Agile Methods' Contributions in Software Evolution

Ned Chapin
InfoSci Inc., Box 7117
Menlo Park CA 94026, USA
NedChapin@acm.org

Abstract

The agile methods analyzed and reported on in this paper are those consistent with the Manifesto for Agile Software Development. *For these collectively, this paper examines the methods' contributions in software maintenance and evolution, to and by three groups of software system stakeholders: users, customers, and information systems personnel. For each, a picture emerges that has not been previously reported in the literature. That picture is a mix of favorable and unfavorable contributions, and weights that are situation dependent.*

1. Introduction

This one-page poster paper highlights some points from an analysis done on how agile methods are in practice actually used in software work. The agile methods looked at were those consistent with the *Manifesto for Agile Software Development* and the associated *Principles* document [1]. Two examples are eXtreme programming and Scrum.

Three groups of stakeholders were considered. One group is the software customer, the decision maker with authority to pay for software. An example is an airline's chief financial officer. One group is the software users, the persons or organizational units who directly either use the output of the software, or provide the input to the software, or both. An example is the tellers in a bank office. One group is information systems personnel who develop or maintain software. This group may be centralized or distributed within the organization that has users working with systems implemented with software, or external to it.

Software development creates new software, whereas software maintenance deals with existing software [2]. Software evolution is manifest in some kinds of changes over time in software [2]. While agile methods have been promoted for software development [1], their applicability in software maintenance has been noted [3, p. 135].

The analysis results reported in this poster paper were derived from examining both the contributions *to* the stakeholders from the use of agile methods in practice, and the contributions *by* the stakeholders to the use of agile methods in practice.

2. Executive summary

The relative importance of the summary points listed below depends upon the weightings reflecting the actual use situation of the agile methods. Four findings from the analysis were:

- In use, the great majority of agile methods accomplishes software maintenance that qualifies as software evolution. At most, only the first of the many deliveries is software development. All the rest are software maintenance.
- Information systems personnel using agile methods find the work usually more stimulating and personally rewarding. The on-the-job emphasis is on testing, intra-team interaction, and ongoing process improvement, with lots of variety and frequent closure in the work, and very little documentation.
- If the agile team be permanently assigned, the users gain both continuous evolution in the software they use, and a potentially strong voice in the direction of that evolution. Evolution that is too frequent can be chaotic for users, but more frequently evolved software can often better fit rapidly changing user needs. This makes weightings important.
- The customers' situations are moot, with weightings making the difference. The defacto contract with the agile team is a time-and-materials contract in practice, even when not explicitly written that way. Specifically defined functional completion criteria are usually absent and if present, are usually not enforced. Payment terms are usually clearly present and routinely met. The main, and usually only, agile deliverable is intangible user satisfaction. Typically the customer effectively gets nothing substantial for the organization and its software users.

3. References

[1] Agile Alliance, *Manifesto for Agile Software Development*, 2001. Http://www.agilemanifesto.org/

[2] N. Chapin, *et al*., "Types of software evolution and software maintenance," *Journal of Software Maintenance and Evolution*, John Wiley & Sons, Chichester UK, January 2001, pp. 3–30.

[3] K. Beck, *Extreme Programming Explained*, Addison-Wesley, Boston MA, 2000.

Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04)
1063-6773/04 $20.00 © 2004 **IEEE**