

A Summary from a Pair Programming Survey

Subpart from an undergraduate master thesis paper

Increasing Quality with Pair Programming

**- An Investigation of Using Pair Programming
as a Quality Tool**

Kim Nilsson



Supervisor: Professor Claes Wohlin.
Department of Software Engineering and Computer Science.
Blekinge Institute of Technology.

PP Survey Summary

“Pair programming really deserves its own book. It is a subtle skill, one that you can spend the rest of your life getting good at”.

Kent Beck¹

1 Introduction

This is a summary from a survey about Pair Programming. The survey took place during 21st November to 19th December in 2002. Both professionals as well as students with and without Pair Programming experience have been queried. The result is shown as graphs for each question.

1.1 Survey background

Today there are few pair programmers in Sweden so the survey have been made to look at how people in general feel about the benefits from the use of Pair Programming (PP). It has addressed both experienced pair programmers as well as professionals without any [or low] experience from PP. We have also surveyed students, which will be the next generation of developers in a near future to see if they differ from the professionals.

Survey purpose

As well as in Sweden as worldwide isn't the use of PP until today that widely used. Even if research indicates that people who have tried it is satisfied and have confidence in the technique we don't know how developers in Sweden think of it. Therefore is the main purpose to see how confident people here might be with PP. Since the use of PP is quite new here in Sweden we also urge that our survey should cover the confidence developers have in PP, which have not used it yet. How, if and where do developers see the benefits from PP? In which ways might PP be better to use than normal single development? Is there much resistance to start use PP, and is there resistance even for those who used it already? If

¹ Kent Beck, “Extreme Programming Explained, Embracing Change”, Addison-Wesley, UK, 1999.

people don't feel confident in the technique or have great resistance using it then nothing else matters, you can never expect get something good (a success) out of something people don't like to do.

Survey size

The goal has been to query 50 professionals and 50 students, where 25 should have PP experience and 25 not for the professionals. We have calculated with a 2/3 in answering frequency. For the students we hoped for a natural division between them since we knew some of them have already had experience from it.

Therefore is the survey results divided into the four groups as follows: Professionals with PP experience, professionals without, students with PP experience and finally students without.

Survey response facts

In total 104 person have been queried and 67 have answered. There have been 55 students queried though direct-addressed e-mail and here we had 37 responses. This gives us an answering frequency of 67 % for the students.

The answering frequency is a little more vague for the professionals since we haven't had direct contact with them all, especially not the pair programmers. There where 49 professionals asked in total and from them we had 30 responded, this gives us an answering frequency of 61 % (we have the same frequency when they are divided into PP experience).

There where 80 % male and 20 % female among the professionals and for the students the figures where 92 % males compared to 8 % females.

The average as well as median age for the professionals was 33 years for the students the average was 27 and the median was 24.

1.2 How to read

All the graphs have a separation between professionals and students. Except for the two first graphs they have also been divided between people with and without Pair Programming experience. There is also a summary-group where "all" the responds have been put together.

All the graphs (except for the three first ones) are shown with the answers in percentage per category this eliminate the difference in number of people in each group. One answer equals about 5 % in each category (9 % for pro's with PP experience). The two first graphs are shown as the number of respondents instead.

The graphs 18-25 are shown as scales where a higher value is shown as a higher cone. The summary with all responds has been made as cylinders and are doubled in height to be easier to read.

Abbreviation in graphs

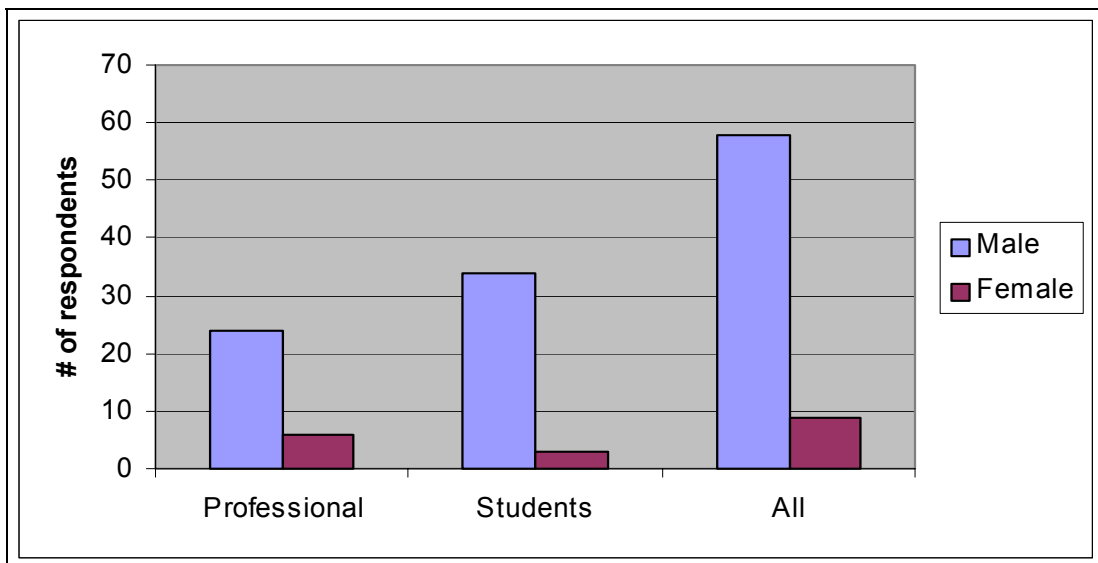
We have used the following abbreviations for the graphs.

- Pro PP: Professionals with Pair Programming experience (PP for at least 1-2 months).
- Pro Non: Professionals without PP experience (also include < week PP experience).
- Stud PP: Students with PP experience (PP for at least 1-2 months).
- Stud Non: Students without PP experience (also include < week PP experience).

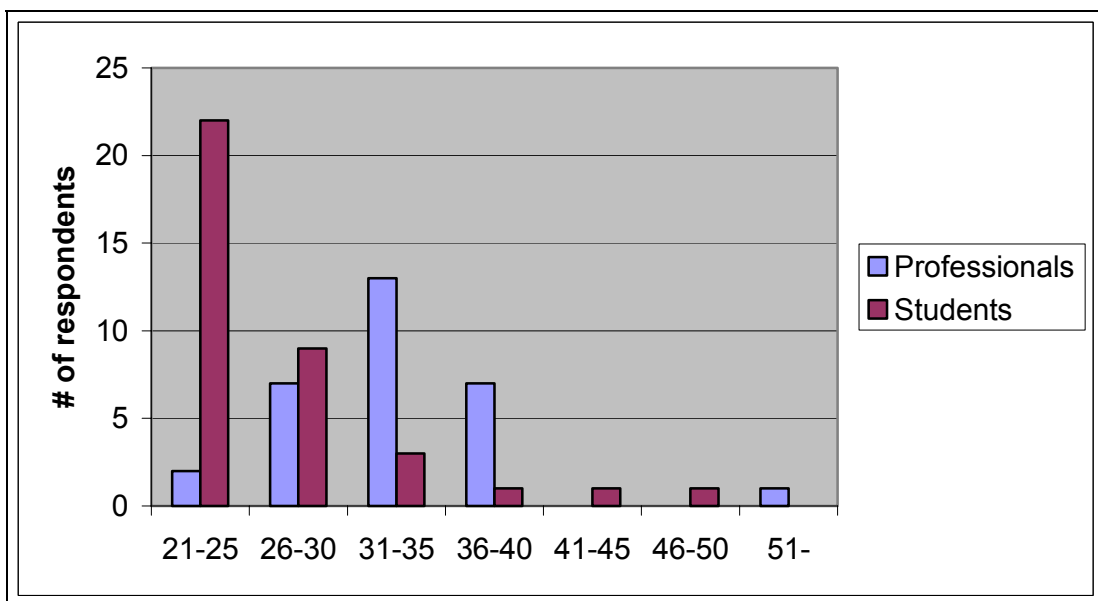
2 Survey response results

2.1 Part 1 (Affiliation, 7 questions).

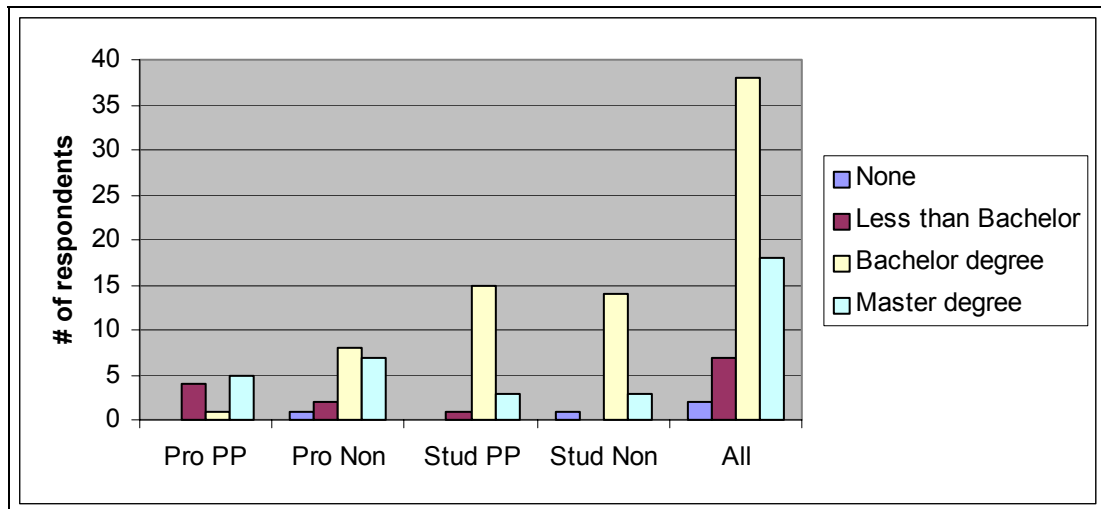
1. Your gender?



2. Year of birth (transformed into age-groups)?



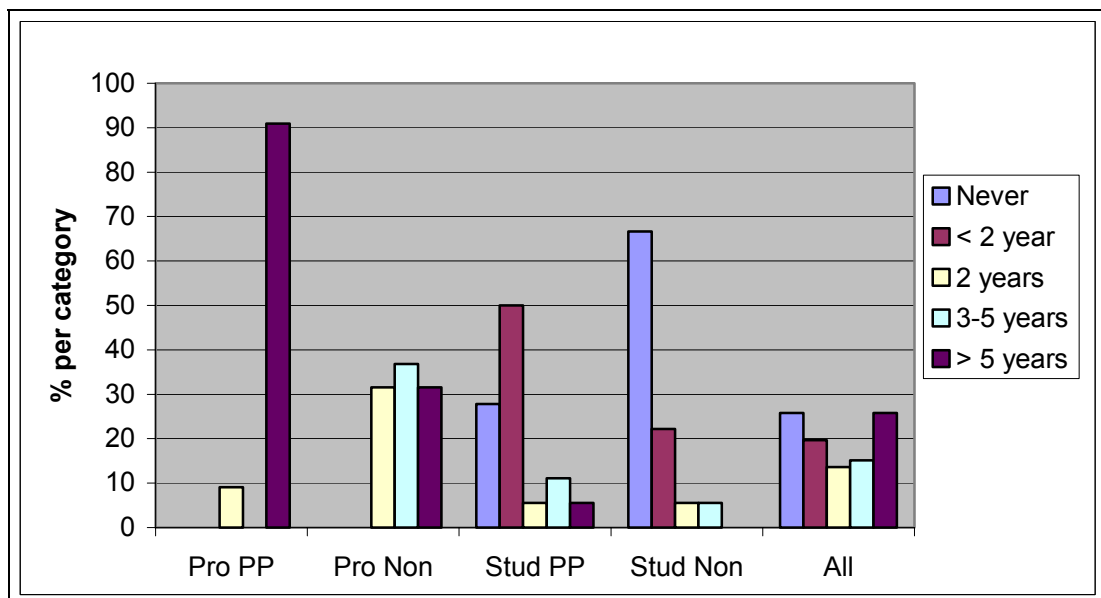
3. Your academic status (the closest)?



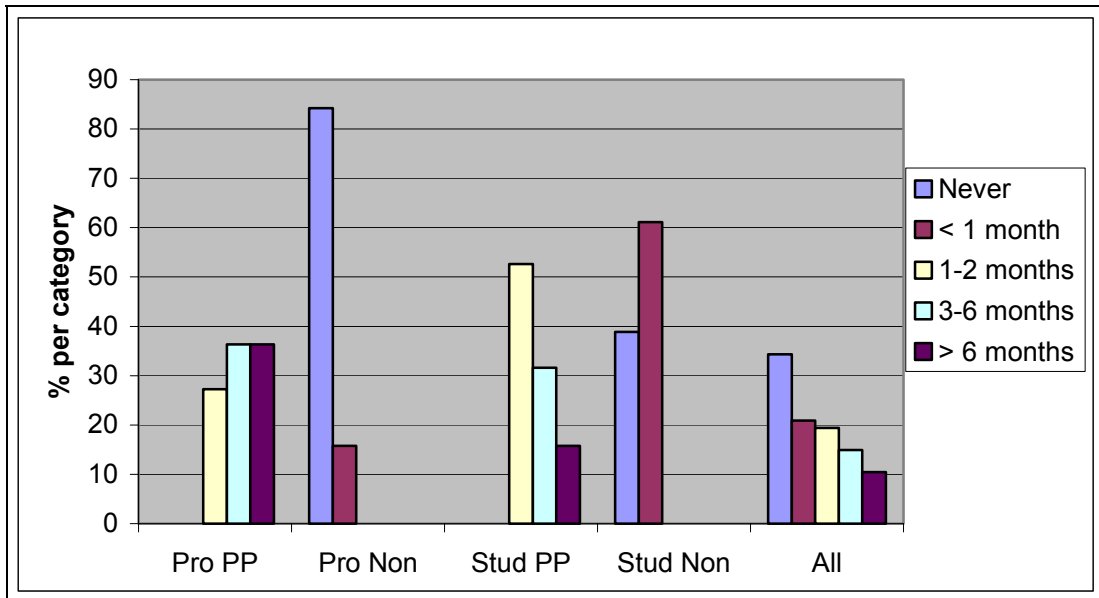
4. Are you working with software development?

No graph needed, used only to separate professionals and students.

5. For how long have you've been working professionally as a software developer in total?

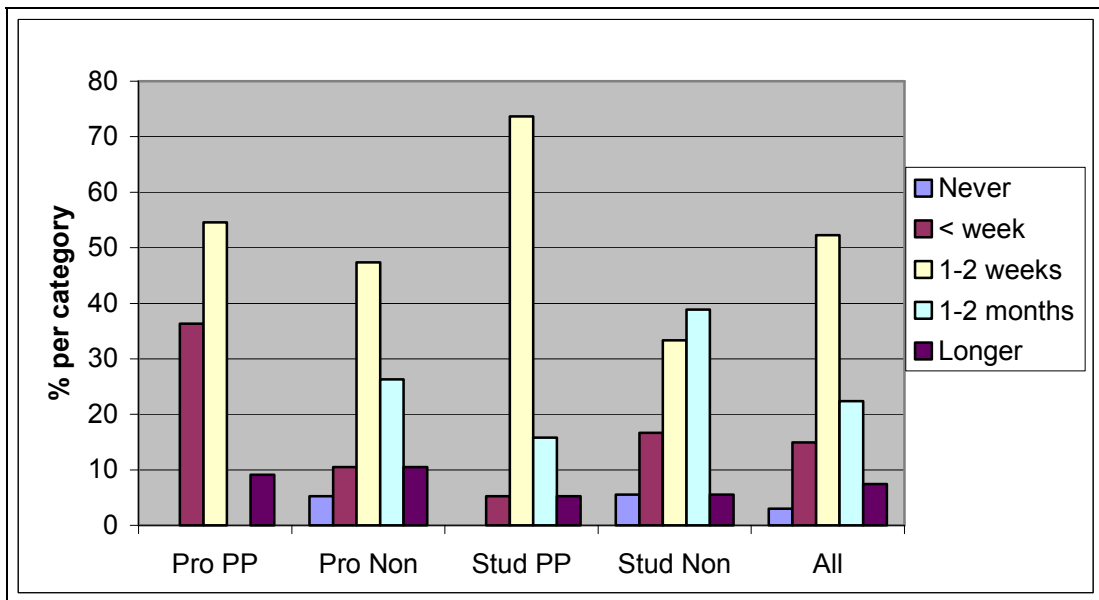


6. How long have you been used Pair Programming, or worked for a longer time with a pairing colleague on a single task on a computer in total?



Only used to separate people with and without experience, as seen in the graph, but PP experience can be seen.

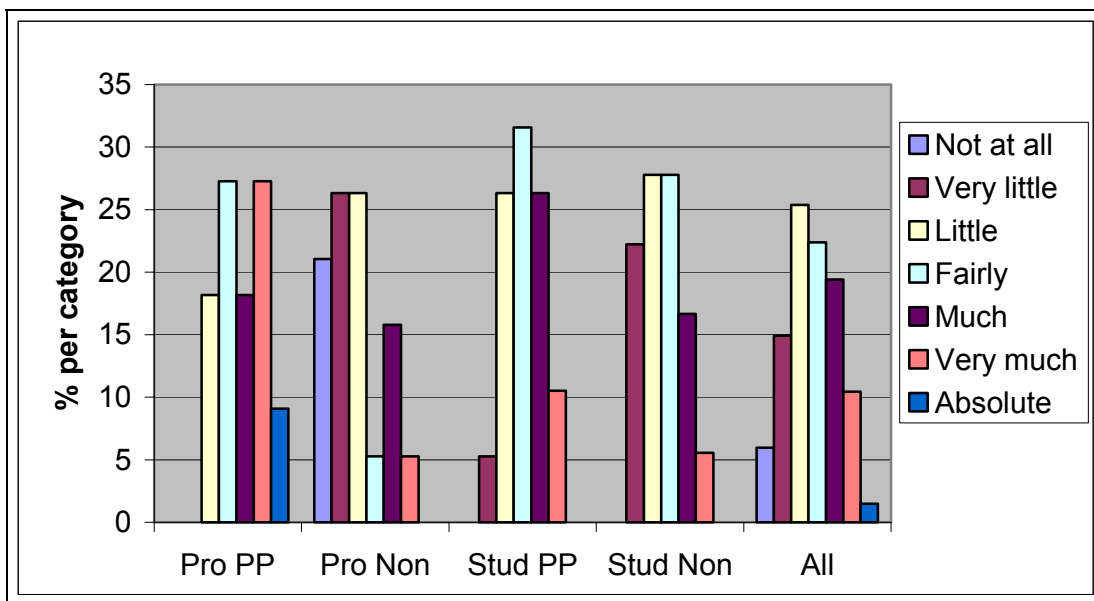
7. How long do you believe it will take in average before two persons work efficiently as a joined pair?



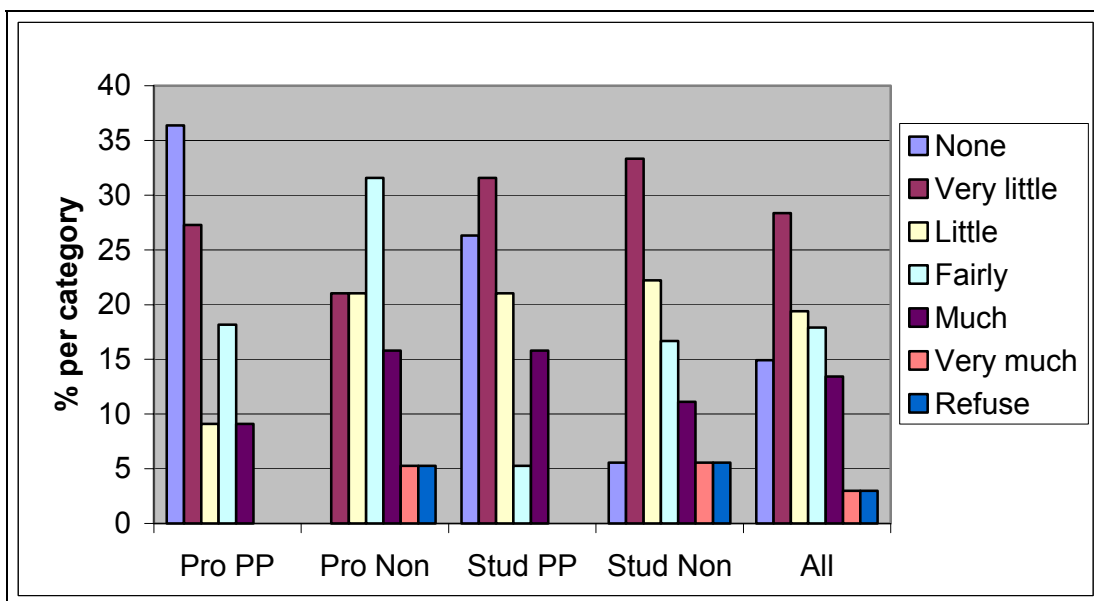
2.2 Part 2 (Percentage, 10 questions).

The 7 answer-alternatives were accompanied with a percentage value, e.g. Not at all 0%, Very little 10% etc. The percentage values were 0%, 10%, 30%, 50%, 70%, 90%, 100%

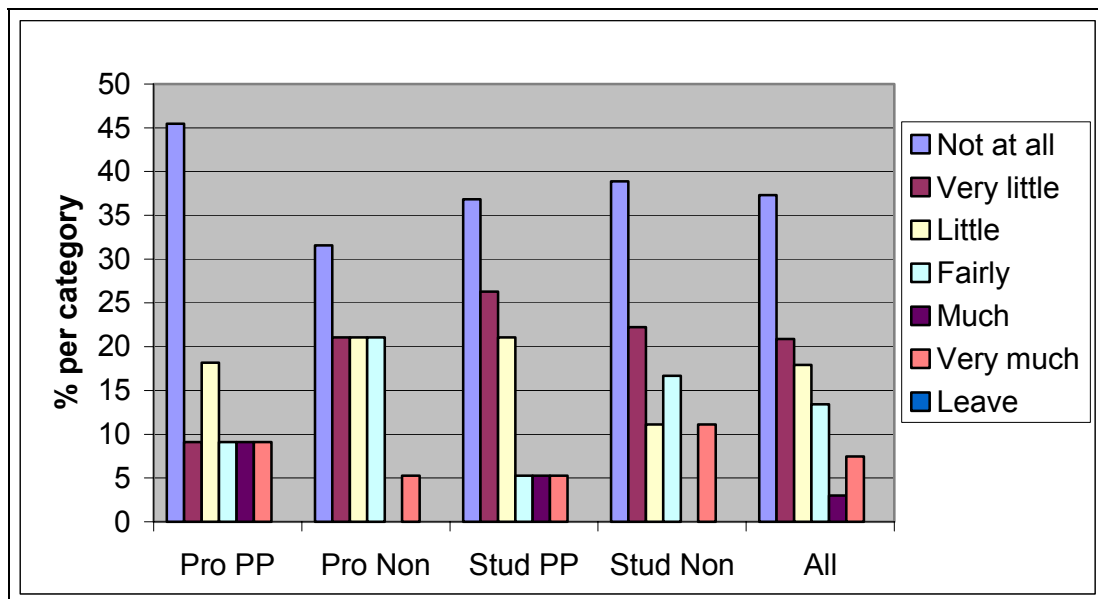
8. According to your own opinion, how familiar are you with the concept of Pair Programming?



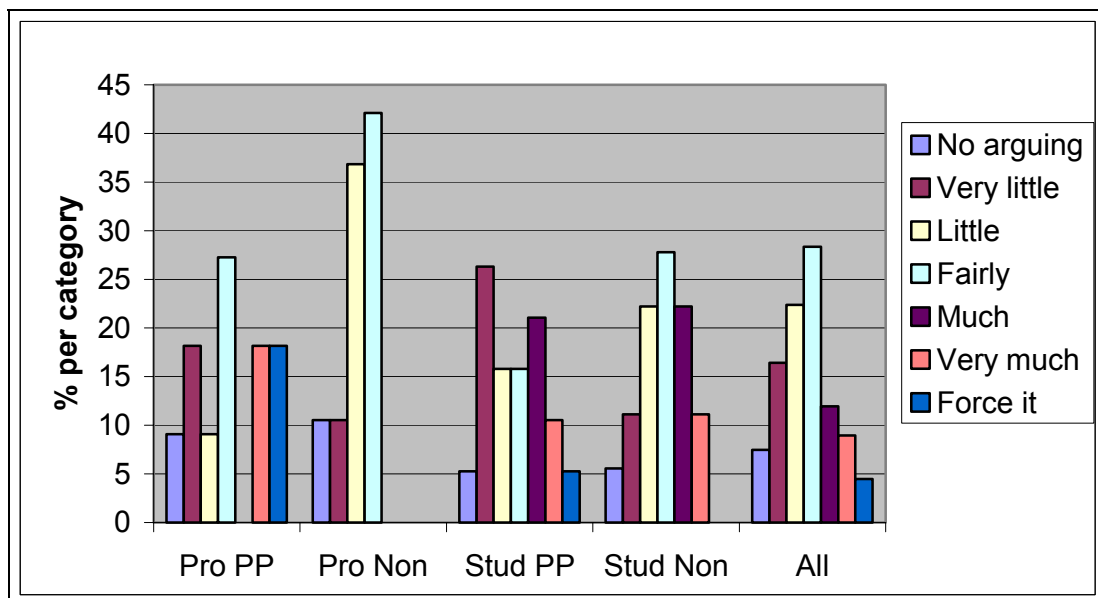
9. How much resistance do you personally have to change to Pair Programming, or did you have before first time?



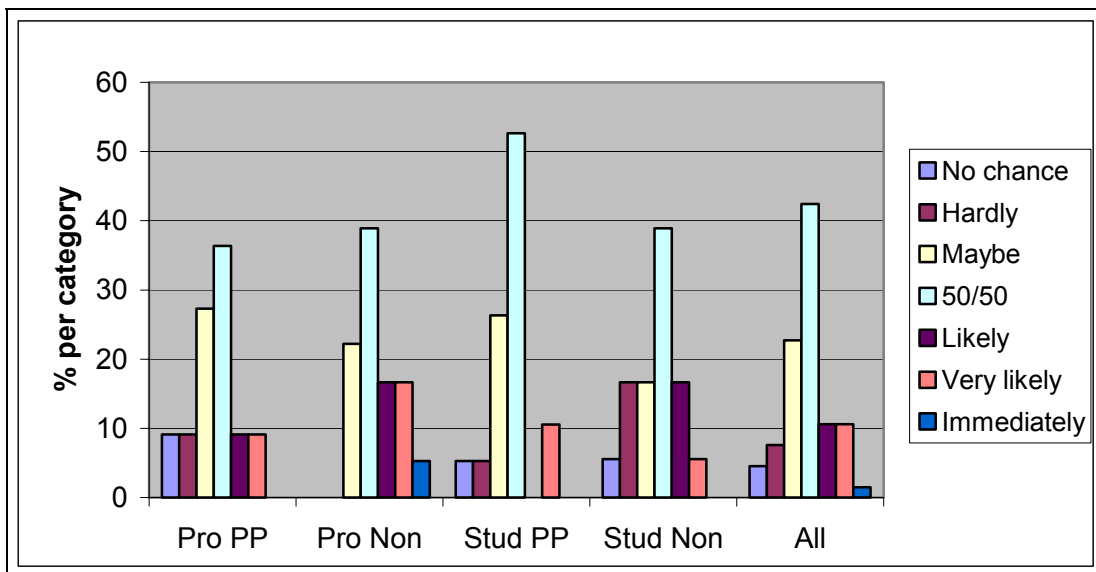
10. If your organization changed and said that all implementations should be done by pair programmers, to what degree would you refuse using it or even leave for another job?



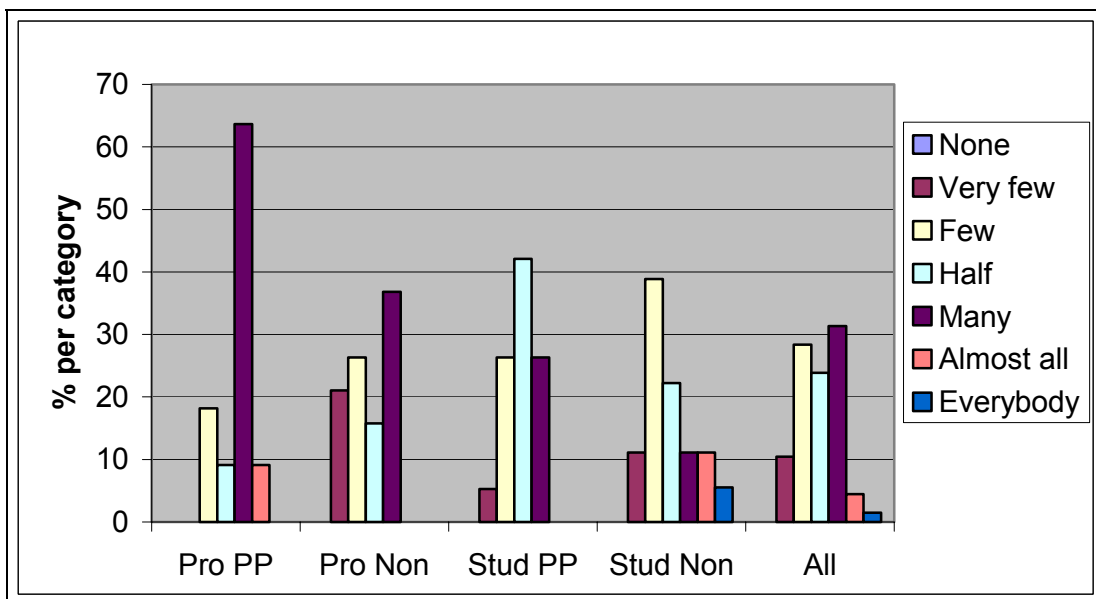
11. If colleagues from other places reported on successfulness from using Pair Programming, how likely you think that *you* personally (with your colleagues) would argue for start using it?



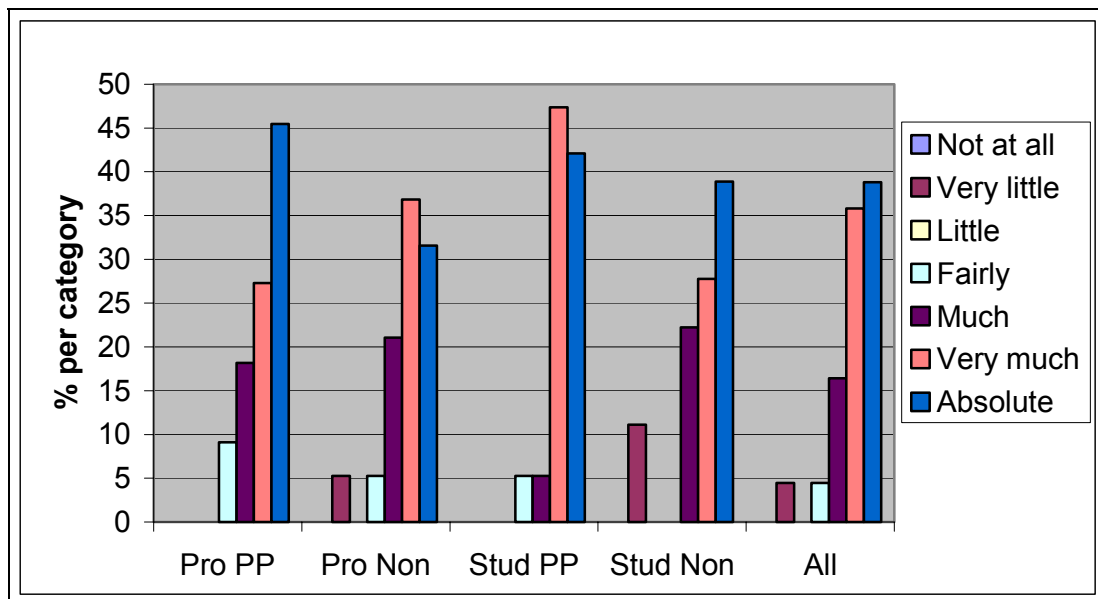
12. How likely do you believe it is that you would like to go back to normal development after experiencing Pair Programming?



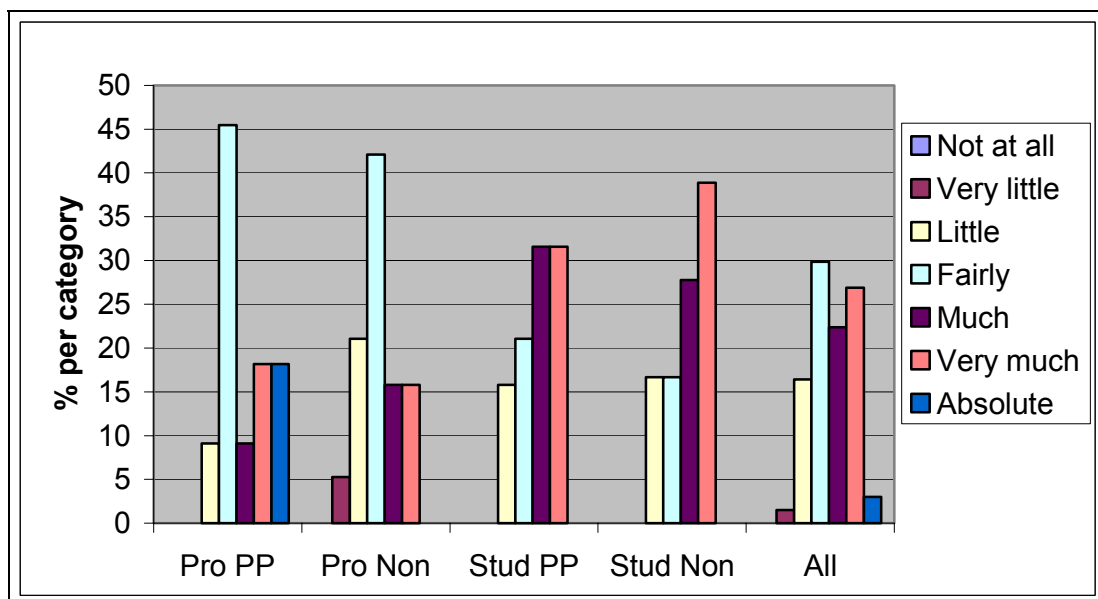
13. According to your opinion how many developers in an average organization do you believe will fit to be working as pair programmers and not only alone?



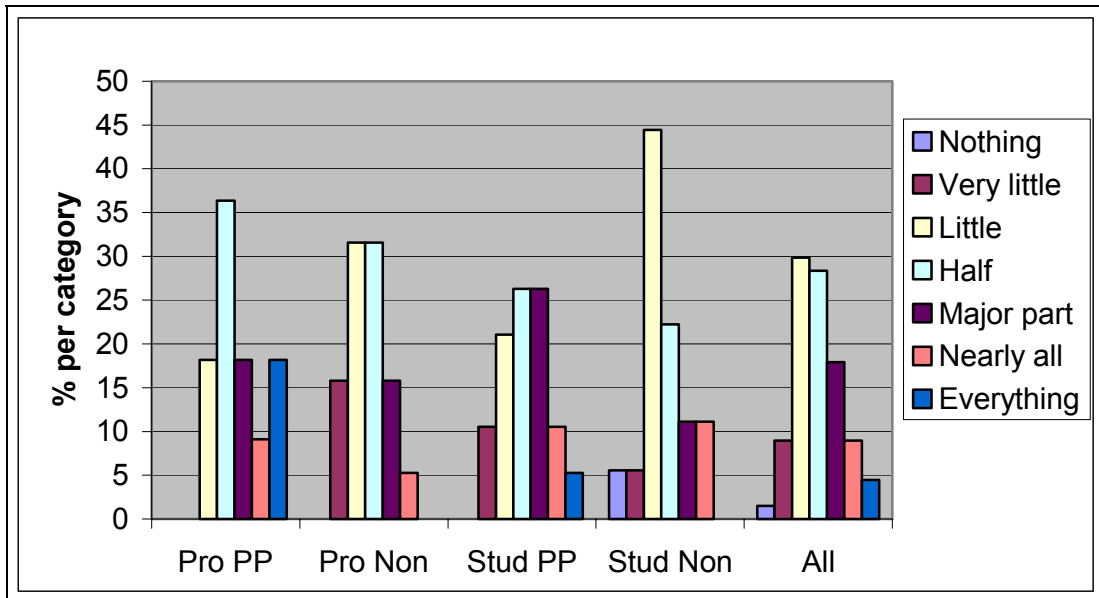
14. How important do you feel it is that a defect is found early, in any moment, compared with found but at a later moment?



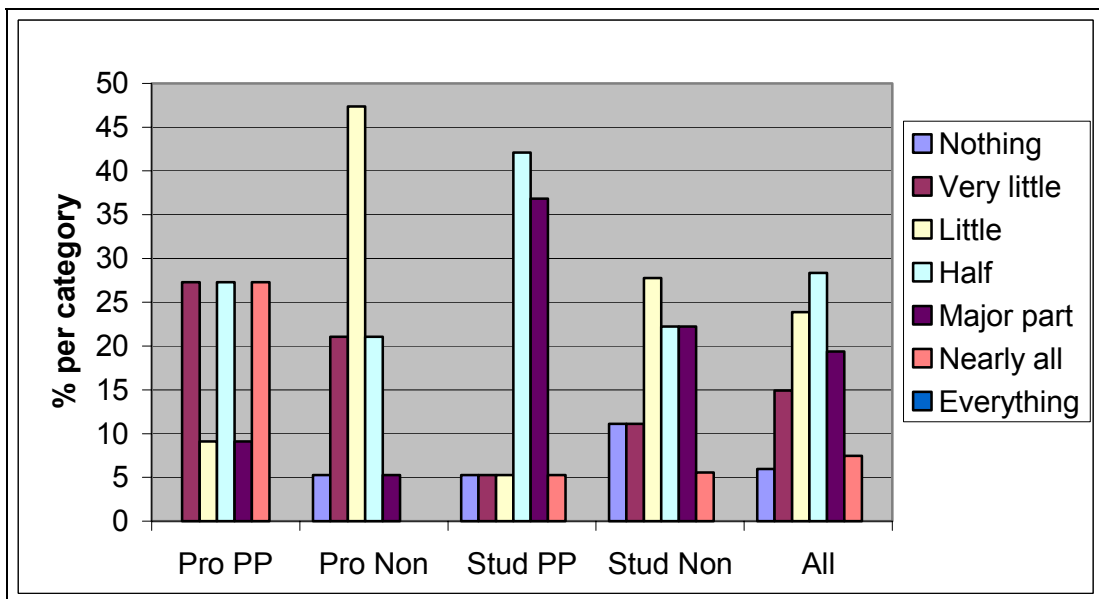
15. How efficient do you believe that Pair Programming can be to find/prevent early defects producing documents, code etc.?



16. How much of the code produced should be made by pairs in normal projects, according to you?



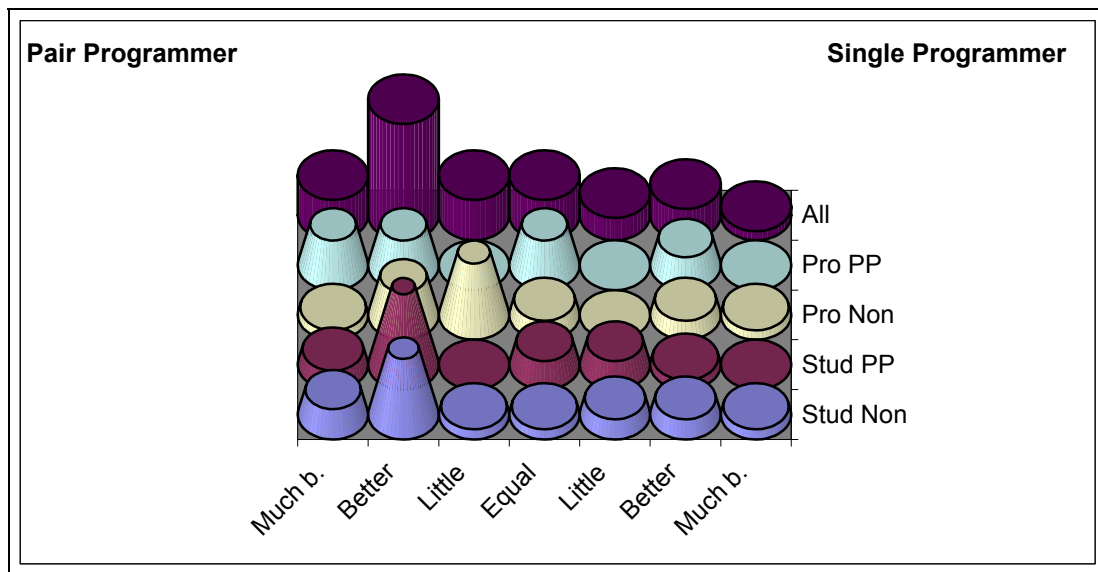
17. How much of the text in the documents should be written by pairs in normal projects?



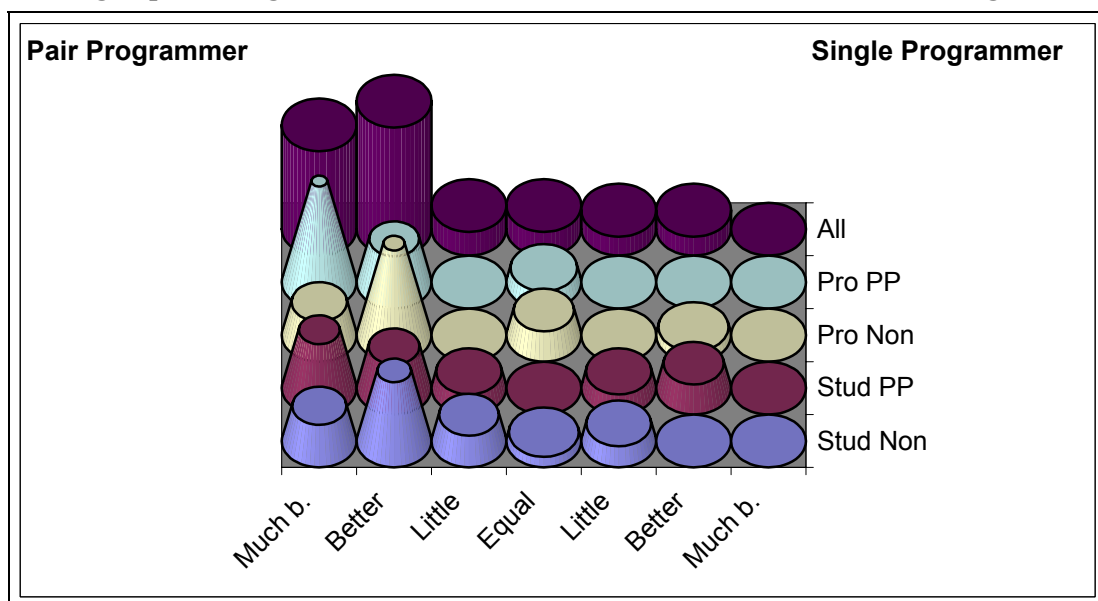
2.3 Part 3 (Compare, 8 questions).

Here we also had 7 answer-alternatives which were accompanied with percentage values, too, i.e. 90/10, 75/25, 60/40, 50/50, 40/60, 25/75, 10/90. They were made as a comparing scale between Pair Programming and Single programming (SP) where one side increased in value while the other option decreased. 50/50 were the middle “Equal” option and had the same other alternatives on each side. The height of the cones represent “% per category”.

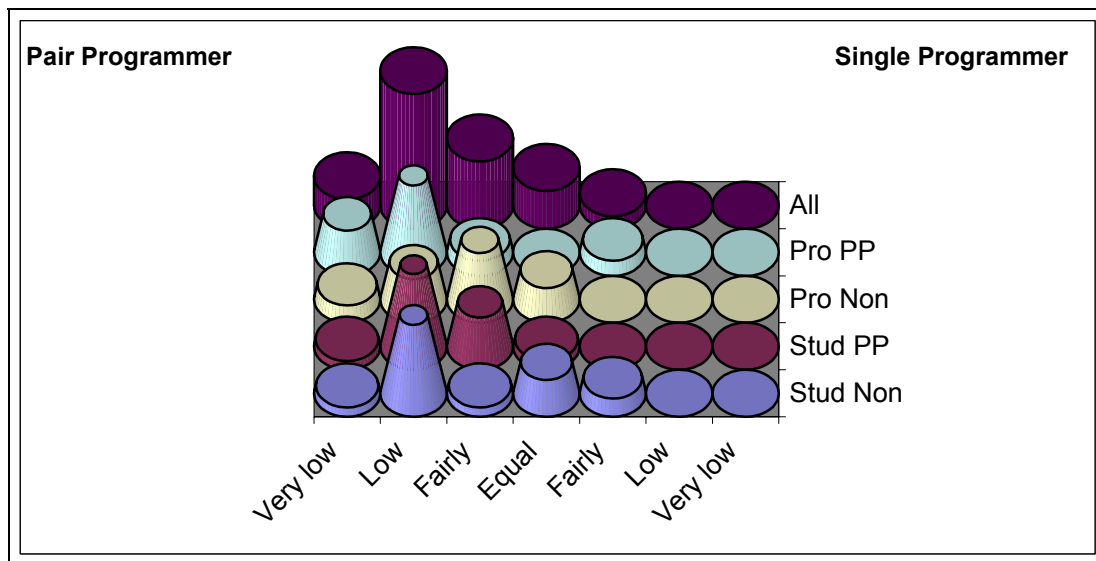
18. When writing documents as requirement specifications etc. Weight if Pair Programming is as good solution to exclude errors as having a document written by a single person and reviewed later on by another person?



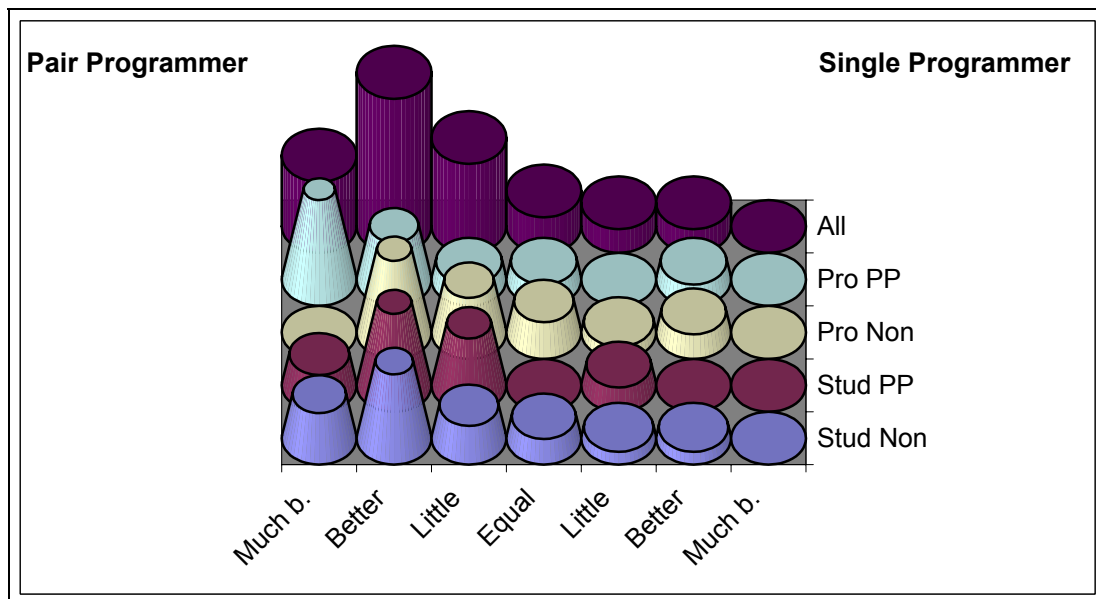
19. In architecture and design phases: is a discussion between a pair better than having two singles presenting two solutions? Who will create the best final solution in general?



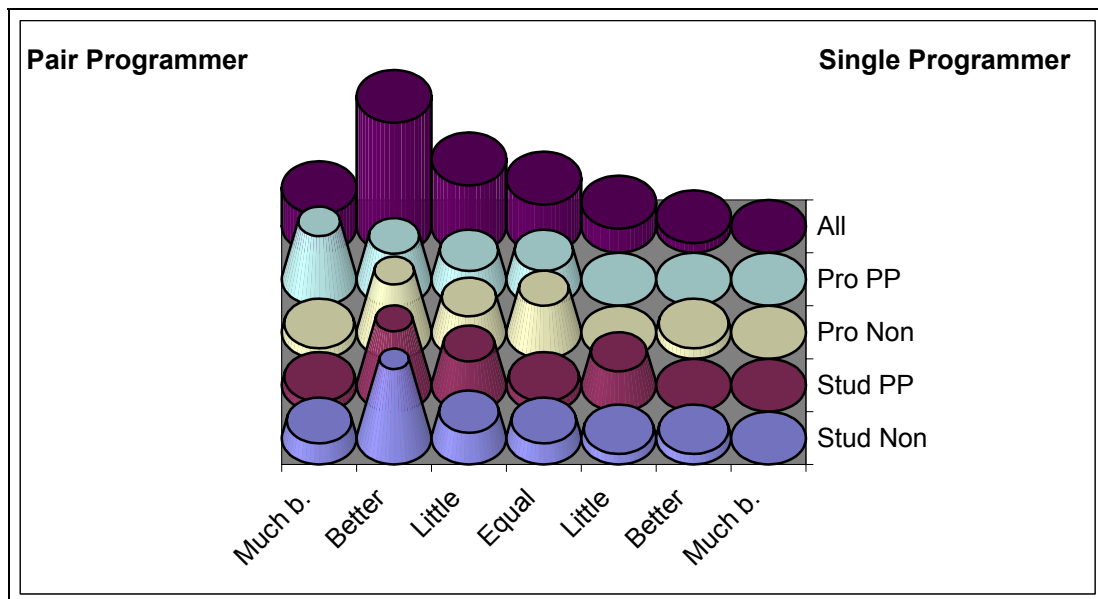
20. Do pair programmers produce code with equal amount of defects or lower/higher compared to a single? Mark where and how much lower amount of defects we can expect!



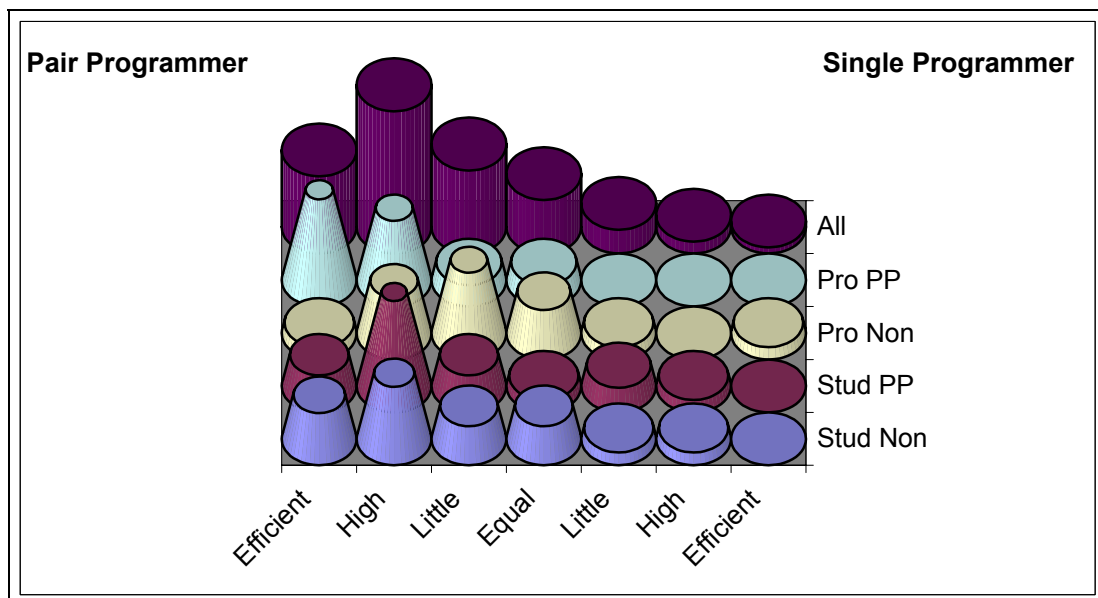
21. Do you believe developers find and implement better code solutions by using Pair Programming than singles? Weight your answer.



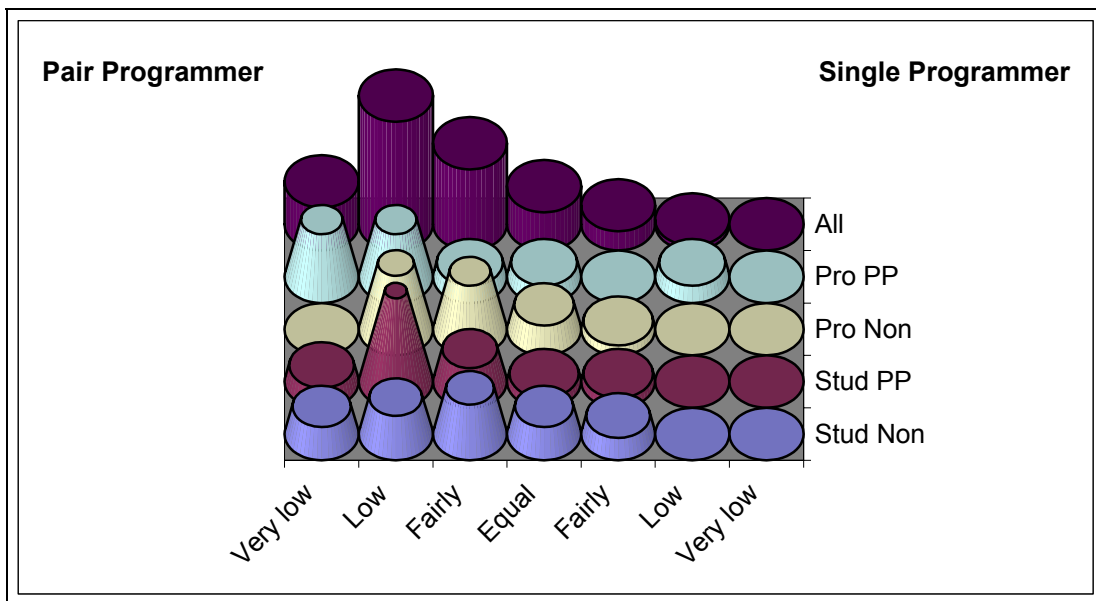
22. Do you believe people build better tests in general (basic, unit, integration, system etc.) with higher detection results by working as pairs than singles (exclude auto-generated tests)?



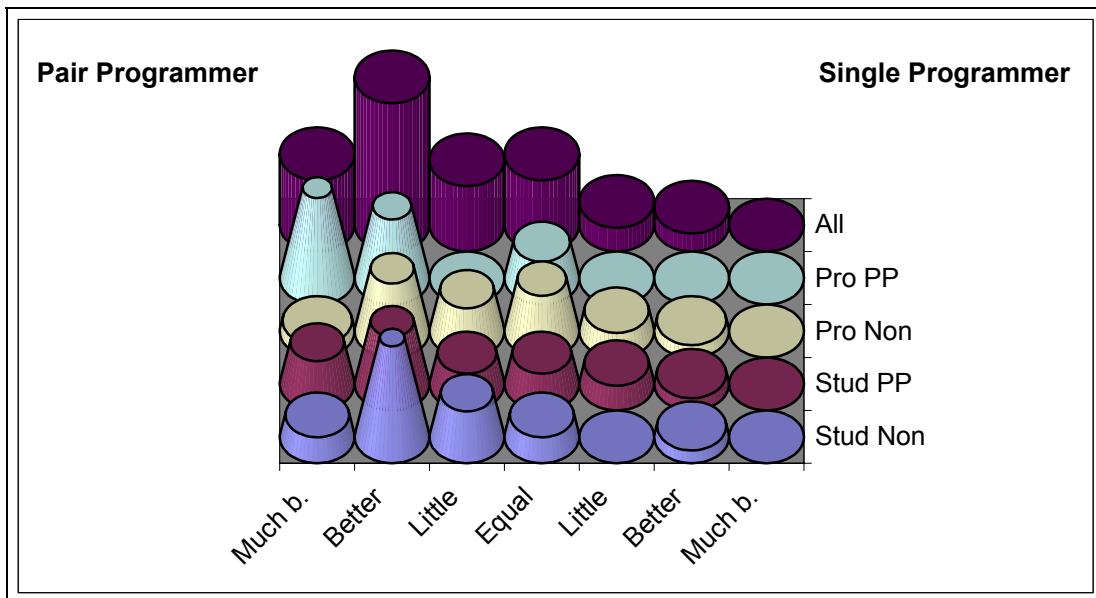
23. Compare the efficiency in finding defects in programming code. By constantly using Pair Programming and remove them before they happen or find them later having another person reviewing the code produced?



24. Which case above (Pair Programming vs. Single development with an additional review) will have the lowest amount of defects in developed code according to your opinion?



25. When people are tired or stressed is it easier to insert defects. Do you think that Pair Programming or Single Programming is best when working late hours, i.e. working overtime?

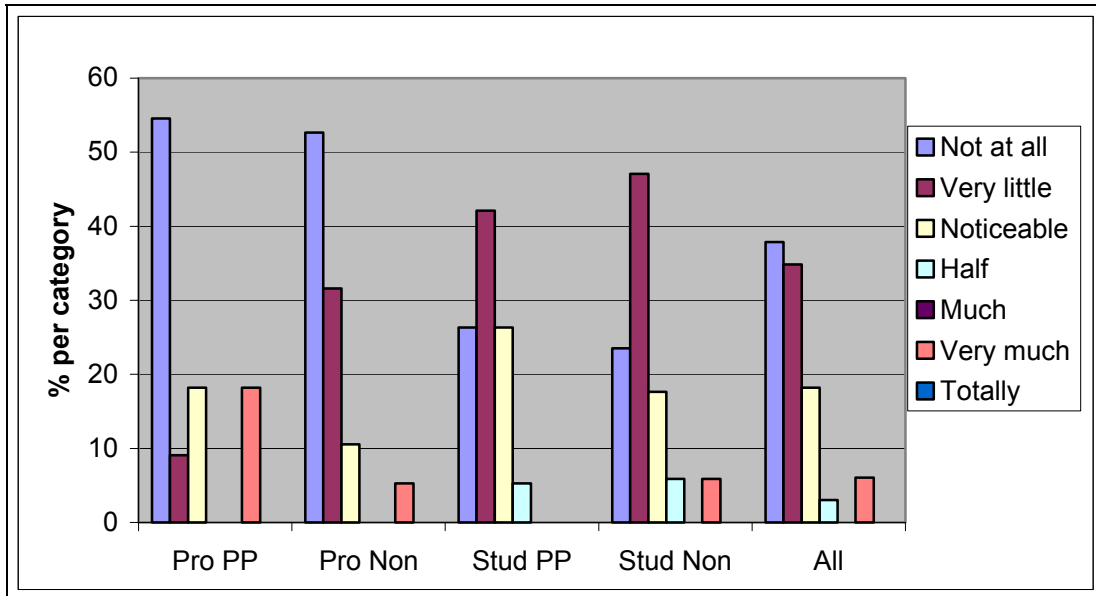


2.4 Part 4 (Efficiency, 4 questions).

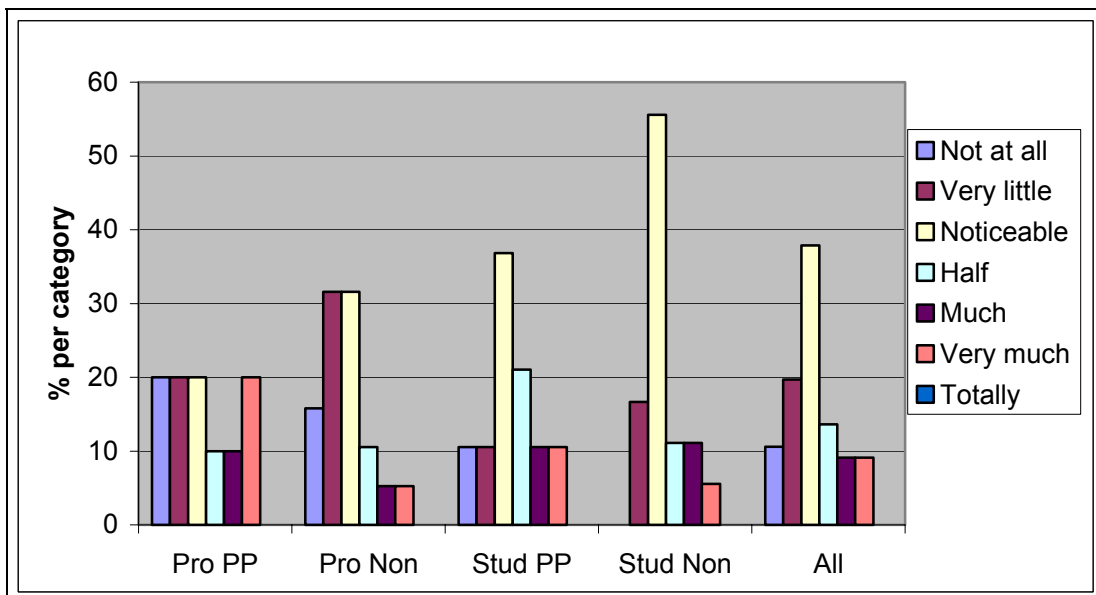
Also here we had 7 answer-alternatives accompanied with a percentage values. The percentage values where 0%, 10%, 30%, 50%, 70%, 90%, 100%

26. By using Pair Programming do you believe it is possible to:

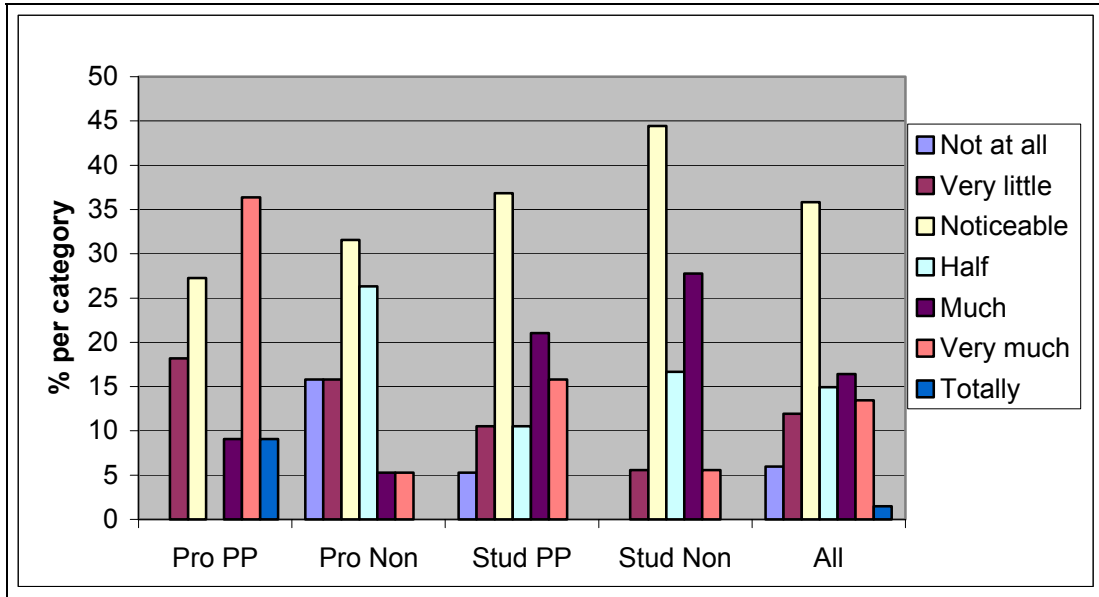
a. Reduce documentation?



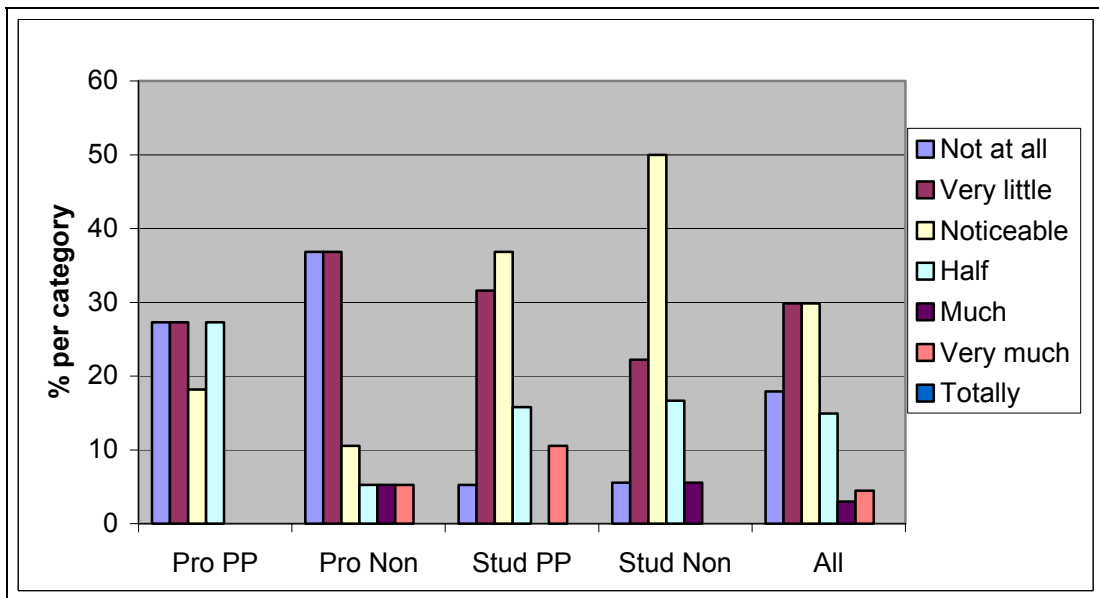
b. Reduce reviews/inspections of documents?



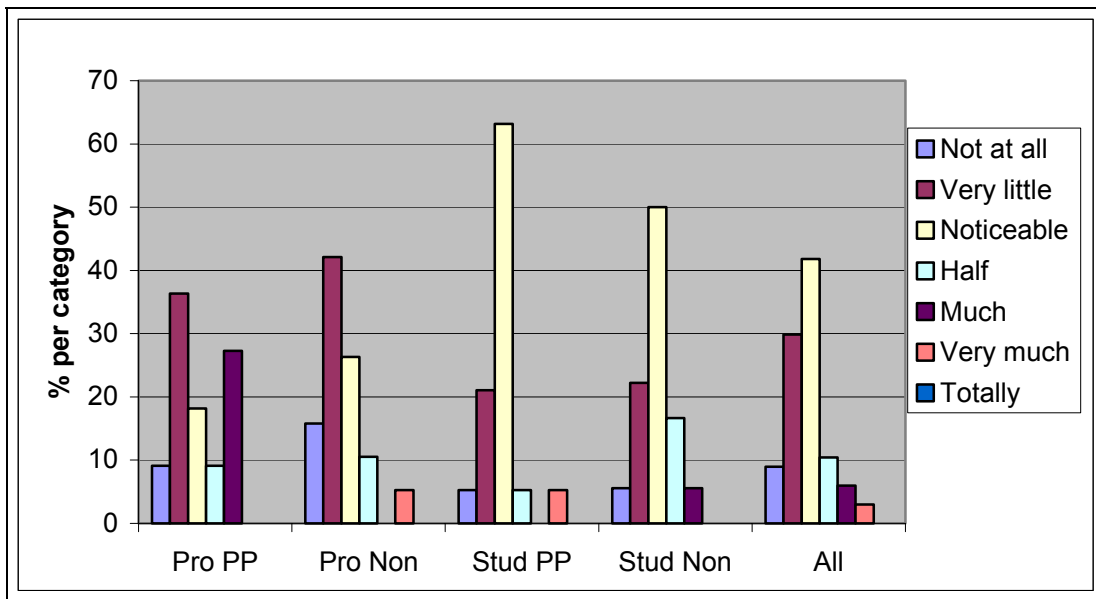
c. Reduce reviews/inspections of code?



d. Reduce creation of tests needed, due to early defect removal?

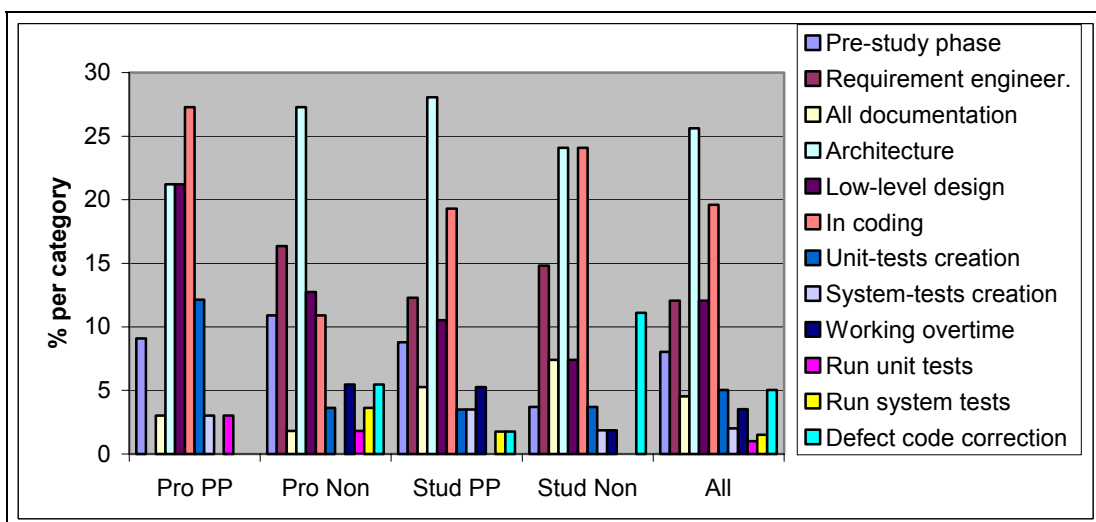


e. Reduce total test time needed?



27. Mark three options you think are most important to use Pair Programming in to get the highest benefit/profit in a project in total. The three best parts to use Pair Programming is in:

People have marked the 3 highest valued options. The graph is converted to percentage values, this still give the same picture, though.



2.5 Survey's open query comments (from part 4)

28. Do you have anything to add to this subject or things you feel is neglected in this survey, please feel free to add a comment etc.

22 of the respondents (32 %) chose to add a comment or wrote their reflections in the final open question. All their reflections and comments are all inserted below.

Those written in Swedish have been freely translated into English, where only one has been shortened. One in English has had its last sentence edited to better fit (duplicated text removed).

They are divided into two groups: first comes the comments from people, which have stated that they have PP experience and then people without PP experience.

With PP experience.

“According to my opinion PP is only good for specific tasks and not for all. In a large project is 80% of the code easy to write, to sit 2 is not only a waste of time it is also frustrating. For the other 20 % is PP very good and is used a lot”.

“When it comes to PP, you have to be a pair all the time, and this can in itself cause stress, as you do not get the time to take the much needed micro-pauses, as you have to be aware of what is going on all the time, if it is to be true PP. If you alternate between performing the tasks, where one does something else, even for a shorter period of time, important context information gets lost, that in itself inevitably will lead to the introduction of faults, by misunderstandings, oversight, or result in slowdowns as the work has to be explained. Taken from my own experiences, having a partner that i worked very well with, we stopped using PP because of the introduced stress, and a number of other reasons, reverting back to semi-PP, in the sense of SP with continuous crosschecking, sitting side by side with having a computer each. This way, we found that what we did as PP in a week, only took us about three days to accomplish using parallel SP”.

“The personal chemistry means a great role when people create pair for the first time. When everyone has got routine using PP then doesn't the personality affect as much”.

“I believe PP is good for documentation and design, also to solve difficult coding problems like algorithm, but it was hard to answer the survey when the questions was very general written.”

“A question how about how important the matching of pairs is, their background and experience. Can you mix an experienced programmer with a fresh student and still produce good results?”

“It is my opinion that PP is better suited for complex tasks than SP. On the other hand can PP be a waste of time for too simple tasks”.

“It is a matter of discussion whether or not the productivity gain is worth the extra resources spent on the activities”.

“I think you must consider the fact that different people have different backgrounds and ability to work with other people. Some people fit, some don't”.

Without PP experience.

“Regarding question 7: The time is very much depending on how well you know the other person, and how well you get along. If I have to work with a person that I get along well

with and that I have, for example, went to school with, then I think the start-up time will be < week or perhaps 1-2 weeks. If I'll have to work with a person that I don't really get along with and haven't been working with before, the start up time will be long, and the result may not be good”.

“How well it works depends on how well you work with your ‘partner’”

“I personally think it is good in some cases but not in all”.

“Generally I think that working in pairs results in a higher quality of documents, code, tests or whatever. But I do not think it is very time-efficient. The question is whether or not the increased quality is worth the cost measured in working time. Especially when the deadline is getting closer and closer...”

“A pair may be useful when solving problems and when testing but I think that the hands-on work such as coding and low-level design is better for a single person”.

“One thing that got neglected is the form PP should be used. I don't think it helps if two developers are sitting all the time in front of the same computer. I think it's better if the work gets split up in different parts. Somewhere the developers are sitting together but somewhere they work for their own. In the later case handovers should be done, so that both developers are working with both areas. Example: one developer doing some implementation work, while the other is writing test application. After a predefined time period they swap, thus, a combination of the PP and SP development”.

“I haven't worked myself as a pair programmer, only as a project manager when they have used it. Only worked myself as a single programmer”.

“Pair Programming seems to be suitable in some cases but does the benefit really cover the very expensive costs. For example I do not think two programmers produce code twice as fast as a single programmer”.

“PP is an excellent way to increase quality since you do an implicit inspection on code and document all the time. But we must consider if the value won compared to the extra resource used. If our resources had been infinite then would probably 80 % be a good figure. I know that studies have shown that PP leads to higher total gain (in quality, in reliability etc.), but my experience from PP is that the personal chemistry is one of the most important things, I think studies haven't take this aspect in their result. In short: PP works in some cases, I know from my own experience. But it have shown that it works with those people I like to work with while it doesn't work that good (resources exceed the value won) with people I don't feel comfortable with”.

“In coding it's just necessary to use PP in critical code parts”.

“I don't think PP is a bad method, but personally it's hard for me coding or writing when someone else is watching me”.

“It's sad that there is no possibility to add comments on the pre-defined answers to the questions”.

“Like a co-pilot in an airplane you have the social thing among the work and when working overtime. Humans are social in nature and feel more comfortable I believe when sharing a task with someone else, like the problems in the work with finding a good solution in the system being developed. Also the backup guaranty that if one become ill the other one can continue or explain for the others if necessary. In marriage/life I think your wife is your co-pilot and vice versa, although I don't think you should get married with your work ;)”

“I think it is important that we don't force people to PP. They must like to do it themselves to make it work. The personal chemistry is important”.