Influential Agile Software Parameters

Subhas C. Misra
Carleton University, Canada

Vinod Kumar
Carleton University, Canada

Uma Kumar
Carleton University, Canada

INTRODUCTION

Successful software systems development is a delicate balance among several distinct factors (Jalote, 2002) such as enabling people to grow professionally; documenting processes representing the gained experiences and knowledge of the organization members; using *know how* to apply the suitable processes to similar circumstances; and refining processes based on achieved experience.

Software projects have two main dimensions: engineering and project management. The engineering dimension concerns the construction of a system, and focuses mainly on issues such as *how to* build a system. The project management dimension is in charge with properly planning and controlling the engineering activities to meet project goals for optimal cost, schedule, and quality.

For a project, the engineering processes specify how to perform activities such as requirement specification, design, testing, and so on. The project management processes, on the other hand, specify how to set milestones, organize personnel, manage risks, monitor progress, and so on (Jalote, 2002).

A software process may be defined as "a set of activities, methods, practices, and transformations that people use to develop and maintain software, and the associated products and artifacts."[1] This is pictorially depicted in Figure 1 (Donaldson & Siegel, 2000).

BACKGROUND

Premise of Agile Software Development

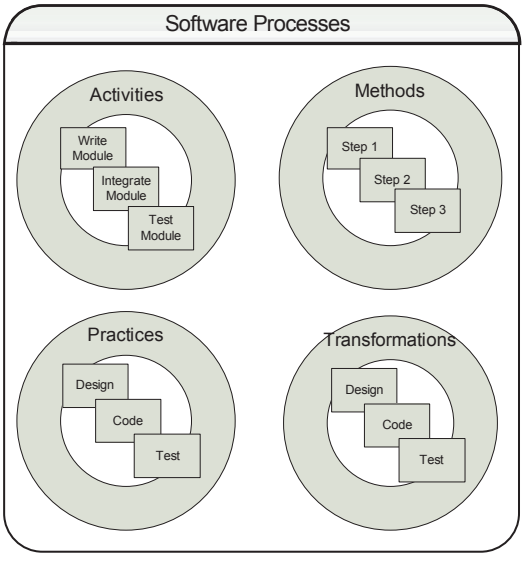
The professional goal of every development team is to deliver the highest possible value to the project and customers. Yet, projects fail, or fail to deliver value, at a frustrating rate due to an increase in process inflation. Plan-driven methods are those in which work begins with the elicitation and documentation of a *complete* set of requirements, followed by architectural and high-level design development and inspection. In this context, the concept of *agile* appeared where principles and values were shaped as a way to help teams avoid the cycle of process inflation and to focus on simple techniques for reaching their goals.

Agile processes allow adjustments of requirements during all phases of the development cycle and stress collaboration between software developers and customers and early product delivery (Donaldson & Siegel, 2000).

Key motivations of *agile methods* apparition are

- Iterative development is of lower risk than waterfall development (Larman, 2004).
- Early risk discovery and improvement.
- Promotes early change: consistent with new product development.
- Early partial product apparition.
- Satisfaction through early and repeated successes.
- Continuous testing activity.
- Final product matches client's desires better.

Figure 1. Software processes



Features Of Agile Software Development

The core of the "Manifesto for Agile Software Development" (n.d.) is as follows (Fowler, 2002; Fowler & Highsmith, 2001; Martin, 2001):

- individuals and interactions over processes and tools;
- working software over comprehensive documentation;
- customer collaboration over contract negotiation; and
- responding to change over following a plan.

The central aspects of agile methods are simplicity and speed. These goals can be achieved by software whose development is incremental, cooperative, straightforward, and adaptive.

The agile software development approach considers people the main resource of the development. In this context, its approach is collaborative considering that software development is, in fact, a collaborative team activity. In this way it steps away from the individualistic software engineering paradigm, and instead considers software development as a collaborative team activity. This enables agile software development teams to learn how to work together and thereby provides a mechanism for resolving the inevitable misunderstandings that occur during a project.

The set of consistent approaches that arise from agile software development processes are (Abrahamson, Salo, Ronkainen, & Warsta, 2002; Fowler & Highsmith, 2001; Larman 2004)

- human resource issues,
- amount of documentation to be as reduced as possible,
- communication is a critical issue, and
- modeling tools are not as useful as in other development processes.

Agile processes characteristics are stated as follows (Fowler & Highsmith, 2001):

- modularity is used on development process level;
- iterative activities with short cycles enable fast verifications and corrections;
- time spent with iteration cycles takes from 1 to 6 weeks;
- temperance in development process that removes all useless activities;
- adaptive;
- incremental; and
- collaborative and communicative working style.

There are many agile methods sharing common characteristics and starting from the same approach. Some of them are: *extreme programming, crystal family of methodologies, rational unified process, dynamic systems development method,* and *adaptive software development.* Each of them has its own processes, principles, practices, roles, and responsibilities, but they all have in common the agile approach.

INFLUENCE PARAMETERS OF AGILE SOFTWARE DEVELOPMENT PROJECTS

In agile approach it is quite hard to distinguish between success and failure factors of projects. What was considered decisive for a project echoing success may be considered as a limit in another one. At the same time, a certain factor may influence not only a project, but may also persuade the rest of agile factors. In this context, it is difficult to draw a fixed and clear line between influence factors and how they act in agile software projects. Different factors[2] that influence the success or failure of software projects are shown in Figure 2 and are described hereafter.

Organizational Factors and Their Influences

- **Organization Culture:**
  1. A dynamic, agile organization will find agile methods extremely suitable for it (Abrahamson et al., 2002). Importance of customer feedback and control on an agile project requires an adaptive and collaborative working environment. In this working environment, which is, in fact, the organization, all its members, for example, management, developers, and testers must be in total agreement to use agile processes, because without organization commitment to being agile, failure may develop into a strong possibility (Smith & Pichler, 2005).
  2. In a bureaucratic organization where respecting plans, rules, and directives are a way of work, agile is inappropriate as a new path in creating value. A stagnant organization with a culture believing in efficiency, control, and process rigor tends to neglect the fact that success evolves from new successes, failures, and different alternatives used. The deeper these factors are grounded in the organizational culture, the more difficult they are to revolutionize, and the more easily they can became obstructions in adopting new approaches (Highsmith, 2000).