

## Chapter 12

# Introducing Agility into Plan-Based Assessments

**Minna Pikkarainen**

*University of Limerick, Ireland & VTT Technical Research Centre of Finland, Finland*

**Fergal McCaffery**

*Dundalk Institute of Technology, Ireland*

### ABSTRACT

*Agile or Plan-driven approaches to software process improvement (such as the Capability Maturity Model Integration (CMMI) and the ISO/IEC 15504 standard) claim to provide companies with improved processes, higher quality software and faster software development. Assessment is an important component of any software process improvement programme, as in order for an organisation to commence an improvement programme, they must first be aware of the current state of their software development practices. Therefore, in the case of small companies such assessments need also to be cost effective and focused only on the most relevant process areas.*

### INTRODUCTION

In this chapter we will address the question of how to integrate agile practices into traditional software process assessments. We will begin by presenting a background section that will detail traditional

software process assessment and agile software development. We then will illustrate research we performed to develop and implement a low resource hybrid approach (AHAA) (McCaffery et al. 2008) for software process assessment and improvement that integrates CMMI (CMMI, 2006), Automotive SPICE™ (Automotive S, 2007) and agile practices together. We will also discuss the

DOI: 10.4018/978-1-60960-215-4.ch012

applicability of combining agile and plan-driven software development approaches for different types of domains e.g. financial, safety-critical. Next, we provide an empirical examination, based on assessments that were performed in two Small to Medium Sized Enterprises (SMEs) using the AHAA method. Finally, we discuss the empirical findings from both assessments, the evolution of the AHAA method and provide conclusions and recommendations for researchers and practitioners wishing to combine traditional and agile software development practices.

## **BACKGROUND**

This section describes SPI, software process assessment and agile development.

### **Software Process Improvement**

Continuous SPI can assist companies to satisfy customers through providing high quality deliverables in an efficient and repeatable manner. This may be particularly beneficial to SMEs that have to satisfy increasingly demanding customers using a limited pool of resources. In fact, SMEs often lack maturity in their software development processes. In many cases, SMEs have chaotic models of operation that impact the success of the entire organisation (Batista & Figueiredo, 2000). However, just as the standards world has recognised that software engineering standards should not only apply to large high maturity organisations but also to low capability level organisations and this has led to the development of an International standard for Very Small Enterprises (Laporte et al. 2008), there also is a need for SPI models and assessment methods for such organisations. A software process assessment may be used to determine weaknesses in an organisation's software development processes and consequently be used to initiate SPI work within an organisation (Humphrey, Snyder et al. 1991). In SMEs, such assessments need to

be cost effective (Batista & Figueiredo, 2000) and focused on specific and important process areas (Richardson, 2001). Agile approaches constituting a set of principles, methods and practices have become popular within software companies (Hansson et al. 2006). The reasons for this adoption are obvious in that companies need to be agile in order to survive in dynamic business environments (Kettunen, 2009). An agile approach can also provide a systematic mechanism to manage projects (Sutherland, Viktorov et al. 2007). Although it has been argued that: "Both agile and planned approaches have situation-dependent shortcomings that, if left unaddressed, can lead to project failures" (Boehm & Turner, 2003) and that companies should integrate best practices from both agile and traditional software development, there has been very little input from the research community or practitioners on how to increase agility through performing software process assessments.

Agile methods promise companies improved software productivity and quality (Holström, Fitzgerald et al. 2006). Such improvements have previously been achieved through adopting traditional SPI models and assessment methods (Galin & Avrahami, 2006; Niazi et al., 2006). It has been shown, for instance, that CMMI based SPI programmes have resulted in companies obtaining between 28–53% improvements in lead time and between 70 to 74% improvement in terms of quality (measured by the amount of defects) (Galin & Avrahami, 2006). To obtain benefits from traditional and agile methods, both approaches must however be deployed. Yet, agile researchers have known for some time that there are fundamental differences between traditional (e.g. CMMI) and agile methods that may cause difficulties when integrating these approaches. For example, whilst agile methods emphasise face to face discussions and reduced documentation, CMMI improvements can lead to a situation in which the developers have implemented more documentation than software code (Boehm & Turner, 2003).

32 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/introducing-agility-into-plan-based/51977](http://www.igi-global.com/chapter/introducing-agility-into-plan-based/51977)

## Related Content

---

### Aligning Strategic-Driven Governance of Business IT Services With Their Agile Development: A Conceptual Modeling-Based Approach

Konstantinos Tsilionis and Yves Wautelet (2021). *Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products* (pp. 228-247).

[www.irma-international.org/chapter/aligning-strategic-driven-governance-of-business-it-services-with-their-agile-development/259180](http://www.irma-international.org/chapter/aligning-strategic-driven-governance-of-business-it-services-with-their-agile-development/259180)

### Improving the Bluetooth Hopping Sequence for Better Security in IoT Devices

Matt Sinda, Tyler Danner, Sean O'Neill, Abeer Alqurashi and Haeng-Kon Kim (2018). *International Journal of Software Innovation* (pp. 117-131).

[www.irma-international.org/article/improving-the-bluetooth-hopping-sequence-for-better-security-in-iot-devices/210459](http://www.irma-international.org/article/improving-the-bluetooth-hopping-sequence-for-better-security-in-iot-devices/210459)

### A Comparative Analysis of Software Engineering Approaches for Sequence Analysis

Muneer Ahmad, Low Tang Jung and Noor Zaman (2013). *Software Development Techniques for Constructive Information Systems Design* (pp. 285-295).

[www.irma-international.org/chapter/comparative-analysis-software-engineering-approaches/75751](http://www.irma-international.org/chapter/comparative-analysis-software-engineering-approaches/75751)

### Towards Risk Based Effort Estimation: A Framework to Identify, Analyze, and Classify Risk for Early Identification at Requirement Engineering Phase

Cerene Mariam Abraham, M. Sudheep Elayidom and T. Santhanakrishnan (2018). *International Journal of Information System Modeling and Design* (pp. 67-84).

[www.irma-international.org/article/towards-risk-based-effort-estimation/220458](http://www.irma-international.org/article/towards-risk-based-effort-estimation/220458)

### Issues, Limitations, and Opportunities in Cross-Cultural Research on Collaborative Software in Information Systems

Dongsong Zhang and Paul Benjamin Lowry (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 2194-2229).

[www.irma-international.org/chapter/issues-limitations-opportunities-cross-cultural/29502](http://www.irma-international.org/chapter/issues-limitations-opportunities-cross-cultural/29502)