Chapter 6.2
Best Practices Guidelines for Agile Requirements Engineering Practices

Chetankumar Patel
Leeds Metropolitan University, UK

Muthu Ramachandran
Leeds Metropolitan University, UK

ABSTRACT

Developing software that meets the customers or stakeholders' needs and expectation is the ultimate goal of the software development methodology. To meet their need we have to perform requirement engineering which helps to identify and structure requirements. In traditional software development methods end users or stakeholders predefined their requirements and sent to the development team to analysis and negotiation to produce requirement specification. In many cases it is risky or very difficult and not economical to produce a complete, verifiable set of requirements. Traditional software development has a problem to deal with requirement change after careful analysis and negotiation. This problem is well tackled by the Agile Practices as it's recommends an on-site customer to represents their requirements through user stories on story cards. Generally customers have rarely a general picture of the requirements or system in their mind which leads problems related to requirements like requirements conflicts, missing requirements, and ambiguous requirements etc, and does not address non-functional requirements from exploration phase. This chapter introduces best knowledge based guidelines for agile requirements engineering to enhance the quality of requirements (story cards).

DOI: 10.4018/978-1-61350-456-7.ch6.2

1. INTRODUCTION

Developing software that meets the customers or stakeholders' needs and expectation is the ultimate goal of the software development methodology. To meet their need we have to perform a requirement engineering step, which is one of the crucial steps in to software development methodology. Overall project success and failures of the project is depending on the user requirements. Requirements elicitation process is one of the challenging processes in the software development methods. In traditional software development methods end users or stakeholders predefined their requirements and sent to the development team to analysis and negotiation to produce requirement specification. Traditional software development has a problem to deal with requirement change after careful analysis and negotiation. This problem is well tackled by the XP, which is one of the agile software development methodologies.

Extreme (XP) programming is a conceptual framework of practices and principles to develop software faster, incrementally and to produce satisfied customer. It is a set of twelve practices and four principles, which makes XP successful and well known among all the agile software development methods. The goal of XP is to produce the software faster, incrementally and to produce satisfied customer (Beck 2000). According to Boehm the cost of change grows exponentially as the project progresses through it lifecycle (Boehm 1981). The relative repair cost is 200 times greater in the maintenance phase than if it is caught in the requirement phase (Faluk 1996). XP maintain the cost of change through iterative software development methods and Refactoring.

In XP, Development starts with planning game where customer writes user stories on story cards. Those cards are estimated by the developer, based on those estimation customer priories them depends on their needs to establish a timebox of an iteration. Developers develop those story cards through pair programming and test driven development. At last customer provides acceptance test to accept the developed functionality. In between they consider all of the XP practices in mind to improve the quality of the software.

Story cards are one of the important aspects of the XP. They are playing vital role in XP. It describes functionality of system or software to be build that will be valuable to either purchaser or user of software. User stories are composed of three aspects (Cohen 2004):

- A written description of the story used for planning and as a reminder
- Conversation about the story that serves to flush out the details of the story
- Tests that convey and document details and that can be used to determine when a story is complete

Story cards are written by the customer in XP to articulate their business needs. According to Cohn story cards must be testable, estimatable, valuable to the customer, small and independent (Cohn 2003). These story cards must be written by the customer because they know their business need very well compared to developer.

XP strongly recommend an onsite customer to write their business need. Business is well understood by the customer. But generally customers have rarely a general picture of the requirements or system in their mind (Kotyana and Somerville 1997). Traditional XP story card framework or template is not well defined for the requirements elicitation. It supports to write requirements or user needs in two to three sentences and it not discover any information rather than user functionality. Different stakeholders have different needs. End user has rarely a picture of a clear of system to write down the user stories. This will lead to problems related to requirements like requirements conflicts, missing requirements, and ambiguous requirements etc, and not address non-functional requirements from exploration phase. Due to this reason they hardly make a decision or