On Infrastructure for Facilitation of Inner Source in Small Development Teams

Johan Linåker, Maria Krantz, and Martin Höst

Software Engineering Research Group, Computer Science, Lund University, Lund, Sweden
{johan.linaker,martin.host}@cs.lth.se

Abstract. The phenomenon of adopting open source software development practices in a corporate environment is known by many names, one being inner source. The objective of this study is to investigate how an organization consisting of small development teams can benefit from adopting inner source and assess the level of applicability. The research has been conducted as a case study at a software development company. Data collection was carried out through interviews and a series of focus group meetings, and then analyzed by mapping it to an available framework. The analysis shows that the organization possesses potential, and also identified a number of challenges and benefits of special importance to the case company. To address these challenges, the case study synthesized the organizational and infrastructural needs of the organization in a requirements specification describing a technical infrastructure, also known as a software forge, with an adapted organizational context and work process.

Keywords: Inner source, Open source software, Software development practices, Software ecosystem, Life cycle, Programming teams, Software process models, Software reuse, Software forge

1 Introduction

Many open source software products have been successful in recent years, which have led to an increased interest from the industry to investigate how the development practices could be introduced in a corporate environment and take advantage of the benefits seen in open source projects. Such practices include e.g. universal access to project artefacts [8], early and frequent releases, and “community” peer-review [2].

Mistrik et al. [11] address how closed development organizations could benefit from open source practices as an area where further research is needed. Though studies conducted so far are quite limited, several success stories [1,2,8,13,20] can be found of large corporations adopting open source development.

The phenomenon of adopting these development practices in a corporate environment has in research been referred to by many names, e.g. inner source [17], corporate open source [2,3] and progressive open source [1]. In this report we have chosen to use the term inner source, as described by Stol [17].
The changes required when adopting inner source in a corporate environment led Gurbani et al. [3] to suggest two different methods to effectively manage inner source assets; an infrastructure-based model and a project-based model.

In the infrastructure-based model, the corporation provides the critical infrastructure that allows interested developers to host individual software projects on the infrastructure, much like SourceForge\(^1\) or Github\(^2\) does with open source projects. Platforms like these, also known as software forges [13], can be resembled as component libraries where each project represents a component of different abstractions, e.g. modules, frameworks or executables. Developers can browse between the components and use or contribute to those they wish. The reuse of software can be considered opportunistic or ad hoc and there is no limitation on the number of projects to be shared within the organization. Success stories include cases from SAP [13], IBM [16,19], HP [1,10] and Nokia [7,8].

In the project-based approach the software is managed in a project, instead of as a long-term infrastructure. Gurbani et al. [3] describe how an advanced technology group, or a research group funded by other business divisions in a corporation takes over a critical resource and makes it available across the organization. This team is often referred to as the “core team” and is responsible for the project and the decision making. Philips Healthcare [20] and Alcatel-Lucent [3] are two documented cases where this variant has been adopted.

In order to assess the applicability of inner source on an organization, Stol [17] developed a framework. This framework is based on reviewed literature and a case study of a software company referred to as “newCorp”. Though the framework focuses on project-based models, it is based on success factors and guidelines described in both project-based and infrastructure-based case studies. The framework consists of 17 elements divided into four categories; Software product, Development practices, Tools and infrastructure, and Organization and community. These categories were inspired and mapped to those proposed by Gurbani et al. [2]. The elements can be found in the left column of Table 1. In a later publication Stol et al. [18] present an updated version of the framework where it has been restructured, now consisting of nine elements divided into three groups; Software product, Practices and Tools, and Organization and Community. The elements are more generalized and covers all of those described in the initial framework [17].

Adopting inner source requires significant effort and change management, which is one reason why it may be of interest to start on a smaller scale before investing globally. However, this requires an understanding of how inner source can be implemented on smaller teams and what parts that can be implemented and evaluated. This study is focused towards the latter and aims to contribute theoretically by using a framework by Stol [17] to assess the applicability of inner source, and based on the identified challenges synthesize a solution that addresses the organizations needs.

\(^1\) http://www.sourceforge.com/
\(^2\) http://github.com