# Agile on a large scale

Bass, J

http://dx.doi.org/10.1093/itnow/bwz023

## Large Scale Agile

Large-scale agile methods are used where multiple teams cooperate over an extended period of time on a shared development programme. Large-scale development programmes often entail a complex mix of technologies and a wide range of interested stakeholders. Further, large-scale often goes hand-in-hand with geographical distribution. Cross-border development teams and international customer relationships are commonplace. These are typically high value projects that can impair careers and attract unwelcome public attention when things don't go well.

Conventionally, agile methods have been associated with small collocated teams. However, the attractive ability of agile methods to respond to changing priorities and mitigate risk stimulates adoption in large-scale and cross-border team settings. While agile teams are self-organising (Hoda, Noble & Marshall, 2011), cooperating agile teams have to sacrifice some level of autonomy to work with each other.

You can find frameworks around for large-scale agile such as Large Scale Scrum (Less) (Larman & Vodde, 2016) and Disciplined Agile Delivery (DAD) (Ambler and Lines, 2012). However, you find more mature adopters often tailor and evolve their own unique approach.

Agile method enthusiasts identify artefacts and roles as defining aspects of software development processes. You find that development artefacts and roles have to evolve when undertaking large-scale development programmes. When compared to conventional agile, an extended range of development artefacts are created by teams as part of large-scale software development programmes (Bass, 2016). You also find people with a wider variety of job titles in large-scale agile development programmes, than in conventional agile methods.

## Development Artefacts

### Risk Register

A risk register identifies potential sources of risk, estimates the likelihood of occurrence and describes mitigating actions for each risk. Agile methods advocates do not normally consider a risk register as a normal part of approaches such as Scrum. However, large organisations in this study were seen to monitor risks and plan mitigation strategies. You find product owners performing a risk assessment, across all the teams, during each sprint.

### Reference Architecture

A reference architecture is used to inculcate a consistent design approach among co-operating development teams working on the same development programme. Reference architectures promote prevailing architecture standards when they are disseminated to cooperating teams. There is evidence that reference architectures reduce development and software maintenance costs (Martínez-Fernández, et al., 2013). Further, reference architectures can facilitate induction of new staff as well as re-deployment across project teams in the development programme.

## Architecture Standards

You find that agile teams on large-scale development programmes have to comply with architecture standards that define how technologies support business strategy. Architecture standards help to ensure software systems are well structured, internally consistent, simple to understand, easy to maintain and satisfy non-functional requirements. Architecture standards help cooperating teams coordinate their use of technology.

## Release Plans

The “potentially shippable” code produced by self-organising teams at the end of successive sprints is collected into releases. Release plans enable cooperating teams to coordinate code development to minimise the impact of inter-dependencies. Release plans also enable phased delivery of functionality in ways that support progressive deployment, user testing and performance testing under load.

## Activities within Roles

### Product Sponsor

In large scale agile, you find the product owner role evolves into a product owner team (Bass, 2015). A product sponsor needs to surround themselves with a team to liaise with the many stakeholders in the development programme. For example, one product sponsor for a bank transformation project in this study was a board member responsible for over 5000 IT staff. That CIO needs to set the project vision and maintain focus on key goals.

### Intermediary

You find an intermediary interfacing with the product sponsor to disseminate information about the project goals and vision. Intermediaries need to have a sound understanding of the business domain and project objectives obtained through regular discussions with the product sponsor. Intermediaries set priorities in order to achieve project goals.

### Technical Architect

The technical architect activity is used to support the product owner team members that are conventionally more business-oriented. Architecture standards and the reference architecture need technical expertise to coordinate agile teams. The architect establishes and refines conventions for structuring large scale software systems by providing support to team members and helping to disseminate best practices.

### Traveller

You know that face-to-face communication is the gold-standard for building trust and consensus building. In global software development, time spent together is invaluable. However, entire teams

cannot be transported to collocated sites. Instead, you find a member of the product owner team needs to spending time with clients and stakeholders at key sites in the development effort.

### Communicator

Where face-to-face communication is not possible, then you find audio and video conferencing being used. Daily stand-up meetings, scrum-of-scrums and meetings to remove impediments are all conducted using communication technologies. With increasing use of trans-continental software development models, the communicator often has to fit in with the time zones of other stakeholders by attending events outside their normal office hours.

### Governor

You find product owners performing governor activities when they are ensuring compliance with quality and technology standards. Large projects often sit within a compliance framework to ensure consistency between teams and inter-operability of moving parts within a software project. One of the companies in this study manufactures medical equipment for use in hospitals worldwide. Regulators meticulously review many aspects of their development process. The governor works with teams to ensure appropriate attention is paid to such standards in the development process.

Agile methods in large-scale development programmes seek to capture the benefits of phased delivery, responsiveness to change and emphasis on collaboration while introducing new artefacts and activities to manage scale.

### Methods Sidebar

These findings are mainly derived from over 90 practitioner research interviews in 21 organisations conducted over an 8-year period. In addition, examples of public and commercially-sensitive project documentation have been reviewed and ceremonies observed, such as stand-up meetings. To recruit interview subjects snowball sampling was used alongside intensity sampling within participating organisations.

The practitioners interviewed have included development team members with job titles such as developer, software engineer and tester. In addition, corporate senior management have been interviewed with job titles such as CIO, CTO and head of engineering. Interviewees have also, of course, included agile coaches, scrum masters and product owners.

The organisations have included UK-based local- and national-Government bodies, as well as European, North American and Indian multi-national companies. Offshore software services companies, predominantly in India, have kindly participated in the study.

Research interviews have been audio-recorded and transcribed. Interview transcripts were analysed using a “Glaserian” grounded theory approach (Glaser, 1992). The approach involves open coding, and “memoing” topics found in the transcripts which through constant comparison evolve into categories with successive rounds of interviews until theoretical saturation has occurred (Glaser & Strauss, 1967).

References

Ambler, S. W., and Lines, M. (2012). Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. Upper Saddle River: IBM Press.

Bass, J. M. (2016). Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, 75, 1–16. https://doi.org/10.1016/j.infsof.2016.03.001

Bass, J. M. (2015). How product owner teams scale agile methods to large distributed enterprises. Empirical Software Engineering, 20(6), 1525–1557. https://doi.org/10.1007/s10664-014-9322-z

Glaser, B. G. (1992). Basics of Grounded Theory Analysis: Emergence vs. Forcing. Mill Valley: Sociology Press.

Glaser, B. G., & Strauss, A. L. (1967). Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine.

Hoda, R., Noble, J., & Marshall, S. (2011). Developing a grounded theory to explain the practices of self-organizing Agile teams. Empirical Software Engineering, 17(6), 609–639. https://doi.org/10.1007/s10664-011-9161-0

Larman, C., & Vodde, B. (2016). Large-Scale Scrum: More with LeSS (1 edition). Boston: Addison-Wesley Professional.

Martínez-Fernández, S., Ayala, C. P., Franch, X., & Martins Marques, H. (2013). Benefits and Drawbacks of Reference Architectures. In K. Drira (Ed.), Software Architecture (pp. 307–310). Springer Berlin Heidelberg.