**SSW 555 Agile Methods for Software Development**

**Homework 5**

**Name: Mrunal Phadke**

**Suppose your boss asked you whether your organization should adopt pair programming.**

**1. Search for evidence that pair programming is better or worse for software development than traditional individual programming.**

**2. You may cite any sources, but peer-reviewed articles have more weight than individual opinions. You must find at least 3 sources (not including the required reading for this week), including at least one peer-reviewed academic article. That is, it must have appeared in a peer-reviewed journal or conference. One easy way to find articles of this type is to use Google Scholar: http://scholar.google.com/ However, not everything returned by Google Scholar was peer-reviewed. You need to check to make sure that it appeared in a peer-reviewed publication. Many of the articles that would be appropriate for this assignment were published in conferences and journals of the ACM or IEEE.**

**3. Provide complete citations to the articles you find. That is, include the following information: a. Name(s) of author(s) b. Place of publication (journal or conference name if peer-reviewed) c. Volume and number if in journal d. Page numbers of article e. Date of publication**

**4. Based on the evidence you find, give your answer to your boss's question.**

**------------------------------------------------------------------------------------------------------------**

**Answer:**

**Following are the evidences taken with their details:**

1. **Evidence 1**

* **Document Title:** Investigating students? behavior and code quality when applying pair-programming as a teaching technique in a Middle Eastern society.

**a. Names of authors:** Mamoun Nawahdah and Dima Taji.

**b. Place of publication:** Global Engineering Education Conference (EDUCON), 2016 IEEE

**c.** **DOI:** 10.1109/EDUCON.2016.7474527

Source: <http://ezproxy.stevens.edu:2109/document/7474527/>

**e. Date of Publication:** 23 May 2016

* **Abstract Introduction:**

Pair-programming is a software development technique that was introduced as part of Extreme Programming. In pair-programming, two developers share a computer to work together on developing one piece of code. To test Pair-programming effects on students' behavior and performance in a Middle Eastern society where some interaction restrictions are found, we devised an experiment that was carried out over an entire academic year consists of two semesters. The experiment targeted two sections per semester of an advanced computer programming course. The students of one of the sections worked in pairs during the lab sessions, applying pair-programming rules and techniques. The other section had students who worked individually, as it is the norm in most programming labs. Video recordings were recorded throughout the lab sessions, and then studied and analyzed. In addition, code samples were collected from the students to study the effect of pair-programming on the students' code quality. Through this experiment we found out that pair-programming has the potential to increase the students' confidence, their enjoyment of the course, and improved the course's completion rate. In addition, the students in the pair-programming sections showed that they were able to individually produce code of better quality than the students in the traditional section.

**2. Evidence 2**

* **Document Title:** Analyzing the cost and benefit of pair programming
* **a. Names of authors:** F. Padberg and M.M. Muller

**b. Place of publication:** Software Metrics Symposium, 2003. Proceedings. Ninth International

**c.DOI:**10.1109/METRIC.2003.1232465

Source: <http://ezproxy.stevens.edu:2109/document/1232465/>

**e. Date of Publication:** 23 September 2003

* **Abstract Introduction:**

We use a combination of metrics to understand, model, and evaluate the impact of pair programming on software development. Pair programming is a core technique in the hot process paradigm of extreme programming. At the expense of increased personnel cost, pair programming aims at increasing both the team productivity and the code quality as compared to conventional development. In order to evaluate pair programming, we use metrics from three different categories: process metrics such as the pair speed advantage of pair programming; product metrics such as the module breakdown structure of the software; and project context metrics such as the market pressure. The pair speed advantage is a metric tailored to pair programming and measures how much faster a pair of programmers completes programming tasks as compared to a single developer. We integrate the various metrics using an economic model for the business value of a development project. The model is based on the standard concept of net present value. If the market pressure is strong, the faster time to market of pair programming can balance the increased personnel cost. For a realistic sample project, we analyze the complex interplay between the various metrics integrated in our model. We study for which combinations of the market pressure and pair speed advantage the value of the pair programming project exceeds the value of the corresponding conventional project. When time to market is the decisive factor and programmer pairs are much faster than single developers, pair programming can increase the value of a project, but there also are realistic scenarios where the opposite is true. Such results clearly show that we must consider metrics from different categories in combination to assess the cost-benefit relation of pair programming.

**3. Evidence 3**

* **Document Title:** Are Two Heads Better than One? On the Effectiveness of Pair Programming
* **a. Names of authors:** Tore Dybý, Erik Arisholm, Dag I.K. Sjýberg, Jo E. Hannay and Forrest Shull

**b. Place of publication:** [IEEE Software](http://ezproxy.stevens.edu:2109/xpl/RecentIssue.jsp?punumber=52)

**c. Volume and number if in journal:** Volume: 24, Issue: 6, Nov.-Dec. 2007

**d. Page numbers of article:** 12 to 15

Source: <http://ezproxy.stevens.edu:2109/document/4375233/>

**e. Date of Publication:**05 November 2007

* **Abstract Introduction:**

Pair programming is a collaborative approach that makes working in pairs rather than individually the primary work style for code development. Because PP is a radically different approach than many developers are used to, it can be hard to predict the effects when a team switches to PP. Because projects focus on different things, this article concentrates on understanding general aspects related to effectiveness, specifically project duration, effort, and quality. Not unexpectedly, our meta-analysis showed that the question of whether two heads are better than one isn't precise enough to be meaningful. Given the evidence, the best answer is "it depends" - on both the programmer's expertise and the complexity of the system and tasks to be solved. Two heads are better than one for achieving correctness on highly complex programming tasks. They might also have a time gain on simpler tasks. Additional studies would be useful. For example, further investigation is clearly needed into the interaction of complexity and programmer experience and how they affect the appropriateness of a PP approach; our current understanding of this phenomenon rests chiefly on a single (although large) study. Only by understanding what makes pairs work and what makes them less efficient can we take steps to provide beneficial work conditions, to avoid detrimental conditions, and to avoid pairing altogether when conditions are detrimental. With the right cooks and the right combination of ingredients, the broth has the potential to be very good indeed.

----------------------------------------------------------------------------------------------------

**4. My Answer based on the evidences:**

To summarize, based on the research done, in majority of the articles, I found the following:

1. Pair Programming may prove very beneficial in tasks like: (WHEN?)

* Designing an optimal system based on user requirements.
* Writing complicated code modules.
* Improving the efficiency of already implemented system.
* If a deliverable is to made production ready in a small period of time.

2. The major advantages of Pair Programming being: (WHY?)

* Better Code Quality thus, less time investment in QA and Testing.
* Training of junior developers by senior developers directly, so less amount of time spent in learning company outlines and coding paradigms.
* Having constant conversations about problems, design, and patterns improves collaboration and team spirit.
* Reduction of downtime and more accountability of every individual.

3. When PP is not beneficial: (WHY NOT?)

* If the tasks are simple, PP may end up taking more time.
* When things can be executed in parallel by two experienced programmers.

So finally, I would explain my boss when Pair Programming is useful and when it is not. Also I would explain him, the benefits of using PP whenever it is applicable. Thus, decision of implementing PP strategy is going to be situational.