Chapter 7
Toward Agile Interactive Software Development Process Models for Crowd Source Projects

Izzat Alsmadi
Yarmouk University, Jordan

Saqib Saeed
Bahria University, Pakistan

ABSTRACT

Typical traditional software development models are initially designed for company-style software project teams. They also assume a typical software project that has somewhat clear goals, scope, budget, and plan. Even Agile development models that are very flexible in considering previous project parameters assume somewhat stable team and project structures. However, in recent years, the authors have noticed expansion in software projects that are developed in a very illusive flexible team, scope, budget, and plan structures. Examples of such projects are those projects offered in open competition (also called crowd sourcing) structure for software developers to be part of. In typical open competition projects, initial, high level project ideas are submitted to the public through the Internet. The project initiators give their initial requirements, constraints, and conditions for successful products or submissions. Teams can be organized before or through the competition. Submission and evaluation of deliverables from teams are subjected to project initiator evaluation along with evaluation teams organized through the open competition host. This chapter investigates all traditional project characteristics. The authors elaborate on all those elements that should be modified to fit the open competition agile structure. They use several case studies to demonstrate management issues related to managing software projects in open competitions.

DOI: 10.4018/978-1-4666-4301-7.ch007

INTRODUCTION

The evolution and expansion of the Web makes it easy and convenient to move many human activities to this venue that connects humans around the world together. Even governments are making use of some websites to get feedback from the citizens as a whole (i.e. the crowd). For example, websites such as: http://www.ideastorm.com, http://ideascale.com, and http://www.uservoice.com/to encourage citizens to submit their opinions and ideas to their government to be assessed and evaluated. In this scope of software projects, our focus in this chapter is in the software projects styles aside from the traditional business style companies. In this chapter, we will discuss issues related to challenges of managing software development projects in untraditional styles where all project major elements (i.e. team, scope, budget, success factors, software development model) are not clear. In the first section, we will summarize current software development models taking those major elements into consideration. The second section will discuss software open competition projects. We will list some of the known websites who offer open competition software projects. We will elaborate on their marketing strategies. Using the same major project attributes mentioned earlier (i.e. team, scope, budget, success factors, software development model, etc). In comparison with traditional software development projects and using case studies, we will demonstrate how traditional software development models will fail short in conducting successful open competition projects. The last section will include a proposal for a new open competition model or at least possible enhancements for traditional software development models. Case studies will be also used here to demonstrate the proposed models. Below are initial possible references that will be used in the chapter.

Software process is a set of complex activities required to develop software systems. There are many known software process models such as water fall, spiral, incremental, Rational Unified Process (RUP) and agile models. All those process models carry out same mandatory activities e.g. requirements elicitation, software design, coding, testing, evolution and project management. However, they are different from each other based on the way they perform such tasks. For example, a water fall model follows a straightforward process that starts from requirement. Once requirement stage is finished, design stage is started and so on until finishing all software activities. Such traditional model does not allow back links from a stage to the previous stages. Furthermore development stages are distinctly divided and separated and it does not allow overlapping of stages. On the other hand, most of the other development methodologies are flexible in this manner and allow incremental cyclic process where in each cycle a small part of the project is developed and all process activities are executed in a cyclic manner.

Figure 1 describes typical software process activities. Project initiation and planning are the early tasks in a software development lifecycle where the project boundaries and feasibilities are generated. After the approval of customer and developers, detailed level planning is carried out by development team, focusing on estimation of cost, resources and scheduling. In the next phase system and user requirements are gathered, documented, prioritized and finalized. The major deliverable of this initial stage is software requirement specification (SRS), which highlights system and user requirements. In this initial stage, problem domain is analyzed looking for why we are developing this software, who asked for this solution, what were the problems in the domain or in the existing system (if a system exists). Software and system analysts are the people who usually perform activities in this task or stage. They meet with domain users and stake holders in trying to analyse the current system and specifying requirements for the sought solution.