Innovative Approaches in Pair Programming to Enhance the Quality of Software Development

Naresh E., Jain University, Bangalore, India
Vijaya Kumar B.P., Ramaiah Institute of Technology, Bangalore, India

ABSTRACT

The article tries to shed some light on the impact of human psychology on the effective use of pair programming in the modern Software development lifecycle such as SCRUM, Extreme Programming which are in turn used on heterogeneous software projects. This article also tries to identify that if the authors try to pair people keeping their psychology in mind that pair can come up with code with fewer defects, with efficient code, if the paper tries to pair people randomly without any planning or thinking might create a pair which let aside create inefficient code and lead to be unproductive nature, and even it will create a negative impact on the project and the team. This article introduces a few novel approaches in framing the pairs in pair programming’s like known and unknown pairs, coder and reviewer pair and coder and tester pair. Applying the described approaches, an industry can improve the quality of the delivered product and improve the efficiency of software engineers.

KEYWORDS

Code Quality, Coder and Reviewer Pair, Effective Pairs, Known and Unknown Pairs, Pair Programming, Software Engineer’s Attitude

INTRODUCTION

Pair programming is one of the main features of agile software teams. Two people work together, on a single system, to develop a certain module of code. One member acts as the driver, who writes the code, and the other, is the navigator, who reviews the code as it is being written. These roles keep getting switched between them. This concept is used in most agile teams.

One of the major problems with pair programming is when a decision has to be made to see which among the pair has done more amount of work. As they work as a pair, the output is also considered to be accumulated work of both the members, and it is difficult to decide the better among the two, for making promotion decisions, or decisions involving improving the efficiency of software production.

DOI: 10.4018/IJICTHD.2018040104

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
BACKGROUND

Pair programming is a practice, where two programmers work on a task. An experiment was carried on pair programming in industry and academics and obtained good results. The authors suggested a new approach by pairing different levels of engineering students by applying both qualitative and quantitative approaches (Laurie Williams, Robert Kessler, 2002; Kim Nilsson, 2002; Venkata Vinod Kumar Padmanabhu, Hari Praveen Tadiparthi, Muralidhar Yanamadala, Sagar Madina, 2012; F. Zieris, 2015). Pair programming include the principles like pair pressure, negotiation, brainstorming, pair courage, pair reviews, pair debugging, pair learning and training (Williams, Laurie, 2001).

Pair programming was reported many benefits as compared to individual programming. Most research reports from industry and academia represent positive effects of Pair Programming on programmers’ performance and quality of the software. Williams et al. applied Pair Programming concept to the classroom and studied its impact on students for execution of project modules. The results found that collaborative environment of Pair Programming aids students to achieve desired learning outcomes, more confident and to get better grades in programming assignments (Cockburn, Alistair, Williams, Laurie, 2000; Williams, L., McCrickard, D.S., Layman, L., Hussein, K, 2008; B. Isong, T. Moemi, N. Dladlu, N. Mothlabane, O. Ifeoma and N. Gasela, 2016).

A thorough survey of pair programming shows that psychological factors, attitude, communication and complete education background of the students are compatibility factors which impact on the efficiency and effectiveness of pair programming. Study reports discovered that the member should be paired with a partner having similar or higher skills in order to achieve better performance. There is limited of research work carried with respect to social factors such as soft skills, etc (Norsaremah Salleh, 2008; Silvana Faja, 2011; Saiqa Anjum, Hibba Batul and Mehreen Sirshar, 2015; S. Xinogalos, M. Satratzemi, A. Chatzigeorgiou and D. Tsonpanoudi, 2017).

Distributed Pair Programming (DPP) model allows two engineers to collaborate remotely in order to apply the Pair Programming technique from different locations (Despina TSOMPANOUDI, Maya SATRATZEMI, Stelios XINOGALOS, 2015; Brian F. Hanks, 2004). In (Kim Man Lui and Keith C.C. Chan, 2006), authors proposed an experiment called repeat-programming which provides the understanding of relationships between programming productivity and human experience. To illustrate, we have examined the productivity in pair programming as a case study. From the results of the case study, we are able to present a case in which beginner-pairs against novice-solos are high productive than experienced-pairs against experience-solos.

In (Laurie Williams, Robert R. Kessler, Ward Cunningham and Ron Jeffries, 2000; Robert C. Miller, Haoqi Zhang, Eric Gilbert, Elizabeth Gerber, 2014) authors proposed a new interaction method called ‘pair research’ where programmers are paired weekly to work on each other’s project mutually. The authors assess the results from the deployment of two projects and results shows that the pair research used many ways for brainstorming, usability testing, data collection and analysis.

In (Terry, J.E., Williams, P.A.H. and Mahnckel, 2006; C. McDowell, 2006), student’s experiences and learning perspectives are evaluated for 1st year information science students through pair programming technique. Think-Pair-Share (TPS) is a classroom-based active learning strategy, in which students work on a problem proposed by the teacher, first individually, then in pairs, and finally as a class-wide discussion. Authors experimented with the first year CS1 subject and applied the TPS strategy and the results shows that the students effectively engaged for 61% during think phase and 70% during pair phase (Aditi Kothiyal, Rwitajit Majumdar, Sahana Murthy and Sridhar Iyer, 2013).

In (Naresh E and B. P. Vijaya Kumar, 2016), authors introduced Community based Approach using pair programming concept at various levels in the organization, which in turn helps for good team management, increases the productivity of an engineer and reduces the overall cost of software development.
Related Content

The Current State of Dialogue in South African Universities: Change through Open Conversations and the Facilitating Role of ICTs
[www.igi-global.com/chapter/the-current-state-of-dialogue-in-south-african-universities/135856?camid=4v1a](www.igi-global.com/chapter/the-current-state-of-dialogue-in-south-african-universities/135856?camid=4v1a)

Including Users with Motor Impairments in Design
[www.igi-global.com/chapter/including-users-motor-impairments-design/13140?camid=4v1a](www.igi-global.com/chapter/including-users-motor-impairments-design/13140?camid=4v1a)
The Language of Technoself: Storytelling, Symbolic Interactionism, and Online Identity
www.igi-global.com/chapter/language-technoself-storytelling-symbolic-interactionism/70348?camid=4v1a

A Strategy Framework for Digital Heritage
www.igi-global.com/chapter/strategy-framework-digital-heritage/50284?camid=4v1a