Chapter 50
Software Maintainability Estimation in Agile Software Development

Parita Jain
Amity University, Noida, India

Arun Sharma
IGDTUW, New Delhi, India

Laxmi Ahuja
Amity University, Noida, India

ABSTRACT

Agile methodologies have gained wide acceptance for developing high-quality products with a quick and flexible approach. However, until now, the quality of the agile process has not been validated quantitatively. Quality being important for the software system, there is a need for measurement. Estimating different quality factors will lead to a quality product. Also, agile software development does not provide any precise models to evaluate maintainability. Therefore, there is a need for an algorithmic approach that can serve as the basis for estimation of maintainability. The article proposes an adaptive neuro-fuzzy inference system (ANFIS) model for estimating agile maintainability. Maintainability is one of the prominent quality factors in the case of agile development. The proposed model has been verified and found to be effective for assessing the maintainability of agile software.

1. INTRODUCTION

In the engineering of software systems, agile development approach is becoming a popular approach for many software projects day by day. Agile practices are becoming attractive due to faster development approach while maintaining a level of customer satisfaction. Over the past several years, agile methodologies have gained wide acceptance, with maintaining standards and execution approach for develop-

DOI: 10.4018/978-1-6684-3702-5.ch050

ment. The practice promotes continuous iteration of development and testing throughout the software engineering process, accommodating rapid changes as given by Abrahamsson (2017). The agile software development lifecycle starts with requirement elicitation as shown in Figure 1, identifying functional requirements which are fragmented into the set of user stories that are further disintegrated into a smaller task if required. The agile methodology includes an idea of iterative cycles of multiple releases. The benefits arise from improved customer collaboration, responding to change and refactorization of codes, daily standup meetings, retrospection meet after each sprint release.

With the success of agile development on small to medium scale projects where agile methods prove to be a best practice for software development, has led to an intended expansion for large scale and complex projects. Various studies provide evidence for successful completion on small-scale software development projects also. For large scale projects, the problem exists with the quality of the product. Researchers are still not able to prove quality development for such type of projects in a holistic manner.

Quality being an important concern nowadays, the relevant software development process is a necessity. From different developers, it has been observed and found that if the quality of the process is as per the standards then the high quality of the product can be expected as high. While adopting a quick and flexible approach like agile practices is a good option, with the proof that it will lead to a quality product development (Jain et al., in press).

Software process quality used for engineering software systems significantly affects the value of customer satisfaction. Hence, more importance must be given to software process quality rather than software product quality. However, an ample number of studies show that still, researchers are in a way to prove agile practices as a quality development process quantitatively founded by Jain et al. (2016). Quality of the product depends on various different factors. With the identified quality factors for agile development process given by Jain et al. (2016) based on ISO 9126-1 quality model using the ISM approach, the foremost quality factor is maintainability. In the development of software development, maintenance is considered an important activity for software capabilities enhancements and optimization.

The formal definition of software maintenance' given by IEEE (1998) is "Modification of a software product after delivery to correct faults, to improve the performance or other attributes, or to adapt to a modified environment." Maintainability, in general, refers to how these operations can be carried out effectively. For the whole development life cycle, it has been observed that up to 60–70% of effort and time is utilized in the maintenance of the product, making it the most expensive activity in the development lifecycle of software given by IEEE (1993). Marco (1986) suggested that we can control this cost only by measuring it. Influential critical factors have to be determined for measuring the software product. However, measuring these factors is a bit tough to get the final value of maintainability.

In today's era, an ample number of software companies use the agile methodology for the development of software products as it allows efficient software maintenance. With the rise of agile software products, there is a need for effective maintenance aspect. Researchers are constantly trying to devise a method to forecast the maintainability of any software in prior phases of software development life cycle (SDLC) by measuring the characteristics of its design. Agile Software Development does not provide any precise model basis to evaluate estimation. Therefore, there is a need for an algorithmic approach that can serve as the basis of estimation.

The present work propounds an adaptive neuro-fuzzy inference system (ANFIS) model for estimating maintainability of the agile software product. Based on experts' opinion fuzzy if-then rules are constructed in the fuzzy inference system. Although, this will lead to time consumption. ANFIS has an advantage over FIS systems that is, for solving a problem ANFIS combines fuzzy logic with the neural