Innovative Systems Design and Engineering
ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online)
Vol 2, No 3

Partial Pair Programming: Link between Solo and Pair Programming

Bashir Ahmad
Institute of Computing and Information Technology
Gomal University, PAKISTAN
bashahmad@gmail.com

Sheikh Muhammad Saqib
Institute of Computing and Information Technology
Gomal University, PAKISTAN
saqibsheikh4@hotmail.com

Shakeel Ahmad
Institute of Computing and Information Technology
Gomal University, PAKISTAN
Shakeel_1965@yahoo.com

Muhammad Zubair Asghar
Institute of Computing and Information Technology
Gomal University, PAKISTAN

Abstract
Leaving of key team members can make an unrecoverable loss to an organization. It means knowledge of a task should be in mind of two or more than two programmers. Working of two people on same task on different machine is an awkward practice. So alternatively pair programming practice is the best solution for above problem. Besides many advantages of pair programming, it has certain drawbacks such as personality clashes, and these issues may dominant pair programming over solo programming. Here authors suggested a practice 'Partial Pair Programming', which will work as bridge between solo and pair programming practices. In partial pair programming, three drivers can make a pair with a navigator of their own group and navigator can make a pair with navigator of other group. This practice will get all advantages of pair programming as well as its own benefits. And also partial pair programming will remove almost all demerits of pair programming.

Keywords: Programmers, Pair Programming, Solo Programming, Partial Pair Programming, Driver and, Navigators;

1. Introduction

Human does really focus on work for some minutes and then take a break. This is actually habit of human. In development environment, people have a hard time pacing themselves. Pair programming is the best practice for solution of such issue where two people works on same code on same time alternatively [7].

Previously occurring problems in code of the developed application can be removed with frequent use of pair programming because in pair programming, program code is continuously revised. Actually major responsibility of monitoring the pair programming is falls onto the XP Project Manager, so XP-Project Manager should be very experienced in people skills: forming a team, overcoming resistance, coaching, solving conflicts. This is possible only in rare circumstances. Here it is observed that how a project manager can deal with the obstacles [8] [4].

Tanja Bipp, in his paper has explored the many advantages of pair programming with teams of eight students. They found higher quality of code with fewer complexities, better to read and easier to understand the code. It is observed that error debugging rate was fast. All pairs have done their work very positively but some team members have given negative comment. Two members had refused to work with a partner. So it can be realized that, pair programming is not acceptable for all members [9].

In pair programming, when two programmers, driver and navigator work together on same workstation then different flaws are occurred such as personality clashes, missing comfort between developers, parallel consumption of time on writing & reviewing code and etc. These flaws are created due to association of two people, when one types code while the other reviews each line of code. Different types of mismatching can be occurred between them. It does not mean that solo programming should be prominent over pair programming i.e. due to pair programming, development organization are very relaxed about the issue of leaving of any key team members. Now there should be a perfect mechanism which can follow all advantages of pair programming and deny its drawbacks. Here author proposed a Partial Pair Programming practice for achieving the good points of pair programming and denying their defects. Partial pair programming practice will consist of different groups and each group will consist of three coders and one reviewer. There should be a law of development organization that each driver will be responsible to obey relevant navigator. This law can be easily implemented when reviewers are experienced developers and coders are junior developers.

2. Pair Programming

Extreme Programming (XP) is an agile software development method and concluded of different practices for effective software development processes over time. Such practices are small releases, metaphor, simple design, testing, continuous integration, refactoring and pair programming [7]. In pair programming practice there are two programmers work together on same machine. Here one types the code and on same time other reviews it. Coder is called driver and reviewers is called navigator or observer. Both programmers can switch over their roles. In pair, one is experienced developers and other is junior developers and they can share their knowledge such as technical, application domain, design and application source code knowledge. If a critical member of the team leaves, such type of shared knowledge reduces the damage. In pair programming, all developers can have a better understanding of the whole code. The official definition of pair programming is that “working of two programmers side by side, at one computer collaborating on the same, analysis, design, implementation and test” [5] [1].

3. Partial Pair Programming

In Partial Pair Programming, different groups will be made by the project manager and each senior programmer will be associated with three junior developers. In each group, experienced programmer will be known as Navigator and three junior programmers will be known as drivers. Number of groups should be equal to the total number of experienced programmers (Navigator).

Major theme of Partial Pair Programming is shown in Figure-1.

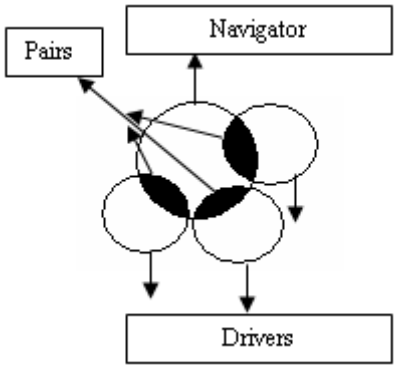

Figure-1: Structure of Partial Pair Programming of Single Group.

In above figure, black area relates to pair, it means each Driver has its pair with one Navigator, who will guide them in coding, debugging and refining.

3.1 Working of Partial Pair Programming

Pair programming is the practice of XP model. In such model story cards are the main pillar for gathering the requirements. Without knowledge of development task underlying the experience would be meaningless [2]. In general pair programming, both developers must have complete understanding of story cards i.e. what thing user of client side wants. In pair programming, pair generates the code up to testing level as shown in following figure.

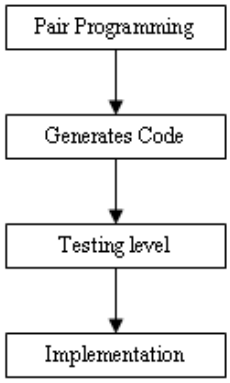

Figure-2: Working Steps of General Pair Programming

In general pair programming, work of driver and navigator is parallel, both work on same machine on same time i.e. one writes code and other review it. While in partial pair programming, navigator will responsible for understanding the story cards from all aspects, as they will have involved in gathering the requirements from scratch. It is not necessary for drivers that they are involved in gathering requirements. According to law, they will responsible to obey their navigator. Drivers will do all those things which are given by their navigator. They will not responsible for achieving what is going to be done, navigator will conclude this thing from the work of drivers and make their dimensions/directions on right side all the time, while drivers will be responsible for only achieving how to done the code suggested by navigator. When drivers will reach on end of code, then they can automatically understand what was the actual propose of developed activity, which they have done. In partial pair programming, each driver will generate the code up to navigator level. Navigator will communicate the developed work with project manager.

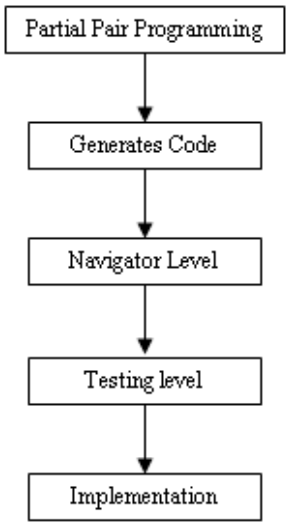

Figure-3: Working Steps of Partial Pair Programming.

In Partial Pair Programming, work of driver and navigator is in collinear way; navigator will guide them on time of problem-line and development approach will be parallel i.e. three drivers will work on different activities of same increment/module.

3.2. Driver-Navigator Pair

Each project's module will be dedicated to each owner/navigator of group. Owner will divide the activities in three drivers. Navigator will announce the story cards to each driver up to coding level and will suggest the type of scheme, format of code, setting of Active-X controls and etc. Drivers will done the work up to coding level. In the mean time they can get the different suggestions from their navigator and navigator will perform the supervision from start to end of activities.

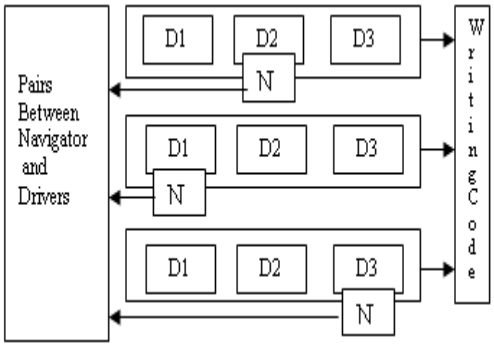

Figure-4: Working of Drivers-Navigator Pairs in Same Group.

When any driver feels some difficulties or problems, he can call his navigator, who will guide him in code and understanding the thing what code he is going to be done.

Figure-4 consists of three steps. In step 1st, Driver-2 is making a pair with navigator when he gets some problem while Driver-3 and Driver-2 are busy in their own work.

In step 2nd, Driver-1 is making a pair with navigator when he gets some problem while Driver-2 and Driver-3 are busy in their own work.

In step 3rd, Driver-3 is making a pair with navigator when he gets some problem while Driver-1 and Driver-2 are busy in their own work.

When navigator will busy with a pair, then other drivers without pair will do their own task. If all drivers will not have any problem then navigator will make a gaze eye over all drivers' code and also can do some other relevant work of project. A situation can occur, when two drivers have a problem on same time, then navigator will guide one driver and for other drivers, navigator will make a pair with the navigator of other group, as explained below.

3.3. Navigator-Navigator Pair

Working of navigator of one group with the driver of other group is known as Navigator-Navigator pair. This pair will be produced on that time when a navigator will busy with one driver and on same time another driver gets some problems. So any driver of a group can call to navigator of another group, who will free. It means that called-navigator will do the work of navigator of other group.

When any driver of a group calls to navigator of another group who will free, it means that called navigator is doing the work of another navigator, thus logically they have made a pair.

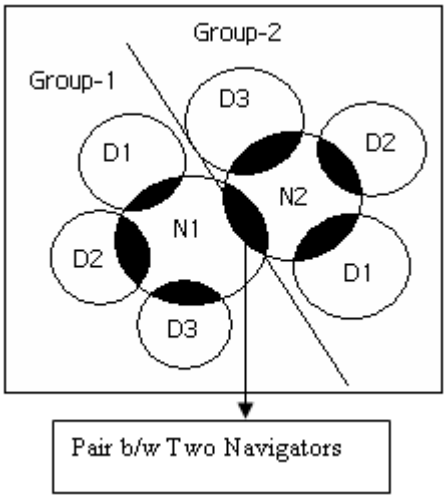

Figure-5: Working of Navigator-Navigator Pairs.

4. Results and Conclusion

In partial pair programming, three driver can make pair partially with single navigator i.e. owner of their group. Drivers can communicate the work with navigator and navigator can communicate the work with project manager.

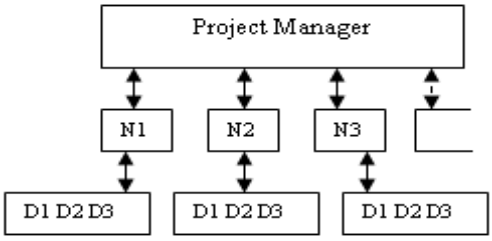

Figure-6: Structure of Partial Pair Programming of All Group.

Partial Pair Programming will get all advantages of general pair programming as follows.

Better code: Overall development time will be reduced and quality of code will be higher.

Increase discipline: Drivers-Navigator will more likely to do right thing.

Pairing Flow: This will happen more quickly because any driver can ask the navigator, "What were we working on?

Improve morale: Pairing work will be enjoyed able as compare to .solo programming.

Collective code ownership: Both driver and navigator will gain a working knowledge of the entire overall code.

New Member: New members of the team can get complete knowledge in speedy way.

Reducing Bugs: Especially bugs will be reduced in code very fast.

Besides all above advantages, partial pair programming has its own benefits. Following table is showing, what number of pair can be created in pair programming and partial pair programming.

Table-1: General Parameters for Pair Programming (PP) and Partial Pair Programming (PPP)

Parameters	PP	PPP
Total Person	m	m
Senior Programmer (S)	m/2	S
Junior Programmer (J)	m/2	S*3
Association	1 to 1	1 to 3
Total Pair	m/2	m-S
Cost of S (Per Month)	X	X
Cost of J (Per Month)	Y	Y
Total Cost of S/Month (TS)	X*m/2	X*S
Total Cost of J/Month (TJ)	Y*m/2	Y*J
Total Cost of /Month	TS+TJ	TS+TJ

Now we assume that if total person are 20, than what number of pairs can be made in pair programming and partial pair programming. Also we will calculate the monthly cost with both programming practices. From discussion and observation with people of working in software houses that monthly pay of junior programmer is almost 40% to 50 % less as compare to senior programmers i.e. if senior programmer takes 100% pay per month then the pay of junior programmer will be 70%.

Table-2: General Total Pairs and Cost in PP and PPP.

Parameters	PP	PPP
Total Person	20	20
Senior Programmer (S)	10	5
Junior Programmer (J)	10	15
Total Pair	10	15
Cost of S (Per Month)	100	100
Cost of J (Per Month)	70	70
Total Cost of S/Month (TS)	1000	500
Total Cost of J/Month (TJ)	700	1050
Total Cost of /Month	1700	1550

It is concluded from above table that when m=20, then in pair programming total pairs are 10 & cost are 1700 and in partial pair programming total pairs are 15 and cost are 1550. Now

%age of Pairs in PP= (10/20) * 100 = 50%.

%age of Pairs in PPP= (15/20) * 100 = 75%.

%age of Per Month in PP= (1700/1700) * 100 = 100%.

%age of Per Month in PPP= (1550/1700) * 100 = 91%.

Hence it is proved that with partial pair programming 25% (75-50=25) pair will be more as compare to partial programming and 9% (100-91=9) cost can be reduced with partial pair programming.

Now almost drawbacks of general pair programming [3] will be removed by adopting the practice of partial pair programming. The major problem to pair programming is that it can push people well outside their comfort zones due to reason that "if they make me pair" while in partial programming, three drivers will be connected with experienced programmers. They can not think about the matter one to one pair.

In pair programming, after a day of pairing, the team is usually exhausted but with partial pair programming, each programmer work just like a solo programming practice, they make pair on time of problem.

In pair programming, one to one all time work can produce personality clash while in partial pair programming one to one work will be done in some intervals.

Reviewing code takes less time as compare to writing code, it means navigator can review the code of three drivers parallel. So in partial pair programming it can be done while in pair programming each driver is associated with one navigator and they can change their role.

In solo programming, when any key member leaves the organization then organization may not recover the work of that member. This drawback can overlap all advantages of solo programming. So development organization adopts pair programming practice for removing such flaw. While due to different flaws of pair programming, there must be an intermediate approach between solo and pair programming. Those people who avoided the pair programming practice, they can easily espouse Partial Pair Programming practice.

References

A Pair Programming Experience, http://accu.org/index.php/journals/254

Randall W Jensen, "A Pair Programming Experience";http://accu.org/index.php/journals/254, 2005

Pair Programming Benefit"s, http://c2.com/cgi/wiki?PairProgrammingBenefits

Dr. Kamran Sharifabdi, "Team Development and Pair Programming – tasks and challenges of the XP coach".

"Pair Programming- Software Development Learning steps", http://www.ianburgess.me.uk/en/project-management/pair-programming-software-development-learning-steps

Guillermo A. Callahan, "Suitability of Extreme Programming and RUP Software Development Methodologies for SOA Applications", 2006

Mendelt Siebenga, "The dirty secret of pair programming", 2009

Rostislav Fojtik, "Extreme Programming in development of specific software", Procedia Computer Science 3 (2011) 1464–1468, ELSEVIER

Tanja Bipp, " Pair programming in software development teams – An empirical study of its benefits", Information and Software Technology 50 (2008) 231–240, ELSEVIER