Key Factors for Adopting Inner Source

KLAAS-JAN STOL, Lero, University of Limerick
PARIS AVGERIOU, University of Groningen
MUHAMMAD ALI BABAR, University of Adelaide
YAN LUCAS, Neopost Technologies
BRIAN FITZGERALD, Lero, University of Limerick

A number of organizations have adopted Open Source Software (OSS) development practices to support or augment their software development processes, a phenomenon frequently referred to as Inner Source. However, the adoption of Inner Source is not a straightforward issue. Many organizations are struggling with the question of whether Inner Source is an appropriate approach to software development for them in the first place. This article presents a framework derived from the literature on Inner Source, which identifies nine important factors that need to be considered when implementing Inner Source. The framework can be used as a probing instrument to assess an organization on these nine factors so as to gain an understanding of whether or not Inner Source is suitable. We applied the framework in three case studies at Philips Healthcare, Neopost Technologies, and Rolls-Royce, which are all large organizations that have either adopted Inner Source or were planning to do so. Based on the results presented in this article, we outline directions for future research.

Categories and Subject Descriptors: D.2.9 [Software Engineering]: Management—Software process models; K.6.3 [Management of Computing and Information Systems]: Software Management—Software development, Software maintenance, Software process

General Terms: Human Factors, Management, Theory

Additional Key Words and Phrases: Case study, inner source, open-source development practices, framework

ACM Reference Format:
DOI: http://dx.doi.org/10.1145/2533685

1. INTRODUCTION

Numerous software development methods have been proposed to guide organizations in their software development: from the traditional waterfall approach and the V-model to the more recent and highly popular agile methods. While these methods all aim at providing a systematic way to develop software, the emergence of successful open source projects has been remarkable, given the seeming absence of a predefined process—what Erdogmus [2009] provocingly called an antiprocess. Open-source communities have produced a number of high-quality and highly successful products, including the...