Available online at www.sciencedirect.com

ScienceDirect

Advances in Engineering Software 39 (2008) 654–658

ADVANCES IN ENGINEERING SOFTWARE

www.elsevier.com/locate/advengsoft

An adaptive software development process model

M. Rizwan Jameel Qureshi *, S.A. Hussain

Department of Computer Science, Comsats Institute of Information Technology, Lahore, Pakistan

Received 7 September 2006; received in revised form 24 August 2007; accepted 31 August 2007
Available online 24 October 2007

Abstract

The concept of agile process models has gained great popularity in software (SW) development community in past few years. Agile models promote fast development. This property has certain drawbacks, such as poor documentation and bad quality. Fast development promotes use of agile process models in small-scale projects. This paper modifies and evaluates extreme programming (XP) process model and proposes a novel adaptive process mode based on these modifications.

Keywords: XP; Adaptive software development; SDLC; OSSD; CBD

1. Introduction

Agile process models stress on agility for software development. Agility means responding to changes quickly and efficiently. Possible changes required in software projects are in requirements, budget, schedule, resources, technology and team. These are "reacting" changes on which agile models stress for a successful software project. The agile golden principles defined in agile alliance meeting in 2001 [1] are:

1. Satisfy customer through early and continuous increments.
2. Deploy first increment within couple of weeks and the whole software within couple of months.
3. Customer and agile teams must work jointly daily throughout the project.
4. Agile team and customer must have face-to-face meetings.
5. Welcome requirements even in late phases of the system development.
6. Trust and respect must be maintained among agile team members.
7. Velocity of the project must be measured after delivery of each increment.
8. Emphasis should be on good design to increase agility.
9. Best architecture and design always come out from self-organization.
10. Adjust and tune according to the situation.
11. Whole development process must follow keep it simple (KIS) principle.
12. Agile project needs consistent work until completion.

The objective of above mentioned principles is to have adaptive software development only for simple and small size software projects. There is no indication to make process model adaptive according to the nature of projects. An analyst has to select traditional software process models if the software is complex in nature (such as spiral and rational unified process models).

Section 2 of the paper describes related work about agile models. Section 3 covers the motivation for adaptive process models. Section 4 proposes an adaptive process model for agile and traditional software development. Section 5 describes how adaptive process model is better than the agile models.

* Corresponding author. Tel.: +92 42 5431602; cell: +92 3334492203.
E-mail addresses: anriz123my@yahoo.com (M. Rizwan Jameel Qureshi), asadhussain@ciitlahore.edu.pk (S.A. Hussain).

0965-9978/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2007.08.001

2. Related work

According to Highsmith et al. [2], all the agile process models stress on the need of quality design. Extreme programming (XP) is the most widely used model among all agile models, such as dynamic system development methodology (DSDM), adaptive software development (ASD), Scrum, Crystal and feature-driven development (FDD). The prominent phases of XP model are planning, design, coding and testing. Agile models focus on delivering the first increment in couple of weeks and complete SW in couple of months. The main features of all models are fast development and cost saving. Fast development leads to poor quality SW and incorporates all disadvantages of rapid application development (RAD) process model.

Lycett et al. in [3] have maintained that, developers are confused to shift from traditional process models to agile process models. Software organizations have invested huge amounts of money to achieve quality standards (International Organization for Standardization (ISO) 9000 and Capability Maturity Model (CMM)), by using standardized process models. The authors are of the view that agile development models have challenged suitability and effectiveness of standard process models. Main disadvantages of agile models are poor quality product, poor design and improper documentation. The authors in this paper portray disadvantages of agile models as their advantages.

The comparative analysis in [4] is performed by means of method's life-cycle coverage, project management support, practical guidance, fitness-for-use and empirical evidence. An agile process model deals with all main phases of system development life cycle (SDLC) and most of them do not provide enough support for project management. Empirical results in this paper are incomplete and are not enough for a complete hypothesis.

Lan Cao [5] has modeled dynamics of agile software development process and has explored the implementation and usefulness of agile methods. The research also investigates influence of agile process models on functioning of software from quality, schedule, cost, and customer's satisfaction perspective.

Highsmith et al. in [6], mention function of agile teams for success of software projects. Main characteristics of agile teams are competency, problem solving, decision-making, collaboration, common scope, trust, respect and self-organization. Agile process models are not appropriate to develop all types of software [6]. These process models are not suitable for large teams and less experienced developers. The authors have made these claims without any sound reasons.

Agile software development and component based software engineering are two different domains [7]. The authors proposed a model using an integrated approach for agile component development. It is not for agile software development; rather it introduces agility in component based development (CBD).

Kuppuswami et al. [8], conducted a study to examine effects of extreme programming (XP) process model on development effort. They developed a simulation model to deduce their results. Fifty user stories were simulated with this model. The results showed that an increase in usage level of individual XP practices reduces development cost. The hypothesis made by the authors yet needs to be proved in real life environments.

Agile process models can be adapted to incorporate software standards such as ISO and CMM [9]. Even after confirming the standards, SW process keeps agile property. The adaptation process is an attempt to bridge the gap between traditional and agile process models. Heavy documentation required to achieve SW quality standards is against the principles of agile software development. Production of heavy documentation is a time consuming activity. It delays the increments, which ultimately delays the whole project release time.

The model in [10] reduces the threats and risks of agile process models for certain projects. Agile models are well suited for specific nature of projects but are not suitable for all types of projects.

Noble et al. in [11] believe that software engineering process models have moved from heavy weight methodologies, such as RUP, to light weight methodologies, such as agile. The authors used agile process models in students' projects. Results showed that students developed software with less effort compared with traditional process model approach. The research examined ways of introducing agile methodologies more steadily in initial courses of their degree programs.

Hazzan et al. [12], have made two suppositions.

- Changes payback to societies and groups that accept it.
- Change is a basic aspect of worldwide system development.

These two suppositions are based on factors such as sex, supervision etc. [12]. The other factors not considered are; requirements, budget, schedule, resources and technology. These changes mentioned have significant impact on agile software development.

Cohan et al. in [13], have proved that a transition from traditional process models to agile process models have an affect not only on software engineering group but also on other groups, departments and management. The common recommendations to initiate an agile process model are [13] are:

Developers: Developers who are too excited for changes or oppose changes cause an agile project to fail.

Managers: Managers who do not interact with their members on daily basis also cause a project to fail. Managers should remove all problems reported by team members.

Shift from traditional to agile process model: There should be a steady shift from traditional process model to agile process model otherwise there is a risk of failure.

Distributed development: Companies should avoid distributed development sites after shifting to an agile process model for at least first two to three months.