Agile in-the-large: Getting from Paradox to Paradigm

Kelsey van Haaster[#1]

School of Computing and Mathematics,
Charles Sturt University, Bathhurst Australia
[1]kvanhaaster@csu.edu.au

ABSTRACT

There is a gap in the research surrounding Agile software development. Whilst approaches to implementing Agile-in-the-small have an evidenced track record of success, approaches to implementing Agile in the large do not. This paper shows that whilst we understand many of the challenges of large scale Agile, we do not yet fully understand how these challenges should be addressed. This is a significant problem as increasingly; organisations embark on a program of broad organisational change, guided by emergent frameworks.

***Keywords* - Agile, Software-development, research-gap**

INTRODUCTION

Agile, in the context of software development, refers to a set of software development practices, methods and approaches. Developed and refined over many years, these practices, which include; Scrum, Extreme Programming (XP) and Crystal [1], may be grouped together under a common banner of lightweight methods [2]. Influenced by the ideas and principles of Lean manufacturing [3], lightweight methods have been developed with the objective of improving the processes and products of software development; and whilst different in detail, share the characteristic of being both iterative and incremental in their approach. In 2001 the designers of some of these methods met with experts in various other aspects of software development in order to seek a common understanding of the characteristics of lightweight methods and to discuss what made them successful. As a result of this meeting, the Agile Manifesto [4] was published; never intended to be a rule book, the Agile Manifesto describes a set of values and underlying principles which guide the software development process through the use of lightweight methods. Taken together, lightweight methods combined with the values and principles expressed in the Agile Manifesto propose a way of thinking about how best to successfully design and build a digital product.

Agile values and principles have a close philosophical alignment to those which underpin Lean [5], an approach to manufacturing often associated with the highly successful Toyota Manufacturing System [6]. Lean focuses on empowering people to achieve the best results, through improving efficiency and eliminating waste. Lean, as an approach, targets the enterprise whereas Agile is more tightly focused on the individual team or project. Agile and Lean are complementary and have been proven to deliver better outcomes, lower costs and reduce the risks associated with software development, at least within the context of individual software development teams [7].

SCALING AGILE

Given the somewhat dismal figures for the success of traditional software development projects in general [8], it is probably not surprising that Agile and Lean approaches to doing business are of great interest to the commercial software development industry as a whole. However, adopting Agility at an enterprise level implies a significant cultural and organisation change which impacts roles and responsibilities, corporate governance mechanisms, reporting mechanisms, approaches to corporate planning and financial planning, marketing, sales forecasting and public relations; as well as demanding new and different conversations with stakeholders, shareholders and the user community. More than a decade of research has resulted in a substantial body of literature highlighting these challenges as well as the practical difficulties of scaling Agile in-the-large, which include; managing variability amongst team processes, lifecycles and approaches to developing and managing requirements [9], ([10], [11], [12], [13], [14], [15].

Many organisations which have attempted to introduce or to scale Agile have subsequently learned that the implications of adopting Agility are far broader than just replicating the approaches and activities which have been successful for individual teams. Whilst a number of high profile digital organisations have successfully adopted Agile as whole of business approach [16], [17]. These success stories are not the majority, so whilst corporate interest continues to grow, so does concern about the risks and challenges implicit in attempting to initiate such broad corporate change. Consequentially, organisational governance teams are seeking assurances about the potential costs and benefits of doing so [18] .

Frameworks for Agile in-the-large

In response to this need a number of frameworks which claim to provide a pathway to Agile-in-the-large have emerged; amongst the better known of these are DaD (Disciplined Agile Delivery), [19], LeSS (Large Scale Scrum) [20] ScrumPLoP (The Scrum Pattern Language group) [21] and most recently, SAFe (Scaled Agile Framework) [22]

DaD (Disciplined Agile Delivery) is proposed by Scott Ambler, and is an evolution of his work on the EUP (Enterprise Unified Process) [23] which in turn extends the RUP (Rational Unified Process), [24]. DaD proposes a hybrid approach to scaling Agile which incorporates strategies from a range of lightweight methods and Lean principles as well as mandating a set of prerequisite core practices. DaD claims to extend Agile principles across the system lifecycle so providing approaches to managing challenges such as geographically diversified delivery teams, complex organisational structures and multiple technology platforms [25].

LeSS (Large Scale Scrum) championed by author Craig Larman, [20] and ScrumPLoP (The Scrum Pattern Language group) founded by Jeff Sutherland, one of the authors of the Agile Manifesto and inventors of Scrum [21] both focus on scaling Scrum from the perspective of the organisational software engineering function. Neither framework appears to directly address issues associated with corporate or software governance in the same way as DaD or SAFe.

SAFe is the most recent of the Agile in-the-large frameworks and is based on the work of Dean Leffingwell [26]. SAFe proposes a three-tiered approach to scaling Agile, which address the needs of the team, the program and the portfolio.

SAFe is the first Agile in-the-large framework to offer a complete package of supporting software, certification schemes, books and training courses driven by an intensive marketing effort [27]. SAFe claims to provide a proven approach to scaling Agile, a claim supported by a number of customer case studies. A number of software vendors have developed tools designed to support SAFe through both their product and through training and certification. SAFe has been the topic of some particularly heated discussions found in the technical literature. These highlight strong concerns voiced by many well-respected authors, practitioners and thought leaders. Who whilst acknowledging that one of the strengths of SAFe is its basis in sound lean principles have raised concerns about whether the framework actually supports the principles and values which underpin Agile, or whether it undermines them [28]. A focus on standardisation, large scale planning and taking a top-down approach are also highlighted as potential problems, primarily due to the apparent lack of focus on people rather than process [29]. SAFe claims to present approach to scaling Scrum, yet Ken Schwaber one of the designers of Scrum argues that SAFe is based on RUP rather than Scrum [30]. Ron Jeffries one of the original designers of XP also highlights the centralised approach to planning dictated by SAFe as an issue [31].

Conclusion

Agile-in-the-large is not well researched; the challenges are well understood but approaches to solving them are emergent and potentially immature. Frameworks such as DaD and SAFe propose an approach and are associated with an enormous amount of supporting material, including books, training courses, certification schemes, supporting software products and the potential for fortunes to be made via consulting fees. However unlike Agile implementations at the team level there is very little empirical evidence in the substantive literature that claims these approaches are proven. This is a matter for concern as organisations struggle to leverage the perceived benefits of Lean and Agile approaches, whilst mitigating a myriad of well-documented inherent risks.

Addressing this gap in the research is an important opportunity for researchers; as well as an urgent need for organisations embarking on large, complex or critical projects using Agile methods. This need is supported amongst the research community [1] who point out that whilst there exists a large and diverse body of work covering aspects of Agile software development, much of this work focuses on Agile practices and methods in-the-small and by [32] who identify an urgent need for a research agenda in large scale Agile.

References

1. T. Dingsøyr, et al., “A decade of agile methodologies: Towards explaining agile software development,” *Journal of Systems and Software*, vol. 85,

no. 6, 2012, pp. 1213 - 1221; DOI http://dx.doi.org/10.1016/j.jss.2012.02.033.
2. A. Cockburn, “Alistair.Cockburn.us | Balancing lightness with sufficiency,” 2014; http://alistair.cockburn.us/Balancing+lightness+with+sufficiency.
3. “Lean Systems Society,” 2012; http://leansystemssociety.org/credo/.
4. K. Beck, et al., *The Agile Manifesto*, 2001.
5. M. Fowler, “AgileVersusLean,” *Book AgileVersusLean*, Series AgileVersusLean 2014, ed., Editor ed.^eds., 2008, pp.
6. TOYOTA, “Production System,” 2012.
7. R. Gourly, “Technical Debt and Interest,” *Book Technical Debt and Interest*, Series Technical Debt and Interest, ed., Editor ed.^eds., 2013, pp.
8. P. Adamczyk and M. Hafiz, “The Tower of Babel did not fail,” *Book The Tower of Babel did not fail*, Series The Tower of Babel did not fail, ed., Editor ed.^eds., ACM, 2010, pp. 947-957.
9. B. Boehm and R. Turner, “Management challenges to implementing agile processes in traditional development organizations,” *Software, IEEE*, vol. 22, no. 5, 2005, pp. 30-39; DOI 10.1109/MS.2005.129.
10. S.C. Misra, et al., “The Organizational Changes Required and the Challenges Involved in Adopting Agile Methodologies in Traditional Software Development Organizations,” *Proc. Digital Information Management, 2006 1st International Conference on*, 2007, pp. 25-28.
11. A. Mahanti, “Challenges in Enterprise Adoption of Agile Methods -- A Survey,” *Journal of Computing & Information Technology*, vol. 14, no. 3, 2006, pp. 197-206; DOI 10.2498/cit.2006.03.03.
12. G. van Waardenburg and v.V. Hans, “When agile meets the enterprise,” *Information and Software Technology*, vol. 55, no. 12, 2013, pp. 2154 - 2171; DOI http://dx.doi.org/10.1016/j.infsof.2013.07.012.
13. C. Rand and B. Eckfeldt, “Aligning strategic planning with agile development: extending agile thinking to business improvement,” *Proc. Agile Development Conference, 2004*, 2004, pp. 78-82.
14. K. Logue and K. McDaid, “Agile Release Planning: Dealing with Uncertainty in Development Time and Business Value,” *Proc. Engineering of Computer Based Systems, 2008. ECBS 2008. 15th Annual IEEE International Conference and Workshop on the*, 2008, pp. 437-442.
15. Heikkila, et al., “A Revelatory Case Study on Scaling Agile Release Planning,” *Proc. Software Engineering and Advanced Applications (SEAA), 2010 36th EUROMICRO Conference on*, 2010, pp. 289-296.
16. K. Power, “The Agile Office: Experience Report from Cisco's Unified Communications Business Unit,” *Proc. Agile Conference (AGILE), 2011*, 2011, pp. 201-208.
17. P. Saddington, “Scaling Agile Product Ownership through Team Alignment and Optimization: A Story of Epic Proportions,” *Proc. Agile Conference (AGILE), 2012*, 2012.
18. J. Pernstål, et al., “The lean gap: A review of lean approaches to large-scale software systems development,” *Journal of Systems and Software*, vol. 86, no. 11, 2013, pp. 2797 - 2821; DOI http://dx.doi.org/10.1016/j.jss.2013.06.035.
19. S. Ambler, “Disciplined Agile Delivery,” 2014; http://disciplinedagiledelivery.com/.
20. “Craig Larman,” 2014; http://www.craiglarman.com/wiki/index.php?title=Main_Page.
21. J. Sutherland, “Scrum Pattern Community,” 2014; http://www.scrumplop.org/.
22. “Scaled Agile Framework,” 2014; http://scaledagileframework.com/.
23. S. Ambler, “Enterprise Unified Process (EUP): Strategies for Enterprise Agile,” 2014; http://enterpriseunifiedprocess.com/.
24. “IBM Rational Unified Process (RUP),” 2014; http://www-01.ibm.com/software/rational/rup/.
25. A.W. Brown, et al., “Agility at scale: economic governance, measured improvement, and disciplined delivery,” *Book Agility at scale: economic governance, measured improvement, and disciplined delivery*, Series Agility at scale: economic governance, measured improvement, and disciplined delivery, ed., Editor ed.^eds., IEEE Press, 2013, pp. 873-881.
26. D. Leffingwell, “Dean Leffingwell,” 2014; http://deanleffingwell.com/.
27. P. Saddington, “The Scaled Agile Framework (SAFe) - A Review | Agile ScoutAgile Scout,” 2014; http://agilescout.com/scaled-agile-framework-safe-review/.
28. N. Killick, “The Horror Of The Scaled Agile Framework | neilkillick.com,” 2012; http://neilkillick.com/2012/03/21/the-horror-of-the-scaled-agile-framework/.
29. A. Elssamadisy, “Has SAFe Cracked the Large Agile Adoption Nut?,” 05/08/2014 2014; http://www.infoq.com/news/2013/08/safe#.
30. K. Schwaber, “unSAFe at any speed,” *Book unSAFe at any speed*, Series unSAFe at any speed, ed., Editor ed.^eds., @BigScrum, 2014, pp.
31. R. Jeffires, “Issues with SAFe | xProgramming.com,” *Book Issues with SAFe | xProgramming.com*, Series Issues with SAFe | xProgramming.com, ed., Editor ed.^eds., 2014, pp.
32. D. Torgeir and B.M. Nils, “Research challenges in large-scale agile software development,” *SIGSOFT Softw. Eng. Notes*, vol. 38, no. 5, 2013, pp. 38-39; DOI 10.1145/2507288.2507322.