The International Journal of Management Education 11 (2013) 119–131

Contents lists available at ScienceDirect

The International Journal of Management Education

journal homepage: www.elsevier.com/locate/ijme

An agile method for teaching agile in business schools

Marija Cubric

Hertfordshire Business School, University of Hertfordshire, UK

ARTICLE INFO

Article history:
Received 12 October 2012
Received in revised form 6 September 2013
Accepted 11 October 2013

Keywords:
Agile
CSCL
Experiential learning
Project management
Scrum
Wikis

ABSTRACT

The aim of this paper is to describe, evaluate and discuss a new method for teaching agile project management and similar subjects in higher education.
Agile is not only a subject domain in this work, the teaching method itself is based on Scrum, a popular agile methodology mostly used in software development projects. The method is supported by wikis, a natural platform for simulation of software development environments.
The findings from the evaluation indicate that the method enables the creation of "significant learning", which prepares students for life-long learning and increases their employability. However, the knowledge gains, resulting from wiki interactions are found to be more quantitative than qualitative.
The results also imply that despite the active promotion of agile values of communication and feedback, issues regarding the teamwork are still emerging. The engagement of the teacher in the learning and teaching process was discovered to be a motivational factor for the team cohesion.
This paper could be of interest to anyone planning to teach agile in the higher education settings, but also to a wider academic community interested in applying agile methods in their own teaching practice.

1. Introduction

Agile is an "umbrella" term used for a set of closely related software development and project management methodologies, such as Scrum, eXtreme Programming (XP), Kanban, Dynamic Systems Development Method (DSDM) and others.

Agile methodologies are based on the values and principles from the "Agile manifesto" (2001), a statement published by a group of leading software practitioners in response to the growing concerns about the failure rate of IT projects (Standish Group, 1994). The failures were mainly attributed to the traditional, process- and documentation-heavy methodologies. The purpose of the Manifesto was to establish an alternative platform for delivery of successful projects, where the emphases is on: "individuals and interactions over processes and tools, working products over comprehensive documentation, customer collaboration over contract negotiation and responding to change over following a plan" (Agile Manifesto, 2001).

The most popular agile methodology is Scrum (Schwaber & Beedle, 2002), a lightweight project management framework based on the empirical process control model (Ogunnaike & Ray, 1994) and characterised by frequent iterative and incremental inspection and adaptation. In Scrum, all software features are grouped initially in a "product backlog". Each development iteration (called "sprint") starts with the planning meeting where the "product owner" (PO) decides which of the features will be included in the next sprint. All sprints are of equal duration, usually 2 weeks, but they could be shorter or

E-mail address: m.cubric@herts.ac.uk.

1472-8117/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ijme.2013.10.001

longer depending on the project. Teams are "self-organising" and during a sprint, the team members work on development of the selected features and coordinate their work through "daily stand-up" meetings. Each sprint ends with a review meeting, where the PO gives feedback to the team on the features developed during the sprint. The "Scrum master" is a member of the team responsible for maintaining the development process and insuring that the team has everything it needs to work effectively.

XP (Beck, 2001; Beck & Andres, 2004) is another popular agile methodology, which focuses on software development practices recommended by the leading practitioners: planning game, small releases, coding standards, stories, automated testing, pair-programming, test-driven development, continuous integration, collective code ownership, automated builds, refactoring etc.

Both, practitioners and researchers are reporting on many benefits of the agile approach. Organisations are claiming that agile leads to improved time-to-market, increased quality, reduced waste, better predictability and better team moral, although not all of this is supported by empirical evidence (Devine, 2008; Erickson, 2005; Schwaber, Leganza, & D'Silva, 2007). The empirical studies report on benefits of agile in the areas of customer collaboration, work processes for handling defects, learning, thinking ahead for management, focusing on current work and effort estimation for developers (Dybå & Dingsøyr, 2008).

These widely recognised benefits are resulting in an increase in popularity of agile methodologies, and have earned them the status of the mainstream software development approach. This was confirmed in the 2010 Forrester report on agile adoption (West & Grant, 2010), where 35% of the survey respondents stated that agile most closely reflects their development process, with the number increasing to 45% with an extended definition of agility. The most recent VerisonOne (2012) annual survey on the state of agile, provides further evidence than "agile is not a fad": 84% of 4048 survey respondents said that their organisation were practicing agile development, 50% worked in companies that have been practising agile for more than 2 years, 48% worked in companies that have used agile across 5 or more teams, and 41% of companies used agile in 6 or more projects.

Given the popularity and the scale of agile adoption, it is not surprising that the industry is seeking more and more graduates with skills in agile and that the demand frequently outpaces the supply (Taft, 2012). According to a recent empirical study on challenges of agile, the lack of suitably trained IT graduates is one of the main problems in the industry today: "We cannot seem to find any graduates who have done anything hands on or even gone beyond one or two lectures on agile methods" (a manager in company "L" quoted in Conboy, Coyle, Wang, & Pikkarainen, 2011:56)

The demand for agile practitioners is not restricted only to software industry. According to the Project Management Institute there is a growing demand in organisations for an agile approach to project management (PMI, 2013). However, the lack of suitably trained agile practitioners is even more prominent problem in the project management area, as agile is only occasionally mentioned in business school curriculums.

The importance of teaching agile at business schools is further confirmed by recent initiatives on "agile business transformation" in organisations such as BT and IBM (Grout & Bonham, 2012), which are aiming to increase the use of agile techniques across the whole business. In addition to that, an increasing number of non-IT organisations are embedding agile principles and techniques in their practice. The latest examples are: the UK Government (NAO review, 2012), US Government (GAO report, 2012), banking sector (Sarran, 2012) and other highly regulated environments such as the pharmacological industry (Fitzgerald, 2012).

This paper reports on experiences from the Hertfordshire Business School, where Agile Project Management has been taught since January 2007 as a separate one-semester module within the Masters programme in Project Management. One of the imperatives in designing the module was to provide students with "hands-on" experience in agile development, as it was felt that the subject is very practical and that only through an experiential approach can students gain knowledge and understanding of "agility". This in turn presented the main difficulty in the module design, due to the diversity of students' educational and work experience. It was not obvious what the relevant common factor of students' domain knowledge was that could serve as a foundation for the "hands-on" learning activities. In other words, students could not simply be asked to develop software, as only a few of them would know how to do it! The "highest common factor" turned out to be the subject they were studying i.e. the agile project management. Hence, the solution was to use the module coursework as a "project" paradigm, where the students were split into small project teams, and guided in how to develop their project in an "agile " way. In order to simulate the software development environment, wikis were chosen as a development platform for the project.

The perceived benefits of adopting an agile approach for teaching agile can be linked to the experiential learning theories of Dewey, Kolb and Piaget, which all state that knowledge develops as a result of direct experience, as well as to Chickering and Gamson's (1987) principles of good teaching practice which are closely related to the values from the Agile Manifesto. More discussion on this will follow in the Literature review section.

The initial solution was first used in the 2007/08 academic year and since then it has been tailored to its present form in four successive academic years through a cyclic process based on Design Science Research (DSR).

Prior to 2010, student feedback from the module evaluation and informal discussions indicated that students value regular and prompt feedback, teacher's interest and enthusiasm for the subject and links with practice. It also indicated that student learning was enhanced with constant review of the subject and early engagement with the module. However, the results (grades) did not fully match student satisfaction with their own learning. Another problem was that cooperation and teamwork remained a problem for some students.