value functionality for implementation. However, collective decision-making did not involve all stakeholders. For end users, usability

and user experience issues were not part of these two practices, nor did software engineers have input into priority setting for high-level requirements.

DAD practices-

DAD methods brought a better approach across our Define practice categories (Appendix A & B), through several practice-role combinations. Product planning, portfolio planning, and program planning practices combined with multi-stakeholder input and a collaborative partnership between business management and software engineering, led to identifying long term needs, reliable and achievable ideas for development, a solid understanding of market needs, fully focused on product management. The shift of responsibilities from the sales-driven US headquarters to the engineering focused Australian operation encouraged a better link with the necessary development skills and more collaborative planning incorporating both top-down and bottom-up elements for global software development.

TABLE 2. Dynamic Capability Enabling Practices and Roles Assessment (Absorptive)

Absorptive	Phase/category	Practice	Organisational level	Roles	Site/location	Negative/Positive (-ve/+ve)
	RUP- Define	Upfront design models	Project management	Principle & senior engineers	Local	+ve Vision for the entire project for developers, testers & technical writers -ve lack flexibility for change
	RUP- Define	Project plan	Project management	Project manager	local	-
	Hybrid- Define	Product backlog	Project management	product development manager	local	+ve identified the work to be done in a project -ve only had new development work, no fixes or any other work
	Hybrid- Define	Project plan	Project management	Project manager/ product development manager/ engineering manager	Local	-ve team view only +ve flexibility
	Hybrid- Make	Daily stand up meetings Reflection meeting	Project management	product development manager	local	+ve visibility
	DAD- Define	Story boarding	Portfolio Management	Product owners	Global	+ve focus on value creation for customers and stakeholders rather than just on functionality +ve quality driven
	DAD- Define	Project plan	Program management	Project management (PO, Team leader, team architect)	Local	+ve minimise team overhead
	DAD- Define	Release plan	Portfolio management	Portfolio manager	Global	+ Strategic product management
	DAD- Define	Marketing plan	Product management	Product manager	Global	+time to market
	DAD- Define	Work item list - program	Program management	Program management (program manager, product owner, enterprise architect.	Global & local	+ve complete picture of program and individual project work +ve includes bug fixes +ve enhanced coordination and visibility of projects +ve quality driven +ve common culture & practices +ve risk driven development +ve self-contained project teams +ve shared project leadership +ve Product owners drive development +ve achievable project plan +ve create needed technical skills
	DAD- Make	-Daily tactical huddles	Program management	Program manager	Global	+ve project status on daily basis
	DAD- Make	-Iteration release show/tell practice	Program management	Product owner	Global	+ve regularly showing progress through working code and getting feedback on implementation from stakeholders

Absorptive – using information effectively

RUP practices-

RUP was seen to enable absorptive capabilities across our Define practice categories (Appendix A & B). Specific practices of up front design and project plan gave visibility for the entire project across the team, but at the cost of inflexibility in coping with change.

Hybrid Agile practices-

The hybrid method enabled absorptive capabilities across our Define and Make practice categories (Appendix A & B), through several practices, development and project plans built from the product backlog, providing organisational visibility, and daily

standups and reflection meetings providing project visibility and practice improvements.

DAD practices-

DAD methods contributed to absorptive capabilities across our Define practice categories (Appendix A & B), through practices addressing both focus and planning/monitoring; namely focus on value creation (not just functionality) through story boarding leading to quality driven development, supported by release plans, marketing plans, work item lists. The standardization of practices and unifying of cultures across both global and local teams better enabled the global scaling of software development across multiple sites.

TABLE 3. Dynamic Capability Enabling Practices and Roles Assessment (Innovative)

Innovative	Phase/category	Practice	Organisational level	Roles	Site/location	Negative/Positive (-ve/+ve)
Innovative	Hybrid- Define	Business case	Vision planning	Product manager & steering committee (decision making)	Global	+ve collective decision making +ve, visibility organisation wide +ve clearly identify high level req. (mostly 1- 3 months) enables short duration projects -ve SE no input on priority setting, purely market-driven or executive decision
	DAD- Define	Feature funnel	Portfolio management	Portfolio manager	Global	+ve creates certainty for new development for a long period of time (over 10 years) +ve focus on value creation for customers and stakeholders rather than just on functionality +ve invest in developing functionality that will provide value to the organization soon
	Hybrid- Make	Projects independently carried out at three different countries	Project management	Project manager/product development manger	Local	+ve concurrent development of multiple functionalities -ve skill allocation challenges for high value implementations and fixes
	Hybrid- Make	2-week Iteration Cycles Iteration planning (re-estimate) Iteration delivery-team commitment	Project management	Project manager/product development manger	local	+ve confidence and certainty for regular (in short cycles) emergences of working code
	Hybrid- Make	Solo effort TDD (unit test) Refactoring Peer code reviews	Project management	Project manager/product development manger	local	+ve upfront ability to deal with quality
	Hybrid- Make	2weeks QA and documentation iterations Integration test system test regression test performance test acceptance test Core reviews	Project management	Project manager/product development manger	Local	-ve Regular external release of features
	DAD- Make	Projects through 10 global development teams Multi-sites	Program management	Program manager	Global	+ve enhanced & efficient utilisation of development capacity
	DAD- Make	3 phases: inception phase (2 weeks), implementation phase (2 weeks iteration cycles) and hardening phase (2 weeks).	Project management	Team leadership (Product Owner, AEO, TL)	Global	+ve continuous deliver in a shorter time frame.
	DAD- Make	Self-organising teams Primary & secondary roles Leadership roles T-skilled individuals	Project management	Team leadership (Product Owner, AEO, TL)	Global	+ve enforces &enhances collective responsibility for delivery Enhances collective effort & quality work Effective coordination of teaks +ve self-contained team for spontaneous collaboration
	DAD- Make	risk mitigation Technical spikes Definition of done (DOD)	Project management	Team leadership (PO, AO, TL)	Global	+ve deal with risk upfront & create certainty +ve bug free internal releases, enhance quality
	DAD- Make	Collective or solo effort (prog)	Project management	Team leadership (PO, AO, TL)	Global	+ve high productivity better quality work
	DAD- Make	Continuous delivery systems testing regression testing performance testing acceptance testing	Project management	Team leadership (PO, AO, TL)	Global	+ve ready for product testing in the simulated product environment
	DAD- Make	Beta testing	Program management	Program manager	Global	+ve feedback from the customers before market and product environment release for
	RUP- Deliver	Big bang release-packaged	Product management	Product manager	global	-ve team burnout -ve patch releases (sometimes major)
	RUP& Hybrid-Deliver	Support (packaged software) - 24 hour after sales, installation support, long term maintenance & enhancement support, customer bug reporting	Product management & software engineering	Product manager	Global	+customer relationship management

	Hybrid-Deliver	Regular releases-packaged and SaaS (every 3 months)	Product management	Product manager	global	+ve strategic releases
	DAD- Deliver	Regular releases (packaged- every 3 months) & continuous deployment (SaaS- 2 weeks)	Program management	Program manager	global	+ve strategic releases
	DAD- Deliver	Support (SaaS)- SAS deployment-continuous monitoring, enhancement, maintenance	Program Management	Program manager	Global	+ve efficient & stable operations environment

Innovative – new product development

RUP practices-

RUP enabled innovative capabilities across our Deliver practice category (Appendix A & B). Specific practices of big bang release packages and packaged software support contributed to predictable and substantial product deliveries, but at the cost of team burnout and occasional need for major patch releases, but positively to the ability to manage customer relationships.

Hybrid Agile practices-

The Hybrid method enabled innovative capabilities across our Make and Deliver practice categories (Appendix A & B). Specific Make practices of regular delivery of working code, TDD, and multi-site (3) development supported the ability to concurrently develop multiple pieces of high quality functionality, but with resulting challenges in allocating developer resources for bug fixing and enhancements. Deliver practices through regular releases of packaged and SAAS software and packaged software support enabled strategic releases to be made on demand, and again contributed positively to the ability to manage customer relationships.

DAD practices-

DAD methods brought improvements across our Define, Make and Deliver practice categories (Appendix A & B), through several practice-role combinations. Define practices include feature funnel, self-organising teams, primary and secondary team roles, T-skilled individuals. These contributed the ability to innovate through providing: early scrutiny on ideas for development at the global level of portfolio management; the necessary level of team autonomy and the inclusion of multiple perspectives and skills both globally across the organisation and locally within the teams.

Make practices included multi-site (10) development teams which supported the ability to concurrently develop multiple pieces of high quality functionality across additional sites and the whole product, including bugs and enhancements. DOD (definition of done) was a project governance practice that ensured zero bug iteration in short cycles for high quality. User stories, User acceptance tests and unit tests together with pair programming and continuous delivery practices are also global DAD practices for enhancing quality and enabling product innovation. These also supported a consistent approach across distributed teams enabling global software development.

Deliver practices included continuous deployment for testing in a subset of the production environment. DAD builds on the

Hybrid methods strategic market releases to further enhance the Innovative capability of the organisation with continuous support for packaged software and SAAS, monitoring, enhancement and maintenance, for global releases.

Global and local dimensions

In the course of this global software vendor’s transition to a scaled agile implementation, we saw an intriguing shift in the balance between local and global dimensions. In the initial relatively linear RUP development cycle, global product management drove the direction but parcelled work out to two autonomous engineering sites (US and Australia, with product management headquartered in the US). The hybrid-agile phase in which the company changed ownership and added a third development site (India), saw, through the introduction of agile practices, some engineering team consultation about product direction, and increasing team autonomy, the ability to concurrently develop across the three sites. In the later DAD phase the agile progression continued, but with increasing collaboration between global product management (now in Australia) and local engineering teams in determining product direction. Software engineering had now become more integral to the planning process. The agile practices and roles of DAD introduced standardisation within development teams, and a broad shared blueprint for the product direction. Thus, software development within the organisation became plan driven ‘in the large’ and ‘iterative in the small’. This enabled the increasingly autonomous (and now ten development teams across the three sites) to work concurrently towards a shared endeavour.

Summary

Through the above, we see the link between a scaled agile progression in one software vendor, and the progressive enhancement of the ‘dynamic capabilities’ of the firm, which supports our overall conjecture, C1: that dynamic capabilities of software vendor organizations would be enhanced as a scaled agile transition progresses.

Surprisingly, in our analysis, agile practice and role combinations contributed not solely to the development of absorptive capabilities through improved internal process capability and information management, but most significantly in the adaptive and innovative capabilities, which are critical for a market-driven software firm. In section 2 we had proposed three further conjectures, namely that:

C2: Responding effectively to **time pressures** through an agile transition would most directly enhance the firm’s **adaptive capabilities** [17] – i.e. its ability to respond to opportunities.

Finding: Supported (cf. table 1 mapping for each finding), but also addresses **time** in the **Innovative capability** through Table 3 *deliver practice regular releases*

C3: Responding effectively to **scope pressures** through an agile transition would most directly enhance the firm's **adaptive capabilities** [17] – i.e. its ability to respond to opportunities.

Finding: Supported, but also addresses scope in the **Innovative capability** through Table 3 *define practice feature funnel*

C4: Responding effectively to **quality pressures** through an agile transition would most directly enhance the firm's **absorptive capabilities** [17] – i.e. its ability to make use of information within its internal processes (practices and roles), and thereby increase its **innovative** capability in producing products and services for its market.

Finding: **Absorptive** capability enhanced supported, but also addresses quality in the **Innovative** capability through Table 3 define practices including feature funnel, self-organizing teams, continuous deployment etc.

We note that these insights expand on the conclusions of Mithas et al., regarding the positive contribution of information management capability to customer management, performance management and process management capability of the firm [11]. Intriguingly we would see information management capability as broadly equating to absorptive capability in our study, and with a much stronger link to process management capability. So while our capability grouping has some commonality with [11] our findings also diverge, which warrant further investigation.

A further promising avenue for research is the direction of the relationship between dynamic capabilities and agile transformation. As asked by one of the anonymous reviewers, did the dynamic capabilities held by the organization enable their transformation or did the transformation endow the organisation with new dynamic capabilities? While we have argued for the latter in this early investigation, we suspect they are in reality mutually reinforcing.

Limitations

As a single (albeit longitudinal), case study, the generalisability of these findings can be argued. Yet, we believe that many software vendors are engaging in a similar process of change through introducing a scaled agile framework. We imagine the drivers for that transition to be similar. Therefore, we believe these findings may be of value to other practitioners and lead the way for future research.

To establish evidence in the paper, transcripts were coded to support the mapping exercises. The first author performed the majority of the mapping, but construct and internal validity were strengthened by the first author's familiarity built from long term engagement with the organization, over some 12 years, cf. [8, 7]. The mappings were reviewed by the second author (who had limited exposure to the case study site) and we discussed them extensively to reach consensus over their validity and plausibility. In mapping the dynamic capabilities (adaptive, absorptive, innovative) we looked for the capability to which we believed the practices most strongly contributed. Individual assignments may be argued, but we believe that the broad logic of the process and findings holds. As further confirmation, the first author had also discussed several of the issues and drivers for change with the company management on a recent visit to Australia.

As a deliberately broad ranging and high level study we have addressed three major phases in a long term global scaled agile transition spanning a 15-year period. While we have identified enhancements in dynamic capabilities across this period, no doubt other process changes and their intermediate outcomes (such as how teams were configured and how tasks were organized) may also have been in play. While we have investigated the evolution of practices and roles, a more fine-grained analysis may have identified those potentially confounding factors, but that would be the subject of a further study. For instance, this paper has not focused extensively on changing roles and their implications, with them mostly being embedded in the accompanying practices. This is an area that could be further investigated.

As a final limitation we have not included detailed metrics on performance improvement results of the vendor, (such as in 3,14), over our timeline. We did not have access to sensitive company performance data, and we did not have a chance to collect full process improvement data but this could be an area for a subsequent study.

CONCLUSION

In this longitudinal case study, we demonstrate one global software vendor's transition towards an agile enterprise. This transition has brought about changes in practices, responsibilities, distribution of development sites and teams, and in roles across the organisation. We argue that drivers for these changes were the organisation's desire to address issues related to timely delivery, and managing scope and quality of releases, to better serve a global customer base and their evolving needs.

While those may have been the intuitive drivers from a company perspective, we have theorised the transition towards a scaled agile organisation in the light of improving "dynamic capabilities". We found support for our conjectures about the links between agile practices and roles and dynamic capabilities, and argue that this provides a useful framing for considering scaled agile initiatives.

Importantly we found that scaled agile practices contributed to organisational capability in significant ways that transcended simple internal process capability improvement. Improved ability of our case study firm to anticipate and respond to external pressures, and to innovate, appear to have arisen from their deeper and broader adoption of agile practices. These findings may go some way to explaining why and how agile practices are taking hold and scaled agile transitions are becoming more frequent in software vendor organisations.

ACKNOWLEDGEMENTS

We must acknowledge the work of our reviewers who greatly improved the paper, and the support for this study provided by our case study site.

REFERENCES

- [1] Ambler, S.W., and Lines, M.: 'Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise' (IBM Press, 2012)
- [2] Beck, R., Pahlke, I., and Vykoukal, J.: 'Colocation as a Hybrid ICT Sourcing Strategy to Improve Operational Agility', ACM SIGMIS Database, 2016, 47, (2), pp. 9-35
- [3] Harter, D. E., Kemerer, C. F. and Slaughter, S. A. Does software process improvement reduce the severity of defects? A longitudinal

field study. *IEEE Transactions on Software Engineering*, 38, 4 (2012), 810-827.

[4] <http://agilemanifesto.org/>.

[5] Khan, A. A., Keung, J., Hussain, S., Niazi, M. and Tamimy, M. M. I. Understanding software process improvement in global software development: a theoretical framework of human factors. *ACM SIGAPP Applied Computing Review*, 17, 2 (2017), 5-15.

[6] Kirk, D., and Tempero, E.: ‘A lightweight framework for describing software practices’, *Journal of Systems & Software*, 2012, 85, pp. 582-595

[7] Lal, R., and Clear, T.: ‘Scaling Agile at the Program Level in an Australian Software Vendor Environment: A Case Study’: ‘Australasian Conference on Information Systems’ (2017), pp. 1-12

[8] Lal, R.: ‘Strategic factors in agile software development method adaptation: a study of market-driven organisations’. PhD Thesis, Massey University, 2011

[9] Larman, C. and Vodde, B. *Large-scale scrum: More with LeSS*. Addison-Wesley Professional, 2016.

[10] Leffingwell, D.: ‘SAFe® 4.0 Reference Guide: Scaled Agile Framework® for Lean Software and Systems Engineering’ (Addison-Wesley Professional, 2016)

[11] Mithas, S., Ramasubbu, N. and Sambamurthy, V. How information management capability influences firm performance. *MIS quarterly* (2011), 237-256.

[12] Paasivaara, M.: ‘Adopting SAFe to scale agile in a globally distributed organization’, in Sarma, A., and Cruzes, D. (Eds.): ‘Proceedings of the 12th International Conference on Global Software Engineering’ (IEEE Press, 2017), pp. 36-40

[13] Prahalad, K., & Hamel, G. (1990). The core competence of corporation. *Harvard Business Review*, May-June, 1-15

[14] Ramasubbu, N. Governing software process improvements in globally distributed product development. *IEEE Transactions on Software Engineering*, 40, 3 (2014), 235-250.

[15] Raza, B., Clear, T., and MacDonell, S.: ‘Continuous Transition in Outsourcing: A Case Study’, in Cruzes, D., and Sharma, A. (Eds.): ‘Proceedings 2017 IEEE 12th International Conference on Global Software Engineering’ (IEEE, 2017), pp. 41-50

[16] Teece, D.J., Pisano, G., and Shuen, A.: ‘Dynamic capabilities and strategic management’, *Strategic management journal*, 1997, pp. 509-533

[17] Wang, C.L., and Ahmed, P.K.: ‘Dynamic capabilities: A review and research agenda’, *International journal of management reviews*, 2007, 9, (1), pp. 31-51

Appendix A

A.1 Practices and role mapping tables

Excerpts from our practices and role mappings are included here in Tables A1 and B1. Due to space limitations we provide full details of the Tables in a supplementary Google document at https://docs.google.com/document/d/1Hg0s3ha_AYdXzTWocZBuVRphHGteISFemvOSAAdV4gDM/edit?usp=sharing

Table A1. Practices in used across phases, methods and periods.

Practice Category	Method & Period	Practice Sub-Category	Practice in Use by sub-category	Practice In Use Across Categories
Define	RUP (prior 2003)	Requirements elicitation	-Ideas and requirements gathering	Procedures and rules Stakeholder input and feedback Formal documentation & sign offs
		Requirement's specification	-Create specification", analysis models (use case modelling) ,testing plan documentation plan, change management, risk and issue management plans) -Request written feedback from key stakeholders -Colour code (feedback), update and iterate until consensus achieved	Individual ownership Team work based on solo effort Functional responsibility Big bang approach
	Hybrid Agile (2003-2015)	Vision planning	Identify potential features (high level requirements), Requirements culling, Business case, Product planning	Stakeholder collaboration Cross-functional collaboration Generalist skills set Team effort
		Portfolio planning	Prioritise high-level requirements.	Collective decision making Team ownership Empowerment
		Project planning (Design phase)	Product backlog planning- use case- tasks, estimates (planning poker), prioritised UCD approach (personas, scenarios and prototypes),	
	DAD (2016-till present)	Product planning	<i>Product management</i> Propose business & product roadmap plan, Business case, Propose new features for development, Create marketing plan, Release features	Stakeholder collaboration and feedback Cross-functional collaboration Collective effort
		Portfolio planning	<i>Portfolio management</i> Feature funnel practice, Manage product portfolio- continuously prioritise high level requirements, Create visibility organisation wide, Release planning (roadmap plan)	Collective decision making Team ownership Empowerment
		Program planning (Design)	<i>Program management</i> Story boarding , Work item list, Project planning	Knowledge management Empowering leadership and management expertise T-skilled individuals

Appendix B

Table B1. Roles in use across phases, methods and periods

Practice Category	Method & Period	Practice Sub-Category	Roles in Use by sub-category	Team/role responsible in a sub-category
Deliver	DAD (2016 – till present)	Support	Product managers, Sales engineers, End-user consultants, Software engineers, QA engineers, Independent QA, Customer success engineers	Program management