Chapter 8.7
Bridging the Gap between Agile and Free Software Approaches: The Impact of Sprinting

Paul J. Adams
Sirius Corporation Ltd., UK

Andrea Capiluppi
University of Lincoln, UK

ABSTRACT

Agile sprints are short events where a small team collocates in order to work on particular aspects of the overall project for a short period of time. Sprinting is a process that has been observed also in Free Software projects: these two paradigms, sharing common principles and values have shown several commonalities of practice. This article evaluates the impact of sprinting on a Free Software project through the analysis of code repository logs: sprints from two Free Software projects (Plone and KDE PIM) are assessed and two hypotheses are formulated: do sprints increase productivity? Are Free Software projects more productive after sprints compared with before? The primary contribution of this article is to show how sprinting creates a large increase in productivity both during the event, and immediately after the event itself: this argues for more in-depth studies focussing on the nature of sprinting.

INTRODUCTION

Agile and Free Software development have received rapid growth in popularity, both as development paradigms and as research topics. In theory they are very different concepts; the latter, strictly speaking, being just a licensing paradigm with implications for code reuse and redistribution.

The interface between Agile and Free Software is very interesting and a fertile area in which not much rigorous research has been carried out to date. Some comparative studies have been made in the past, but given the scarcity of data from

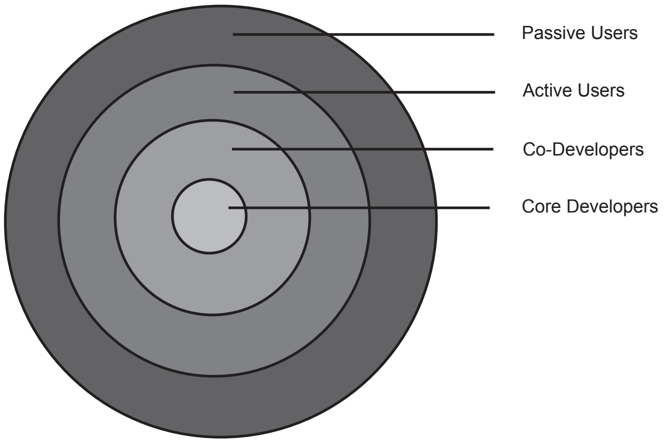
Agile processes, most of the studies have remained on the surface of theoretical discussions (Koch, 2004)(Warsta and Abrahamsson, 2003). Empirical attempts have been also made to measure, on an empirical basis, the degree of *agility* within other development paradigms (Adams, Capiluppi and deGroot, 2008).

This article examines and compares the Free Software and Agile approaches by observing typical Agile practices when deployed within Free Software teams: in particular, it reports on the Plone and KDE PIM projects, where sprinting (Beck, 1999) is commonly used by developers to focus the activity for a limited period of time. Sprinting allows developers to meet in person, get to know each other and create the basis for collaborating more effectively in a distributed environment (During, 2006). In previous works, the PyPy and the Zope projects have been reported to use sprinting regularly within their projects (Sigfridsson, Avram, Sheehan and Sullivan, 2007), and its use is advocated as an "*applied (...) idea of Agile development to the very difficult problem of distributed software development"* (Goth, 2007).

What past literature has not provided yet is a quantitative evaluation of the impact of sprinting on productivity of developers: what has instead been reported is that traditional productivity metrics could fail in capturing the effects of the interactions among developers within sprints (Goth, 2007). In order to tackle this issue, this article explores the use of automatic measures to determine the productivity of developers both before and after the sprinting efforts. A research hypothesis has been formulated as follows: when quantitatively evaluating sprinting, the productivity of developers will display higher values after a sprint than before it. If the null hypothesis can be rejected, this result could prove useful to others in the Free Software communities, encouraging them to adopt this practice and to focus their efforts within a constrained period of time to increase their productivity.

This article is structured as follows: Section 2 introduces the context of the work, explaining how the Agile and Free Software paradigms share some of their process characteristics. Section 3 reports on the methodology, the attributes and the definitions used throughout the article. Section 4 describes how sprinting is accomplished within the two reported case studies, while Section 5 summarises the main findings of measuring the effects of sprinting on developers productivity. Since this work reports on empirical analysis of public data, Section 6 will report on the threats to validity. Finally Section 7 will conclude the article, and illustrate avenues of further research.

Figure 1. The open development model


Passive Users
Active Users
Co-Developers
Core Developers