Artefacts and agile method tailoring in large-scale offshore software development programmes

Julian M. Bass*

University of Salford, The Crescent, Salford, Manchester, UK

A B S T R A C T

Context: Large-scale offshore software development programmes are complex, with challenging deadlines and a high risk of failure. Agile methods are being adopted, despite the challenges of coordinating multiple development teams. Agile processes are tailored to support team coordination. Artefacts are tangible products of the software development process, intended to ensure consistency in the approach of teams on the same development programme.

Objective: This study aims to increase understanding of how development processes are tailored to meet the needs of large-scale offshore software development programmes, by focusing on artefact inventories used in the development process.

Method: A grounded theory approach using 46 practitioner interviews, supplemented with documentary sources and observations, in nine international companies was adopted. The grounded theory concepts of open coding, memoing, constant comparison and saturation were used in data analysis.

Results: The study has identified 25 artefacts, organised into five categories: feature, sprint, release, product and corporate governance. It was discovered that conventional agile artefacts are enriched with artefacts associated with plan-based methods in order to provide governance. The empirical evidence collected in the study has been used to identify a primary owner of each artefact and map each artefact to specific activities within each of the agile roles.

Conclusion: The development programmes in this study create agile and plan-based artefacts to improve compliance with enterprise quality standards and technology strategies, whilst also mitigating risk of failure. Management of these additional artefacts is currently improvised because agile development processes lack corresponding ceremonies.

1. Introduction

Practitioners managing large-scale offshore software development programmes appear to find it increasingly attractive to blend elements of both plan-based and agile software development methods, with the result being a pragmatic tailoring of agile methods which accommodates organisational constraints, governance requirements and geographical distribution. This article explores process tailoring by investigating artefact inventories, using empirical data collected from industry practitioners at all levels representing nine international companies.

Agile methods have been proposed as a way to avoid project failures [1]. Risk of project failure is reduced each time a software increment is delivered, since the highest priority requirements are selected for development during each increment and each increment is used to gather client and user feedback. Increments are delivered regularly and each comprises a carefully defined fragment of the overall development effort. This contrasts with plan-based methods in which risks progressively rise until product handover at the end of the project. There is evidence that agile methods improve both software development productivity and product quality [2]. However, such agile methods are conventionally associated with small, co-located development teams.

The scaling of agile methods to large-scale software development programmes has attracted interest from practitioners [3–5], and has been identified as a priority area for researchers [6,7]. For example, the scrum of scrums approach supports multiple concurrent scrum teams [3]. Teams working in parallel with each

* Tel.: +441224262732.
E-mail address: j.bass@salford.ac.uk
URL: https://www.seek.salford.ac.uk/profiles/JBass.jsp

other need coordination, consequently the scrum masters from each team work together to coordinate activities, manage dependencies and avoid the duplication of effort [8,9].

Outsourcing is the process of procuring products or services from a third-party vendor or provider [10]. In onshore outsourcing the third-party vendor is located in the same territory as the client organisation. Onshore outsourcing is not the focus of this study. Offshore outsourcing involves a geographically remote third-party vendor, often separated from the client organisation by significant temporal and cultural distance. In contrast, some client organisations establish their own in-house offshore development centres. Offshore development (whether in-house or outsourced) can help establish a presence in emerging markets or benefit from their anticipated lower cost base.

The focus of this study is on large-scale development programmes comprising a system under development or integrated portfolio of related products. Inevitably, large-scale systems involve the integration of new features into an existing code base sometimes called a legacy system. In this study, large-scale consists of at least 25 developers configured into multiple cooperating teams working together for a period of 9 months or more.

Various forms of artefact are used to negotiate, record and disseminate decisions made during the development process. An artefact is a tangible product or by-product produced during the development of software, typically including: models, designs, reports and source code. The artefacts act as boundary objects between the different technical specialisms of stakeholders involved in the development programme [11].

Development teams are required to record and share design decisions to avoid duplication of effort and resolve dependencies. The artefacts record the results of negotiations between stakeholder groups, decisions made and decisions revised. Written records contradict the agile manifesto which advocates focus on working software over comprehensive documentation [12]. So, large-scale agile development programmes dictate forms of documentation to coordinate the activities of groups and teams, yet agile methods advocate focus on working code. Thus, artefacts represent an area of tension between traditional plan-based methods and agile methods. As a consequence, artefacts can provide insights into the tailoring of agile methods in large-scale development programmes.

To enhance understanding of software development process tailoring in large-scale offshore agile software development programmes, this research explores practitioner interactions with the artefacts used. The main research question for this study is: "how do practitioners describe the inventory of artefacts they use in large-scale offshore software development programmes?"

This primary research question is further explored using two subsidiary research questions: "how do the artefacts map to software development processes used in large-scale offshore software development programmes?" and "how do these practitioner descriptions contribute to our understanding of artefacts in agile method tailoring in large-scale offshore software development programmes?"

In order to answer these research questions the author has conducted qualitative empirical research with nine international companies engaged in large-scale offshore agile software development programmes; leading to 46 open-ended semi-structured interviews with practitioners ranging from senior executives to novice testers and developers. In addition, documentary sources describing development process standards and guidelines have been reviewed and workplace observations have been conducted.

The main contribution of this article is a systematic description of practitioner interactions with the artefacts created in large-scale offshore agile software development programmes. The project teams in this study use several agile techniques, notably: daily stand-up meetings, short iterations, prioritized backlogs, it-

eration planning, retrospectives and release planning. These techniques have been identified as most popular by respondents to a well established industry survey [13].

Five categories of artefacts emerge from the empirical data collected in this study: feature, sprint, release, product, and programme governance. A taxonomy is then established that relates these artefacts to their role in the software development process. An actor within the development process is identified as primary owner of each artefact. Further, information sources and information consumers for each artefact are derived from the data obtained during this study. It is suggested that agile processes are missing ceremonies for managing certain artefacts and that agile processes need to be enhanced with additional scrum of scrum ceremonies to manage these artefacts in large-scale offshore software development programmes.

The rest of this paper is structured as follows. In the next section a review of related work is undertaken, with agile methods summarised, along with a brief review of global software development, where the use of agile methods in large-scale offshore software development programmes is considered. The article then introduces the research method adopted, providing information on the selected research sites, data collection methods and analysis undertaken. Findings are then presented, organised into sections on programme governance, product, release, sprint, and feature artefacts. At the end of the article, the findings are discussed, and the limitations of the work are presented, along with suggestions for further work and conclusions.

2. Agile software development

There are a range of agile software development methods that are increasingly being adopted in large-scale offshore software development programmes, including Feature Driven Development, Scrum, Extreme Programming (XP) and Lean Software Development [14]. These software development methods build upon three key themes in software engineering: development using short iterations, feature-driven development and the close interaction with customers.

Short iterations are now widely used in software development, providing frequent and regular feedback on smaller scale development activities, rather than more traditional six or nine month development efforts. Within an agile team, short iterations help to identify the causes of any development delays, while the relevant development team also gains feedback from external stakeholders regarding its compliance with agreed requirements. Short iterations require the continuous integration of software code artefacts as they are combined, tested and released [15]. The use of automated software tools migrates code artefacts between the platforms used for build, integration and testing activities.

In feature-driven development, team members work together in self-organising teams to produce end-to-end functionality [16]. Each feature is designed to meet a business need and is just one element of a much larger system. Such features must include all the necessary technical components (including databases, network communications and user interface screens) needed to solve that specific business problem, and thus provide end-to-end functionality. Features are self-contained and independent, making them well suited for managing from the initial requirements stage through design and implementation to final testing. Staff members in feature teams either possess all the necessary skills needed to build all the technical components to implement a particular feature; or occasionally they work together in small groups to bring the required skills together. The feature team concept is in contrast to teams organised around technical specialisms, where, for example, one team has the skills to develop user interfaces with another team responsible for developing databases.