Chapter 3
Agile Software Engineering

Ernest Mnkandla
University of Johannesburg, South Africa

ABSTRACT
This chapter aims to reveal agile techniques that have been applied to software development and have resulted in meaningful improvements in software productivity. Available literature generally state some claims on the gains associated with the use of particular agile methodologies in software development. What lacks however, is a comprehensive analysis of how the application of agile techniques as a family will lead to improvement in software productivity. This chapter therefore provides such details. Software productivity techniques provide ways of measuring three things in order to determine the productivity of software; software products, software production processes and structures, and software production setting. Agile methodologies improve software productivity by focusing on the software production process and structures. The fundamental concern of this chapter is to show that agile methodologies measure the production process activities in a different but effective way from the more traditional approaches. For example, time-to-market is reduced by use of an iterative incremental development approach.

INTRODUCTION
It may seem a little mischievous to use the term ‘agile software engineering’ in this context or any other context for that matter, but it has been used in this chapter simply to emphasize the values that an agile perspective to software development and software engineering has added to the entire discipline of software engineering. Agile software engineering can be defined as the streamlined production of software systems with a focus on a quality end product, relevant customer requirements, and following a minimalist approach to the process density.

There is still a high level of skepticism about the benefits of agile software development methodologies especially in organizations or individuals that have not applied these methods or seen

DOI: 10.4018/978-1-60566-731-7.ch003
them in use. However, it should be made clear that there is published evidence of the benefits of developing software the agile way in sources such as (Abrahamsson et al, 2002; Beck, 1999; Beck and Andres, 2004; Boehm and Turner, 2004; Cockburn, 2000; Lindvall et al, 2002; Collins-Cope, 2002; Fowler, 2000; Highsmith, (2001); Highsmith, 2002b; Highsmith, 2004; Turk et al, 2002). Nevertheless it is important that we continue to publish empirical evidence of the successful application of agile methodologies. In general there is a constant growth in scholarly publications such as books, journals, conference proceedings, white papers and other experience reports that continue to validate the relevance of agile development.

This chapter provides a detailed analysis of how software productivity is achieved through the use of different agile processes.

Software productivity like other agile development concepts and practices such as software quality assurance, iterative incremental development, inclusion of the customer in the development team, pair programming (from Extreme Programming -XP) et cetera is clouded with controversy as to how it is measured in agile processes and whether the metrics used in such efforts prove agile to be productive. In business, people always talk about the bottom line i.e. ROI, it is almost synonymous to what is stated in the ‘good old book’, after ‘the Wiseman’ has advised on almost all issues of life and humanity he gives ‘the conclusion of the whole matter’ which in essence summarises the purpose of all humanity. The conclusion of the whole matter in software development is that ‘our investment should yield working code and a return on our investment’ else we labour in vain.

Agile development is about doing software development rather than talking about it, and those who have applied agile methods in their software development projects have gained margins that are hard to ignore in the areas of product relevance (a result of embracing requirements instability) and quick delivery (a result of iterative incremental development) (Mnkandla, and Dwolatzky, 2007). There is also evidence of unequalled financial savings on project costs (Highsmith, 2002b). Those who have not yet joined this new fun way to develop software probably due to a lack of understanding of the fundamental concepts underlying agile methodologies, or mere unwillingness to face the winds of change have been left out of the frenzy (Mnkandla and Dwolatzky, 2004a). Hence this chapter intends to give the necessary understanding by defining agile methodologies and revealing how the use of agile methodologies leads to improvement in software productivity. The metrics and techniques used to measure software productivity in agile development will also be reviewed.

**Chapter Objectives**

The objective of this chapter is to introduce you to the fundamentals of measuring software productivity in agile software development methodologies in order to unravel the real benefits of agile development. After reading this chapter you will have better understanding of:

- What agile development means in terms of software productivity
- The major difference between agile development and the other approaches
- The agile techniques that will improve software productivity
- The empirical evidence of improved software productivity in agile development

**BACKGROUND**

This section will start by defining agile methodologies based on the existing definitions and practices on agile methodologies. Software productivity will also be briefly defined. The theory of agile development will also be explained in order to reveal the underlying philosophy that is in fact based on improving software productivity.