

Homework #1  
CSE 566: Software Project/ Process/ Quality Management  
(2015 spring)

Submitted By:

Madhu Meghana Talasila

(1207740881)

## **Part 1:**

Pair programming is a method in which both the programmers work together at a single work station. One of the programmer is called “driver” who is typing at the system and other is called “navigator” who observe the driver’s work. Pair programming not only limits to coding but covers all the stages of software development life cycle, requirements gathering, design, coding and testing. Programmers or developers switch their roles at regular intervals of time. This method of pair programming have proved benefits such as enhanced quality of work, confidence in programmers etc.

Now-a-days, because of globalization, companies are located in different locations and if pair programming needs to be implemented in such an environment it is called distributed pair programming. Distributed pair programming is also known as remote pair programming. Distributed pair programming is a programming style where programmers are located in different geographical locations but work together on a single task. This kind of pairing is called virtual pairing. Collocated pairs work together by speaking to each other, looking at each other, sharing computer, keyboard, and mouse and well aware of each other’s activities. In order to accomplish these kind of tasks in distributed pair programming a tool is required. The requirements of the tool which supports remote pair programming were identified by Cox and Greenberg [1]. They are:

1. “Provide a common, visually similar environment for all participants (two in case of pair programming)”.
2. “Provide timely feedback of all actions with in the workspace”.
3. “Supporting gesturing and deictic references (pointing gestures accompanied by verbal cues such as “here” or “this one”) “.
4. “Support workspace awareness (“the up-to-the-moment understanding of another person’s interaction with the share space”)”.

One of the tool that satisfies above requirements is open source screen sharing application “Virtual Network Computing” (VNC). VNC can replicate users screen onto different computers (Desktop Sharing). “Application output is sent to both the computers, while keyboard and mouse inputs from either computer is sent to the applications”. It also contains a microphone so that programmers can discuss while coding. This tool cannot be used as it is because if both the programmers use keyboard or mouse at same time then “keystrokes are interlaced into an intelligible stream”. One solution for the keystroke problem is to use “Tele Pointers” (to support gesturing and deictic references). Another problem is floor control. This can be done socially where programmers can choose when one wants to take floor control.

## **References:**

- [1]. “Empirical evaluation of distributed pair programming”, Brian Hanks, July 2008
- [2]. “Exploring the Efficacy of Distributed Pair Programming”, Prashant Baheti, Edward Gehringer, and David Stotts.
- [3]. “Distributed Pair Programming: An Empirical Study”, Brian F. Hanks

## **Part 2:**

The seven principles of lean are eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in, and see the whole.

For software development organization “A” which is following water fall model, if lean principles are applied it might look as in Figure 1. In waterfall model either business people or users are not aware of the hypothesis of product design in the beginning. To make waterfall a lean process a relative high design need to be maintained in the beginning of each phase. At each phase of waterfall model divide the phase into a number of sub teams and each team work independently on different tasks in one week of time for suppose. Then work done by one team is tested by another peer team. By doing so, team feels responsible for ones task as well as others task. Integrity can be developed. After everything is done for a particular phase then all the work done by sub teams in that phase is integrated – see the whole. As each team member is working on some task, waste can be eliminated. As code review is being done at every phase knowledge of code by each team member is increased. As this is water fall model, decision can be made as late as implementation phase has started. Delivery of each phase can be done as fast as possible.

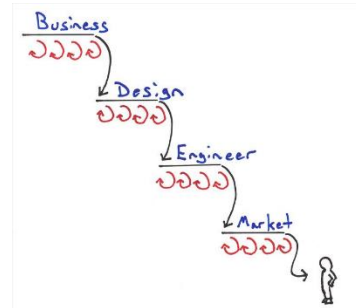


Figure 1 Lean Waterfall model [1]

For software development organization “B” which is following SCRUM model, if lean principles are applied then it look something like follows. “The Scrum sets forth rules for how components work together, but Lean Thinking provides a framework to further optimize the interplay of Scrum’s roles, artifacts, and events”[2]. Scrum model is considered as the basic lean implementation. Scrum in lean thinking needs the following changes. They are

1. Eliminate waste
  - a. Reduce delays like meetings, discussions
  - b. Work on items that are most important i.e, no need to clearly define the tasks on board. Programmers take up the tasks one by one from general items list
2. Include management
3. “Manage workflow explicitly and with limits on how much work can be taking place at any one time” [3].
4. Increase the quality by testing after each phase and taking reviews from clients.

## **References:**

- [1]. <http://www.pinterest.com/pin/51087776997068445/?z=1>
- [2]. <https://msdn.microsoft.com/en-us/library/jj161049.aspx>
- [3]. <http://www.netobjectives.com/resources/lean-scrum>