# How Can Game Developers Use Pair Programming To Improve Software Quality?

**COMP150 - Agile Essay Proposal Assignment**

Samantha Wills

February 25, 2016

## Introduction

In the games industry, the quality of the software can be effected and interpreted in different ways. For many, it is the functional cohesion of the software which constitutes as high quality while others include code comprehension and commenting. During the process of programming, there can be a variety of factors influencing the programmers ability to code and can often change the speed and quality. Therefore, in many agile practices, pair programming is used to reduce common mistakes and focus code direction. This essay aims to investigate the ways pair programming can be used and has proven to improve standards of work. It also aims to investigate the effect of team strength and relationship of the programming pair in their attempt to improve software quality.

## Paper 1

**Title:** Investigating the Effective Implementation of Pair Programming: An Empirical Investigation

**Citation:** [1]

**Abstract:** "Pair programming is a programming technique where two programmers work together on the same programming task. Previous research has shown that it is effective for improving the learning effectiveness, efficiency, and enjoyment of students in introductory programming courses. Much research has also been dedicated to determining effective strategies for forming pairs. This paper discuss two different empirical studies conducted at North Dakota State University to a) test the feasibility of using pair programming in introductory computer science courses and b) determine whether or not major-based pairing produces effective pairs. The results of these studies provide support for implementing pair programming in introductory computer science courses and show that pairing of computer science and non-computer science students may produce pairs which are less compatible than other pairing methods."

**Web link:** `http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=1953346&CFID=586259388&CFTOKEN=83800710`

**Full text link:** `http://delivery.acm.org.ezproxy.falmouth.ac.uk/10.1145/1960000/1953346/p655-radermacher.pdf?ip=193.61.64.8&id=1953346&acc=ACTIVE%20SERVICE&key=223837E73163AEDA.EAA225A8AB01C582.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=586259388&CFTOKEN=83800710&__acm__=1456422682_bce244ad7b80ef506bbd9073ce20b79d`

**Comments:** This is a paper that encourages the use of pair programming to produce higher quality software in computer science course. This idea is then applied to game developers within studios.

## Paper 2

**Title:** Why developers don't pair more often

**Citation:** [2]

**Abstract:** "Applying pair programming in industrial settings can be challenging. This study extends the existing knowledge on applying pair programming in industrial settings by interviewing 31 developers from 4 different companies. We investigate how often developers practice pair programming as opposed to working by themselves, whether developers would prefer to use PP more often or less often, and which aspects hinder the use of pairing. We found that the majority of developers use PP only for 10

**Web link:** `http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=2663664&CFID=586259388&CFTOKEN=83800710`

**Full text link:** `http://delivery.acm.org.ezproxy.falmouth.ac.uk/10.1145/2670000/2663664/p123-plonka.pdf?ip=193.61.64.8&id=2663664&acc=ACTIVE%20SERVICE&key=223837E73163AEDA.EAA225A8AB01C582.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=586259388&CFTOKEN=83800710&__acm__=1456423115_d1fd0f2feac74e94e7a7c749b148165f`

**Comments:** This paper introduces interviews with a variety of developers and uses this to discuss the way game developers approach pair programming.

## Paper 3

**Title:** Conflict in collaborative software development

**Citation:** [3]

**Abstract:** "Pair Programming is an innovative collaborative software development methodology. Anecdotal and empirical evidence suggests that this agile development method produces better quality software in reduced time with higher levels of developer satisfaction. To date, little explanation has been offered as to why these improved performance outcomes occur. In this qualitative study, we focus on how individual differences, and specifically task conflict, impact results of the collaborative software development process and related outcomes. We illustrate that low to moderate levels of task conflict actually enhance performance, while high levels mitigate otherwise anticipated positive results."

**Web link:** `http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=761856&CFID=586259388&CFTOKEN=83800710`

**Full text link:** `http://delivery.acm.org.ezproxy.falmouth.ac.uk/10.1145/770000/761856/p44-domino.pdf?ip=193.61.64.8&id=761856&acc=ACTIVE%20SERVICE&key=223837E73163AEDA.EAA225A8AB01C582.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=586259388&CFTOKEN=83800710&__acm__=1456423292_aab70ee3b23103c23af12027317b43b4`

**Comments:** This paper discusses the individual differences of the programmers and how this can affect the effectiveness of the pair programming approach.

## Paper 4

**Title:** Video Analysis of Pair Programming

**Citation:** [4]

**Abstract:** "This article presents the results of a video analysis of nine pair programming sessions of undergraduate students. The analysis focuses on the keyboard and mouse control of the programming partners. It shows that most pairs do not share the keyboard and mouse equally but rather have one partner who is more active than the other. Keyboard and mouse control changes frequently, casting doubt on the existence of the driver and navigator role as commonly defined in literature on extreme programming."

**Web link:** `http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=1370151&CFID=586259388&CFTOKEN=83800710`

**Full text link:** `http://delivery.acm.org.ezproxy.falmouth.ac.uk/10.1145/1380000/1370151/p37-hofer.pdf?ip=193.61.64.8&id=1370151&acc=ACTIVE%20SERVICE&key=223837E73163AEDA.EAA225A8AB01C582.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=586259388&CFTOKEN=83800710&__acm__=1456423538_0ba82dec4b8d6fb20b64accc6a488d0f`

**Comments:** This paper discusses the interaction of the pair programmer with the input devices: mouse and keyboard. This is then discussed as to how it effects the driver-navigator dynamic.

## Paper 5

**Title:** Pair programming and agile software development: experiences in a college setting

**Citation:** [5]

**Abstract:** "While agile software development methodologies are now becoming more commonplace in industry, they have yet to be fully embraced by academic institutions. To better prepare university students for the marketplace, it is important that students are familiar with these alternative software development methods. For those unacquainted with agile methodologies, this paper first provides a summary of agile practices and a review of reports on teaching these practices in college classrooms. Next it discusses a study that the authors conducted in a graduate course entitled Software Development Process Models, in which the students were required to use eXtreme Programming practices. The discussion describes course assignments and the results from a survey that was administered to assess students' perceptions of their experiences using pair programming and agile methods."

**Web link:** http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=1181927&CFID=586259388&CFTOKEN=83800710

**Full text link:** http://delivery.acm.org.ezproxy.falmouth.ac.uk/10.1145/1190000/1181927/p145-sherrell.pdf?ip=193.61.64.8&id=1181927&acc=PUBLIC&key=223837E73163AEDA.EAA225A8AB01C582.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=586259388&CFTOKEN=83800710&__acm__=1456423748_c2be2186437c36588928a196dda59b76

**Comments:** This paper discusses the use of agile pair programming in academic institutions and perhaps how this could then lead on to more programmers having practised pair programming before starting work as a game developer.

## Paper 6

**Title:** A multiple case study on the impact of pair programming on product quality

**Citation:** [6]

**Abstract:** "Pair programming is a programming technique in which two programmers use one computer to work together on the same task. There is an ongoing debate over the value of pair programming in software development. The current body of knowledge in this area is scattered and unorganized. Review shows that most of the results have been obtained from experimental studies in university settings. Few, if any, empirical studies exist, where pair programming has been systematically under scrutiny in real software development projects. Thus, its proposed benefits remain currently without solid empirical evidence. This paper reports results from four software development projects where the impact of pair programming on software product quality was studied. Our empirical findings appear to offer contrasting results regarding some of the claimed benefits of pair programming. They indicate that pair programming may not necessarily provide as extensive quality benefits as suggested in literature, and on the other hand, does not result in consistently superior productivity when compared to solo programming."

**Web link:** `http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=1062545&CFID=586259388&CFTOKEN=83800710`

**Full text link:** `http://delivery.acm.org.ezproxy.falmouth.ac.uk/10.1145/1070000/1062545/p495-hulkko.pdf?ip=193.61.64.8&id=1062545&acc=ACTIVE%20SERVICE&key=223837E73163AEDA.EAA225A8AB01C582.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=586259388&CFTOKEN=83800710&__acm__=1456423956_3d7b5f1882cc9f024234f2e111ed2558`

**Comments:** This paper discusses the results from four software development projects and evaluated the claimed benefits of pair programming. It also debates the validity of using empirical studies against experimental studies in university settings.

## References

[1] A. D. Radermacher and G. S. Walia, "Investigating the effective implementation of pair programming: An empirical investigation," in *Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '11. New York, NY, USA: ACM, 2011, pp. 655–660. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/1953163.1953346

[2] L. Plonka and J. van der Linden, "Why developers don't pair more often," in *Proceedings of the 5th International Workshop on Co-operative and Human Aspects of Software Engineering*, ser. CHASE '12. Piscataway, NJ, USA: IEEE Press, 2012, pp. 123–125. [Online]. Available: http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=2663638.2663664

[3] M. A. Domino, R. W. Collins, A. R. Hevner, and C. F. Cohen, "Conflict in collaborative software development," in *Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia–leveraging Differences and Diversity in the IT Workforce*, ser. SIGMIS CPR '03. New York, NY, USA: ACM, 2003, pp. 44–51. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/761849.761856

[4] A. Höfer, "Video analysis of pair programming," in *Proceedings of the 2008 International Workshop on Scrutinizing Agile Practices or Shoot-out at the Agile Corral*, ser. APOS '08. New York, NY, USA: ACM, 2008, pp. 37–41. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/1370143.1370151

[5] L. B. Sherrell and J. J. Robertson, "Pair programming and agile software development: Experiences in a college setting," *J. Comput. Sci. Coll.*, vol. 22, no. 2,

pp. 145–153, Dec. 2006. [Online]. Available: http://dl.acm.org.ezproxy.falmouth.ac.uk/citation.cfm?id=1181901.1181927

[6] 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*, ser. ICSE ’05. New York, NY, USA: ACM, 2005, pp. 495–504. [Online]. Available: http://doi.acm.org.ezproxy.falmouth.ac.uk/10.1145/1062455.1062545