

Software quality management

Kartheek

Department of Software Engineering

9304163778

BTH, KARLSKRONA

chilla.kartheek87@gmail.com

Abstract

The paper mainly focuses on the selecting two article that are from same quality area, same tool and same method are used to describe about how is software quality maintained. The two articles chosen describe how pair programming in software design, coding and development show impact on quality of the software product. Both the articles describe impact of pair programming on quality. Both use the same case study empirical research method. Below described the solutions discussed in the articles. The strength and weakness within the articles are described as well.

i. Introduction

1.1. Pair Programming

Pair programming is a part of extreme programming (XP) which involve two people working on computer. One person is involved in coding and the other scrutiny the code. The person coding is called as driver and the other is called as navigator. The main theme of driver is to produce the code. The navigator is mainly the barnstormer for the driver. His line of work involves finding the errors in the code. Finding information and justifying overall structure of the code is his role. Both the article helps to give relevant information and how do they impact the quality of the software product when this pair programming scheme is implemented. The selected articles are as follows.

1) A Multiple Case Study on the Impact of Pair

Programming on Product Quality [1].

2) Pair programming in software development teams –

An empirical study of its benefits [2].

ii. Problem Domain and Scope

Problem domain in both the articles are discussing about the impact of pair programming on quality of the software product.

Scope of article 1: It is carried out in industrial setting. Author main focus on how pair programming adopted

and influenced within industrial settings, and to find if it shows impact on product quality. It focuses on four different developing projects where the impact of pair programming over quality. Four multiple controlled case studies were conducted in actual software projects to understand the impact on quality

Scope of article 2: It is carried out in university settings but not industrial. This workshop involved 13 development teams working in pair and solo to develop the software project. It involved dividing the development teams into two set of teams with different environmental conditions. 95 students from university of Dortmund have participated in two studies. One involving card game and the other one involve quiz. They are given tasks in group and solo to assess the quality (density of defects) and productivity.

iii. Motivation for choosing the articles.

Does the pair programming show impact in reducing the defect density, which in turn improves the quality of the code? Reduce in the defect density by using the pair programming instead of using solo programmer to work on code is the research area where it is keen to know if there is any improvement in the quality, if so, which one is followed to achieve the improvement. Generally, Pair programming means one person handles mouse or keyboard to write the code. The other person scrutinizes or verifies the code for any modifications or improvements to reduce the bad smell codes (ineffective Lines of Code).

iv. Base for References used in articles.

Article1: The motivation for conducting the study is described aptly. Some authors in the article refer to improve in quality while using pair programming. Other authors say no such significant impact in quality with the use of pair programming. The corresponding references are stated correctly. This article selected made a significant contribution behalf of impact of pair programming over quality. The selection of empirical case study is not justified in the article. Author didn't explain why the other methods are not valid in this case.

The author stated the main object of the article by clearly stating the research questions. The author describes the existing knowledge then the relevant main findings are explained in three categories from the knowledge gathered. The author also described in diagrams about the empirical methods used in the past and also the main contribution area of the articles in previous years are shown. Scale of influence in percentages are given for showing the impact of pair over solo programming. The research process to come to conclusion are discussed elaborately. The author clearly shattered which author supported and not supported the quality impact by pair programming. However most of the papers hared knowledge on the productivity and effort expenditure quality impact is significantly less discussed. All the project related and product related finding are stated clearly. The pair programming effort in the four case studies are well stated. The productivity of pair and solo are well stated. The motivation for selecting defect density to identify the quality in the code is not stated. Different views from the participants is taken as well about the pair impact. All the empirical results are stated and output are carried out clearly. The author described elaborately about the four case project.

Data collection is mentioned clearly and values are provided. These defect densities are identified in the pictorial representation which show the impact over pair programming. Only one threat to validity is discussed others are not stated. Conclusion is well stated. **Article2:** There is no clear motivation for selecting the case study as research method. Some of the references that are used in the articles are same and also act as a strong base for the article. The author didn't state clearly the main objective and the research questions are not clearly specified but they are described as hypothesis. Here he involved a complex task and divided the task for a team with pair and the other team with individual students. The author says the previous research involve experimenting on one task only. The author gathered information on the various studies related to the pair programming. Not very specific on how the selected articles impact on the quality are not well described. Later all the characteristics related to the pair programming are detailed like experience level, implementation time. Good justification is done for why selecting the article and how they were useful in his research. The pair programming implementation is detailed clearly. They have shown the data but unlike previous article with diagrammatic view on past research in a detailed way as well. The author did the same project at same time with different teams this help to understand real time how the people are able to solve the task. The difference in the lines of code, defects, implementation time are well described completely elaborate in detail.

Some disadvantage like efficiency are also discusses along with impact over quality which is useful information.

Limitations are well described. The threats to validity are not discussed. The conclusion is consistent with the research. The survey is carried out with questionnaire to support the obtained results although are useful for subjective measures only.

v. Solution and contribution

Article1: The pair programming out of 4 cases 2 cases show no significant difference. One case study shows significant improvement. The density of deviation in both the case in quality terms is higher in case of pair programming which is shown via bar diagram. The article describes the deviation in using the code standards in both pair and solo programming is same. Theses solutions are motivated from the analysis and synthesis of the data collected. Solutions with respect to 6 metrics PP effort percent, productivity, rationale for PP, density of coding standard deviation, comment ration and relative defect density are gathered and the both the existing research and the new finding are well described in the table which is correct data and no false data. The solution obtained in the article are support the researchers except the relative defect density where there are conflicting results. There is no impact of pair programming over defect density. Thus result conflict with the article 1, both the papers have relevant data to prove it. Thus this article is important to support that there is no significant impact over quality. In my opinion the quality is not always impacted by pair programming there are several factors that ae needed to be considered. Proving that it works for small systems doesn't mean that the results are same for the complex task. New research finding in this article will be vary useful for future work.

Article2: The solution in the article is that the pair programming show impact over the quality of the product. Here the solution prove that pair programming help to make the code less complex readable and understandable unlike the solo programming which is important key information to be noticed with respect to how the code is being handled in pair programming. The results are clearly justified and the also the agreement with the other related work in the field. The study contributes towards strengthening that pair programming show impact over the quality of the code. Considering only defect density to identify that the pair programming is impacting the quality is wrong however we also need to consider how the coordination is working, how the information when needed is gathered by the navigator, overall structure of the code also need

to be looked into.

Compare with the Methods from literature.

In both the articles selected the main motive for quality improvisation using pair programming.

Similarities:

The defect density is important in both the case. In the CMMI level the both article belong to process model where the extreme programming is used is applicable in both the cases. Similarities include support for low level of defects. The important common difference within the article is that they both are supporting the defect as a metric for quality improvement which can help to support the articles contribution as their empirical results show that either improved or same no change in quality when pair programming is used. They use ISO standards to justify the metrics selected. Both follow ISO 9001:2008 standard (general assumption) from pattern followed.

Differences:

Article 1 spoke about both relative defect density and comment ration which are under defects in code. The article 2 only show results about the defect density. Article 1 conflicts article 2 in improved quality due to use of pair programming. Article 1 is conducted in university level and article 2 is conducted in industrial context this impact on complexity.

Strength and weakness:

Article1:

Strengths.

It is a case study more case studies are formed which add strength to the research area. Case study are performed in different scenario and are in detail and the result for individual scenario are also explained. The research questions are strong. They show significant impact in the results section by giving direct answer to the question posed. Varying context for the project is well described. Various other standards are described like not only quality but also the quality comment ratio.

Weakness:

Not clearly mentioned the development process followed in the process. Numerical values are not stated about how much percent the impact of pair programming over solo

programming. All the metrics are not considered in the four case studies that are made. Limitations are not well described. The author done an extensive previous article research. But the author did not well describe the result section. The compared projects are of different type and different effort estimations are made and no requirement change influence is not mentioned. This other facto influences the results.

Article 2:

Strength:

All the measures are taken to ensure only compared projects are taken into account. Very well detailed description on how the result and the case study is carried out. There is also survey which support the; research. Very well described the scope of the research. The results for all the outcomes are well stated with complete numerical statistics on the variation between the pair programming and solo programming.

Weakness:

Limitations and threats to validity are not well described. The Text is complex to understand sing diagrams to explain the results would have been nice. Scope of the project is narrow it is only applicable to university level task. Industry level complex task might have a varied result. No good justification is done for choosing the metric for quality measurement.

References

- [1] H. Hulkko and P. Abrahamsson, "A Multiple Case Study on the Impact of Pair Programming on Product Quality," in *Proceedings of the 27th International Conference on Software Engineering*, New York, NY, USA, 2005, pp. 495–504.
- [2] T. Bipp, A. Lepper, and D. Schmedding, "Pair programming in software development teams – An empirical study of its benefits," *Inf. Softw. Technol.*, vol. 50, no. 3, pp. 231–240, Feb. 2008.