A brief introduction to agile methods and the Collective X research project

Becky Blackshaw
UNITEC Institute of Technology, New Zealand
rblackshaw@unitec.ac.nz


Abstract

This document presents a definition of agility and a brief historical review of the 'Agile Movement'. As an example of agility a brief introduction of XP (eXtreme Programming) and Boehm and Turner's Risk Based approach to development are outlined. An overview of Collective X - the application of the principles of eXtreme Programming) to a teaching environment as a research project is also discussed.

Keywords

Agile methods, extreme programming, teaching methods, agile learning

1. What is Meant by Agility?

Agility in Software Development Circles is defined as "Methods and practices that embrace change" (The Agile Alliance 2001). Agile methods are lightweight and all Agile Methods share similarities. These similarities are:

- Individuals and iterations are preferred over processes and tools,
- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation, and
- Responding to change over following a plan. (The Agile Alliance 2001).

The core concepts include:

- Different projects need different processes or methodologies
- Focusing on skills, communication, and community allows the project to be more effective and more agile than focusing on processes (Cockburn, 2001).

2. A History of Agility

In the beginning (when God created the world) there was chaos- no methods. Out of chaos came the age of methods and the great gurus of structured development Yourdon, Coad, De Marco (to name a few) who brought life cycles and documentation (Highsmith & Orr 2000), but still no resolution to development risk or successful delivery of projects from development into production. Risk and extreme cost have driven radical change in system development methodologies and system development environments. Today most organisations do not undertake large development projects, but buy enterprise-wide off the shelf packages like SAP or PeopleSoft. Object Orientation has become mainstream with wide use of VB, Java, and C++. Rapid Application Development (RAD) and iterative development have taken over from the traditional 'waterfall model'approach. Agile concepts are becoming recognised as 'best practice' in software development.

Agility evolved as a result of the inability of large 'heavy'methodologies to deliver projects successfully. Actual agile thinking has been around longer than we think.
Iterative and incremental software development (the cornerstone of agility) has examples as early as 1970. We have seen the iterative approach in:

- 1930's work of Walter Shewhart (PDSA)
- 1950's X-15 hypersonic jet project
- 1960's Project Mercury (Time boxing)
- 1970's IBM FSD Space Shuttle Software
- 1980's Artificial Intelligence Systems
- 1990 to present SCRUM, XP, DSDM and others (Larman & Basili, 2003).

Agile methods as we have defined them appeared on three different continents in a short period of time during the 1990's:

- DSDM (Dynamic Systems Development) in Europe
- Feature Driven Development in Australia

eXtreme programing is a good example of the amount of radical change in developing systems and can backward map off a traditional development approach.

3. An Example of an Agile Method - A Brief Description of XP

Of all the agile methodologies, eXtreme Programming is the one that has had the most attention because of the leadership of Kent Beck. It has rather dominated the agile world. The roots of XP lay in the Smalltalk community, and in particular the close collaboration of Kent Beck and Ward Cunningham in the late 1980’s. XP became very visible with the C3 payroll project for Chrysler. The first phase of C3 was very successful and went live in early 1997 (Fowler, 2003).

XP begins with four values: Communication, Feedback, Simplicity and Courage. It then builds up to a dozen practices, which XP projects should follow. In particular it emphasises testing (Fowler, 2003). The Principles of eXtreme programing are:

- Rapid feedback
- Assume simplicity
- Incremental change
- Embrace change
- Quality work

The Practices of XP are:

- The planning game
- Small releases
- System metaphor
- Simple design
- Continuous testing
- Re-factoring
- Pair programming
- Collective code ownership
- Continuous integration
- 40-Hour work week
- On-site customer
- Coding standards

4. Some Observations

In the development community there is a polarisation of views (for and against) agile methods. There is general agreement that agile methods are good for some things and not for others. It is clear that agile methods are the NBT (Next Big Thing) and as a result there is coming an evolution of processes and methods for using and teaching agile methods.

5. Boehm and Turner’s Risk Based Approach
Boehm and Turner’s risk based approach as described in the June 2003 edition of the IEEE Computer is a new approach based on agile thinking. This approach recognises the shortcomings in both plan driven and agile approaches and uses risk analysis and the unified process framework to assess what type of development approach is appropriate for a specific project. It also makes an in-depth analysis of both the culture of the organization undertaking the project and does an assessment of team members. This results in a tailor-made development strategy depending on the type of development environment (Boehm & Turner, 2003).

Using the Risk-Based Spiral Model Anchor Points Framework, the Rational Unified Process (RUP) and Model-Based Architecting and Software Engineering Process (Mbase), Boehm and Turner have come up with a five-step process (Boehm & Turner, 2003):

- Risk analysis applied to specific risk areas and three specific categories: environmental, agile and plan driven.
- Analyse risk results to determine ‘project home grounds’.
- If no clear ‘home ground’ the team develops an architecture that supports both ‘agile’ and ‘plan driven’ areas.
- Develop overall project strategy that addresses identified risks and risk resolution strategies. A good look at developer/organisation capabilities is taken here.
- Reflection - monitor, evaluate performance and environment and make adjustments.

Included in the Boehm and Turner approach is a sample application family, which is an agent based planning system. The sample application family provides three representative applications:

- Small, relatively non-critical,
- Intermediate, and
- Very large, highly critical.

For each of these applications a representative planning system (methodology) for development is provided.

6. Why Am I Interested in Agile Methods?

There are changing patterns for professional passing of knowledge. Agile methods focus on people and learning (or passing on bodies of knowledge) and the way people communicate. Managing this incompleteness of communication is a key concept in Agility. Core to agile software development is the use of light-but-sufficient rules of project behaviour and the use of human and communication-orientated rules (Cockburn, 2001). As part of my research project I want to explore three models of passing knowledge (Bob Barbour, personal communication, June 2003):

- Professional Aspect
- Apprentice/Mentor
- Artist/Programmer

As part of my research project, I am very interested in understanding how people pass information from one generation to the next in an agile environment. Is the way we pass on bodies of knowledge in software engineering different? I am very interested in applying the principles of XP to a teaching ‘project’ for which I have coined the term Collective X. The Collective X experiment will take an academic stream and apply the core principles of XP to a teaching environment. These will include pair teaching, shared repository of knowledge, continuous integration of knowledge, the concept of ‘on site customer’ who chooses what they want to learn. The results of the Collective X experiment will form a data set for continued research into agile learning.

References


Copyright © 2003 Blackshaw, B.

The author(s) assign to NACCQ and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The author(s) also grant a non-exclusive licence to NACCQ to publish this document in full on the World Wide Web (prime sites and mirrors) and in printed form within the Bulletin of Applied Computing and Information Technology. Authors retain their individual intellectual property rights.

Copyright ©2003 NACCQ.
Krassie Petrova and Brian Cusack (Eds.)
An Open Access Journal, DOAJ # 11764120 , ( zotero)