An Efficient Lean Software Development Model (LSD-Model)

Shehla Afzal¹, Aysha Ayub², Sheraz Malik³

¹²³Riphah School of Computing, Riphah International University, Faisalabad, Pakistan

Abstract— software development is a highly dynamic process so putting the right things to the right place at the right time is the main problem to be solved, it’s all about Lean. To develop a model which consist of lean principles and completely purely work for lean is the necessity of time hence to enhance efficiency in a proper way. This paper mainly focused to propose a model named Lean software Development (LSD-Model). Model is designed through all the software development phases, in addition to Lean principles, having properties of traceability and requirement change principle also. The model is iterative and incremental based. The model will be fast and efficient enough by implementing the lean principles and all the phases those are necessary to develop complete software.

Keywords— Lean, Lean Development, Efficient, Toyota Lean Principles, Lean development.

1. INTRODUCTION

It is the name given to Toyota’s method for producing and developing cars and it has not been used as a software development Process or a model actually but it includes the principles and guidelines to make software “lean”. Why is it of so much interest now to develop the product or any software “lean”. The answer is that the principles of Toyota’s can work at best everywhere. Else practices vary according the environment. It was examined first time that the underlying engineering principles of Toyota if to use carefully to develop vehicles the great success achieved hence it has been shown how they can be applied to software development as well.

If ever work on a software development project, there must be come some problems that: went over schedule? Went over budget? Didn’t meet the requirements of customer? Or it was cancelled? Actually software development is a highly dynamic process so putting the right things to the right place at the right time is the main problem to be solved, Hence it’s all about Lean. By reducing the waste and being able to open and to face any type of change is about Lean. The Waste is actually; without adding any mark able value it is anything that reduces time, resources, effort, space, or money. Thinking the software Development to be Lean is to see that the basic Lean principles can be applied to software development. The question is; what are the basic concepts of Lean Thinking? The answer is; these are: Value, the Value Stream, Flow, Pull, and Perfection. To develop a model which consist of lean principles and completely purely work for lean is the necessity hence to enhance efficiency in a proper way. In which the frequent releases which will provide earlier access to new functionality to the customer, which helps to accommodate changes if any requests by the customers come with minimal disturbances to the entire project.

2. LEAN THINKING

Lean thinking lies at the heart of lean manufacturing. As it is a totally customer focused dynamically knowledge driven process by which eliminate waste in a defined enterprise with the goal of creating value by all people continuously. Like enterprise the value is inclusive of value to the stakeholders like workers, the society and customers. Behind the lean thinking the following are the principles as below:

- Throughout to get the value stream synchronization of work flow
- Of an adaptive product development process follow
- Surity of flawless and error free information
- Perfect quality mission
- At all levels to promote leadership and effective making of decisions
- To use the capabilities and properties of all working people
- Care for the organizational learning
- To establish a trustworthy relationship based on assurance

These practices above prepare the organization to implement Lean Manufacturing based on lean principles So, to understand the big picture of lean Fig: 2.1, here it is below.

![Lean Thinking](image)

**Fig: 2.1. Big picture of Lean**

3. VALUE STREAM MAPPING

From order taking stage to delivery stage the value stream mapping is the mapping of all the activities for a product. To identify the value addition activities and also to find out the non-value addition activities is the
basic purpose of VSM is lean. Figure 3.1 explains what gain through VSM in a product.

**Fig: 3.1. Value Stream Mapping**

**4. REDUCING WASTE STREAM**

Customer is not willing to pay for is the Waste something that is not useful (George, 2002). In well form in Figure 4.1 the types of wastes are described below. A very important factor of lean is waste reducing. Waste can be of any type described below.

![Types of Waste Diagram](image)

**Fig: 3.2. Wastes in Lean**

### 4.1 Defects
When products and components are defective damaged rework is required and they have to be reworked. By bad manufacturing processes defects are caused may be caused by human or machine errors can be also. It increases manufacturing costs and takes additional time for the final product. The items have to be discarded some times in worst case scenario.

### 4.2 Overproduction
When the company is producing more than need that is actually overproduction which arises than the customer really need. Both this may include the production of products or components for which there are no orders, it also include the more items production than currently required. Worst kind of waste this is actually, because it usually results in multiplies other kinds of waste as well. Like it also increases rework rate, processing time, inventory, waiting, as well as unnecessary motion and transportation.

In a manufacturing process each step is dependent on the upstream and downstream stage as well. If production process increased, employees work responses late, equipment’s non availability, delayed information or materials, wasted production time and the cost of production if increased.

### 4.3 Waiting
In a manufacturing process each step is dependent on the upstream and downstream stage as well. If production process increased, employees work responses late, equipment’s non availability, delayed information or materials, wasted production time and the cost of production if increased.

### 4.4 Transportation
Unnecessary movement of information, components and products from one place to another place it is actually. With unnecessary movement it occurs usually, during movement the product damages, lost its parts and systems, is also included in it (Rodriguez et al., 2012).

### 4.5 Inventory
Quantity of items required to manufacture a product on stock is the Inventory. Cause costs to the company these goods. They utilize valuable warehouse space when they are not used, may become out of date may require raw materials, for more important goods that cannot be used. To make sure their system controls the inventory in competitive organizations, so that on unwanted items money is not being wasted.

### 4.6 Excess processing
The extra steps in the manufacturing process are actually over processing which need to be taken. To produce of higher quality products that was not required. Due to incorrectly used equipment this may happen, rework process errors, design of a poor process or bad communication. To not check what the real customers’ requirements is this can be also result of it (Wang and Cawley, 2012).

### 4.7 Motion
When employee is moving around his work space then unnecessary movements occur and as a result of this may time wastes and efforts. Poor working standard practices, poor process design layout these all kind of unnecessary motion is being caused of it.

**5. LSD-MODEL**

The framework is based on the quality assurance steps including lean practices. The steps Included: 1) Current Project Characterization, 2) Quantifiable Goals and Measurements Setting, 3) To Execute Lean
Processes, 4) To Collect Data Validate, 5) To Analyze Collected Data and Recommend Improvements. To perform development the main contribution of this is for lean software development. Our method to achieve continuous the steps marked grey apply improvement towards a lean software process. Metrics those are included in LSD Model are: firstly Requirement specifications, to generate Test Cases, the Change Requests, remove Faults and Failures, wastes control all have been included.

5.1 RCC Rule:
This rule has been working to control the change, there are three steps have been included in requirement change control principal those consist of: Planning, Trying and checking; either the change is applicable or not; described in figure 3.5 below:

![RCC Rule](image)

Fig 5.1: RCC Rule

LSD-Model has the following stages:

5.2 RS Stage:
- RS means Requirement specification, the things those have been noticed in this phase of development cycle are:
  - To gather high level requirement.
  - To gather the requirement in testing phase.
  - Prioritization of requirement that which requirement is necessary to implement first.
  - To notify at the start which requirement change can induce risks and what were those risks.
  - At this stage each and every type of requirement has been notified.
    - To find out the detailed requirement.
    - Requirement in design and implementation phase.
5.3 DD stage:
- DD means design and development stage, in this step of development cycle the important point to remember which will be implement are:
  - Low level designing of project to be desire.
  - A Detailed Design after finding out detailed requirements in first phase RS.
  - Development of all requirements low and as well as high.

5.4 TA stage:
- TA means Testing and acceptance stage, in this step of development cycle the important point to remember are the following:
  - Design of test Cases (There will be one test case for one Individual Requirement, every requirement will have its own test case design for specifically to check that requirement and with its implementation checking.). The following are the test cases have to develop for the requirements.
    - Unit Test
    - Acceptance Test
    - Integration Test
    - System Test
    - Function Test
  - Where unit testing outcomes only like binary either pass or fail means yes or no. Like unit testing the acceptance test suggests also what is actually the reason of failure and applicable things that can be also tell. Integration is also called definition of done, testing in which both components (software and hardware) are combined to assure either that they both can interact and cooperate with each other according to expectations and requirements to give the user the required output. There should be a testing environment exactly so close reflects to original.
    - Specifications Executable .Requirements in the form of executable like running form called "customer tests", and your design instead of non-executable "static" documentation as executable developer tests.
    - Iteration Modeling. At the start of each iteration there will do a bit of modeling as part of iteration planning activitie
    - Prioritized Requirements. Teams implement requirements in priority order, as defined by their stakeholders.

One of the principles of the lean development that is it is for delivering software that must be workable frequently. Which means that the small bits having functionality assembled together and to make a group of them for delivery of system. Function testing is to check the functionality either it was the desired working or not.
- To handle change Requests either request are approved or not.
- Through customer involvement Removing of faults and failures

5.6 DR stage:
- DR means Developed and Released stage, in this step of development cycle the product being developed fully after iterations and released to the customer provided to customer after validation and verification to the customer. The best practices those have been introduced in Lean are:
  - Active Stakeholder Participation. Stakeholders and customers in a timely manner should provide information, in a timely manner should make decisions, and in the development process through the use of inclusive tools and techniques be as actively involved.
  - Architecture designing. At the very start of a lean project to identify a viable technical strategy for your solution. There will be need to do some initial, high-level architectural modeling.
  - Continuous Documentation. Write deliverable documentation in small bits/chunks of development throughout the lifecycle of lean development.
  - Document Late. As late as possible deliverable should be write documentation, avoiding imaginary ideas that are in favor of stable information likely to future change.

References: