Chapter 36

Software Process Paradigms and Crowdsourced Software Development: An Overview

Nitasha Hasteer
Amity University, India

Abhay Bansal
Amity University, India

B. K. Murthy
Centre for Development of Advanced Computing, India

ABSTRACT

Production of quality software requires selecting the right development strategy. The process and development strategies for creating software have evolved over the years to cope with the changing paradigms. Cloud computing models have made provisioning of the computing capabilities and access to configurable pooled resources as convenient as having access to the common utilities. With the recent advancements in the use of social media and advent of software development through crowdsourcing, the need to comprehend and analyze the traditional process models of software development, with regard to the changed paradigm have become ever more necessary. The changes in the way software are being created and the continuous evolution in the processes of development and deployment has created a need to understand the development process models. This chapter provides an insight on the transition from the conventional process models of software development to the software development methodology being used to develop software through crowdsourcing.

DOI: 10.4018/978-1-5225-8182-6.ch036
INTRODUCTION

Software is a product and also a vehicle for delivering a product. Engineering of this Software is termed “Software Engineering”. The term Software Engineering was coined at a NATO Conference in 1968 (Naur & Randell, 1968) and is defined as the systematic approach to the development, operation, maintenance and retirement of the software. The systematic approach enables the production of high quality software and focuses on the activities directly related to the production of the software and thus these activities form the core of a software project. The approach defines the tasks and the order in which they should be executed. The software development approach that we deploy plays a critical role in the quality of software being developed. A non-systematic development approach will produce software of low quality that may not meet user expectations and also might incur additional cost of development. Software engineers use software development models for creating fault free software that meets the requirements of the end users and is delivered within the pre specified time limit and budget. Following a non-systematic approach and developing software without the use of development models, not only violates the budget and time deadlines but also results in creation of low-quality software that would be difficult to maintain.

Software development is a creative and ever evolving area. Organizations use various software development process models and methodologies for developing software. A Software Process Model defines the stages in which a project should be divided, order of execution of these stages, and other constraints and conditions on the execution of these stages (Pankaj, 2010). The process models are based on the paradigms of software development and have undergone evolution from their inception to the present times. From the first published water fall model in 1970 to the Agile Software Development Methodologies until the recent times, the software industry has seen the evolution over years to help the industry cope up with the changing scenarios. There exists some generic process models and many variations of these generic process models have been proposed and in practice a combination of different models may be used. A generic process model defines five key activities: communication, planning, modeling, construction and deployment (Pressman, 2014). Each generic model describes a process flow that describes the activities and tasks along with their sequence of occurrence.

The computing paradigm has seen a revolution with the advent and popularity of distributed computing. Gone are the days when organizations used to buy additional hardware to increase the computing capabilities of the systems. Cloud Computing has much changed the way organizations operate and do business. According to the definition given by the US National Institute of Standards and Technology (NIST), Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for e.g., networks, servers, storage, applications and services) that can be quickly provisioned and released with minimal management effort or interaction of the service provider (Mell & Grance, 2011). This framework provides three service models and four deployment models. The service models are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The deployment of the distributed network can be as a Public, Private, Hybrid or a Community cloud. The software that is created for the cloud is designed and developed to cater to the multi tenancy architecture. The task of creating such software requires process models and frameworks that could incorporate all the security requirements of the multi tenancy environment (Hasteer, Bansal, & Murthy, 2013). The development methods, tools and models for engineering cloud services needs attention and must be addressed to realize the potential of the technology completely as the cloud drives new needs.