ScienceDirect

Information and Software Technology

Volume 75, July 2016, Pages 1-16

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

Julian M. Bass

Show more

Share Cite

https://doi.org/10.1016/j.infsof.2016.03.001

Get rights and content

Abstract

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, release, product and corporate governance. It was discovered that conventional

Cite

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.

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], [4], [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 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, iteration 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.

Section snippets

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 …

Method

This research adopts a grounded theory approach, using the empirical data gathered from research sites in selected international companies, as shown in Table 1. Data collection used documentary evidence, workplace observation of software development practices and semi-structured face-to-face practitioner research interviews, which were recorded and subsequently transcribed. The unit of analysis is the set of software practitioners working on development programmes in selected organisations. …

Findings

The artefacts identified in this study are organised into five levels of abstraction: programme governance, product, release, sprint and feature. Each of these categories is discussed in turn. …

Discussion

The application of agile methods in large-scale and geographically distributed software development programmes is an area of emerging interest [7]. Whereas agile early adopters tended to use engineering practices from the XP process [1], this research confirms more recent findings showing an increase in the popularity of scrum process orchestration practices [13], [49].

Previous research has focused on specific artefacts, such as the story card and wall [40], or on artefacts developed during a …

Conclusions

This article uses a grounded theory approach to investigate artefact inventories used in large-scale offshore software development programmes, using qualitative analysis of

practitioner interview transcripts and supported by documentary sources and workplace observations. The study has focused on a sample of nine international companies and interviews with 45 practitioners. The grounded theory concepts of open coding, memoing, constant comparison and saturation were used to analyse the data …

Acknowledgements

I am grateful to the companies and interviewees who participated in this research. Thanks also go 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 during several research visits. The research benefited in part from travel funding from the UK Deputy High Commission, Bangalore; Science and Innovation Network; and Robert Gordon …

Recommended articles

References (60)

T. Dybå *et al.*
Empirical studies of agile software development: A systematic review
Inf. Softw. Technol. (2008)

J. Vlietland *et al.*
Towards a governance framework for chains of scrum teams
Inf. Softw. Technol. (2015)

J. Vlietland *et al.*
Aligning codependent scrum teams to enable fast business value delivery: A governance framework and set of intervention actions
J. Syst. Softw. (2016)

T. Dingsøyr *et al.*
A decade of agile methodologies: Towards explaining agile software development
J. Syst. Softw. (2012)

B. Fitzgerald *et al.*
Customising agile methods to software practices at intel shannon
Eur. J. Inf. Syst. (2006)

K.M. Lui *et al.*
The effect of pairs in program design tasks
IEEE Trans. Softw. Eng. (2008)

S. Adolph *et al.*
Using grounded theory to study the experience of software development
Empir. Softw. Eng. (2011)

M.Q. Patton
Qualitative Research & Evaluation Methods
(2002)

Y.S. Lincoln *et al.*
Naturalistic Inquiry
(1985)

T. Dybå *et al.*
What do we know about agile software development?
IEEE Softw. (2009)

View more references

Cited by (100)

Revealing the state of the art of large-scale agile development research: A systematic mapping study

2022, Journal of Systems and Software

> *Citation Excerpt :*
> ...The Global and distributed software engineering research stream deals with how companies can use agile methods in large, globally distributed projects. Studies mainly focus on two sub-topics, namely identifying challenges and success factors in adopting agile methods in large, globally distributed projects (cf. Shameem et al., 2018c, 2017, Paasivaara et al., 2008b) and the tailoring of agile methods to meet the needs of these projects (cf. Bass, 2016a,b, 2014). As an example for the first sub-topic, Shameem et al. (2017) describe critical factors that positively impact the adoption of agile methods in large, globally distributed projects based on a systematic literature review....

Show abstract

Why and how is Scrum being adapted in practice: A systematic review

2022, Journal of Systems and Software

Show abstract