

Kan-Ban : A Mean to Survive (“In Software Development Perspective”)

Er.Kirtesh Jailia

M.Tech, SE

Independent Researcher

kirtesh.jailia@gmail.com

Manisha Jailia

MCA, Sr. Assistant Professor (CS)

Banasthali University, Tonk (Raj.)

manishajailia@yahoo.co.in

Abstract - “Software development” this term is so valuable for the software industries and its clients, because this term decides how well they can survive in the market. Development methodology plays an important role in their survival. But as we see in the phase of recession the client wants faster delivery but software industries were unable to deliver and in the result most of the software companies have lost their market share. We saw that during recession some companies were drowning due to the reason of MISSING DEADLINE, OVER DEVELOPMENT, INEFFECTIVE MANAGEMENT and many more. We knew that there are several reasons for the recession in the software market, but the most important thing is to survive in such recession phase by sustaining the customers, by performing orders as early as customer wants and many more. To survive in such conditions we have to rely on lean principles and they will definitely acts as survival tool in recession [1]. Her in this paper we are going to discuss about the term “Kan-ban” which is based on the lean thinking and try to find out the answer “How Kan-ban act as a shield to retain success in the recession phase”?

Keywords: Kan-ban. Lean thinking, Lean thinking based tool, Kan-ban for survival.

WHAT IS LEAN THINKING & LEAN PRINCIPLES?

Lean thinking may be a new term for the readers, so first of all let’s discuss about the term lean thinking. Lean Thinking can be easily defined as “A thinking which is based on lean principles”. Lean principles are the bedrock of the lean thinking and they are as follows Lean software development is based on basic five principles which makes it a special way of software development, let discuss all of them one by one.

- Specify **value** from the standpoint of the end customer by product family.
- Identify all the steps in the **value stream** for each product family, eliminating every step and every action and every practice that does not create value.
- Make the remaining value-creating steps occur in a tight and integrated sequence so the product will **flow** smoothly toward the customer.
- As flow is introduced, let customers **pull** value from the next upstream activity.

- As these steps lead to greater transparency, enabling managers and teams to eliminate further waste, pursue **perfection** through continuous improvement. [2]

ORIGIN OF LEAN THINKING

Lean thinking can be traced to Toyota (Liker and Hoseus, 2008, 15). A Japanese technique, called Just-In-Time (JIT) (Ohno, 1988)1 (Chapter 2.2.1) originates in the late 1940s and early 1950s when Taiichi Ohno developed Kan-bans in order to control production between processes and in order to implement Just-In-Time manufacturing at Toyota (Gross and McInnis, 2003, 1). The TPS is probably the most famous example of how to attain success with the aid of the Lean approach. Lean principles are, however, not enough. What is needed is the culture built into the whole corporation. The fourteen principles of doing things in Toyota, called the principles of the Toyota Way, are statements of beliefs and values which are about Toyota’s culture. This culture can be encapsulated in four items:

- (1) Long-term philosophy (the purpose and reasons of Toyota’s existence),
- (2) Lean processes (which lead to operational excellence) which emphasize continuous elimination of waste.
- (3) Developing and challenging people and partners through long-term relationships, and,
- (4) Organizational learning driven by problem solving and continuous improvement. Despite its success, the way of doing things in Toyota is always changing (Liker and Hoseus, 2008).

WHY THE TERM LEAN IS USED?

The word lean in software development came from the automobile industry amazing but it is true. Toyota motors are the first company which introduced the Lean manufacturing. This was used when there is demand increased too folds and a single mistake would lead to a missed deadline means business loss. The problems that Toyota faced foreshadowed the problems that an increasing number of businesses would face in the late 20th century: customers want more variety and better quality, and they want it now. Traditional business practices are often poorly

suited to deal with these expectations, and management theorists became increasingly restless in their pursuit of new understanding. Much as the industrial practice of yesteryear evolved into more general business practice, the principles of the Toyota Production System were abstracted into the general business philosophy we now call Lean.

WHAT IS KAN-BAN?

Now, Lets discuss that what is Kan-ban?

The Japanese word Kan-ban refers to a signboard. When the term is used in manufacturing, it means a scheduling system that hints what, when, and how much to produce. Toyota, for example, has successfully applied Kan-ban in practice as one part of TPS resulting in a way for promoting improvements (Hiranabe, 2008).

A Kan-ban is a card containing all the information required to be done on a product at each stage along its path to completion and which parts are needed at subsequent processes. These cards are used to control work-in-progress (W.I.P.), production, and inventory flow. A Kan-ban System allows a company to use Just-In-Time (J.I.T) Production and Ordering Systems that allow them to minimize their inventories while still satisfying customer demands. A Kan-ban System consists of a set of these cards, with one being allocated for each part being manufactured, that travel between preceding and subsequent processes.[3]

Kan-ban does not intervene in management despite its importance, i.e., how to do things. Instead, it is inclusive of management. In other words, management is involved and it is committed to abide by the methods the teams have selected to do their work. In addition, management is part of discussions about how the work is being tracked and performed. Kanban combines defining and managing a workflow: this workflow based on queues and control loops is managed by limiting WIPs (Shalloway et al., 2009, 98–100).

WHAT ARE KAN-BAN PRINCIPLES (in software engineering scenario)?

The basic principles of Kan-ban for software engineering are as follows:

1. Limit Work in Process (WIP)
2. Pull value through (with WIP limit)
3. Make it visible (Visual Control)
4. Increase throughput
5. Fixed Kan-ban Backlog
6. Quality is embedded in (not inspected in)

These are the basic principles over which we can develop a Kan-ban system for the software engineering. But before applying these principles to software engineering we have to adhere ourselves to the rules for implementing it, these rules are as follows.

RULES FOR KAN-BAN SYSTEM:

1. **Visualize Workflow** - The workflow of the knowledge work of today is inherently not visible as it is “hidden” in information systems. Visualizing the flow of work and making it visible is core to building an understanding how work works. Without understanding the workflow making the right changes is harder. A common way to visualize the workflow is to use a card wall with cards and columns. The columns on the card wall representing the different states or steps in the workflow and the cards the [feature/ story/ task/ result](#) of the workflow A visual representation of the process lets you see exactly how tasks change from being “not done” to “done right”. The more complex a process is, the more useful and important creating a visual workflow becomes, but Kan-ban can be used if there are just a few steps (do, doing, done) or a lot of steps (plan, design, draft, approve, schedule, implement, test, integrate, deploy). However complex the project may be, creating a Kan-ban board allows you to see the status of the work being done at a glance.[4]



Fig. A “Visualize Workflow in Kanban”

2. **Limit Work in Process (WIP)** - Get more done by doing less. It may seem counterintuitive, but it is a powerful idea that has been proven time and time again to be true. There is a limit to the number of things you can be working on and still do them well, and that limit is often lower than you think. Whether a project is simple or complex or whether the team is small or large, there is an optimal amount of work that can be in the process at one time without sacrificing efficiency. It’s not uncommon to find that doing ten things at once takes a week, but doing two things at once takes hours, resulting in twenty things being done by the end of the week. Kan-ban metrics lets you find that optimal number. If we set your WIP limit to two or three for tasks being “in progress”, it helps you focus on exactly those tasks. And you have an explicit motivation for getting those tasks done as you should only start new stuff if there is an empty space on your Kan-ban board for another “in progress” task.

3. Measure and Improve Flow - Improvement should always be based on objective measurements, and Kan-ban is no different. Finding and applying good metrics is usually a difficult step, but a few simple measures automatically generated by an application like .One of the great things about Kan-ban is that you apply it to your existing process. You are simply identifying ways to improve what you are already doing, so you don't have to start from scratch and you don't have to worry about "throwing the baby out with the bath water" - meaning that you won't lose the things you are already doing well. No sudden changes means there is minimal risk in applying Kan-ban as part of your improvement journey.

IMPACT OF KAN-BAN ON SOFTWARE DEVELOPMENT

Kan-ban effect on the software development is positive because it provides several ways to track your progress and at the same time it offers fast and easy working. We can support this statement from the following research document lines.....

"We claim that Kan-ban has a positive impact on software development when used appropriately. Kanban was found to help in revealing waste, which can save the resources and time of software development projects. Visualizing the progress was found to aid in controlling project activities in flexible yet coherent ways and even to motivate software development teams. In addition, Kan-ban was also found to enable the project teams rapidly to adapt to continuously changing situations."[5]

KAN-BAN AS A MEAN TO SURVIVE

Now after understanding about the Kan-ban, let's discuss about what makes Kan-ban as a shield for software industries in the recession phase. The major problems during recession are as follows

1. Meeting Deadline.
2. Customer satisfaction

These are the most important things for survival in recession, because as we see that in recession though the amount of work is lesser but the deadlines are quite inflexible. Means you have to work with great speed to meet the deadlines and at the same time you have to meet with the desired customer satisfaction. With the help of Lean thinking and by using Kan-ban you can achieve the speed as well as quality.

Kan-ban act as a shield because it provides---

1. **Visual workflow:** Through which you can track your progress and can easily speed up your work.

2. **Limit WIP:** With the Kan-ban approach you can limit your input queue elements, but it doesn't mean that Kan-ban opposes multitasking, it supports optimal tasking.

3. **Effective risk management and better quality:**

With the Kan-ban approach one can reduce the risk of sudden changes in customer requirements, because of small deliveries, it become easier to satisfy the customer needs. Kan-ban also provides pull from demand philosophy through which extra features (Which may cause to missing deadlines) can be eliminated, and the final product will be of good quality.

4. **Measure the lead time:** (average time to complete one item, sometimes called "cycle time"), optimize the process to make lead time as small and predictable as possible.

5. **Efficient Team Management:** With the help of Kan-ban philosophy you can manage the team efficiently because it provides a cross functional working environment in which there is no compulsion that all phases of software development requires only skilled people respective to that particular phase.

CONCLUSION

At last while concluding this paper, we can say that Kan-ban is a great philosophy and it has shown its power in various fields such as in automobile industries, personnel management etc. Kan-ban philosophy in software development is also showing positive effect. The Kan-ban philosophy will definitely works as a shield during recession phase and will give the software industry a sustainment.

REFERENCES

- [1] Lean software development ("As a survival tool in recession") International Journal of Software Engineering and Its Applications Vol. 5 No. 3, July, 2011.
- [2] <http://shapingsoftware.com/2009/06/15/introduction-to-lean-software-development/>
- [3] <http://www.beyondlean.com/kanban.html>
- [4] Getting started with kanban by paul klipp, www.kanbanery.com
- [5] Karhatsu, H., Ikonen, M.3, Kettunen, P.,Fagerholm, F., and Abrahamsson, P. (2010). Building blocks for self-organizing software development teams: A framework model and empirical pilot study. In Proceedings of the 2nd International Conference on Software Technology and Engineering (ICSTE '10), volume 1, pages V1-297–V1-304. IEEE.
- [6] Hiranabe, K. (2008). Kanban applied to software development: From agile to lean. <http://www.infoq.com/articles/hiranabe-leanagile-kanban> [4-May-2011].
- [7] Liker, J. K. and Hoseus, M. (2008). Toyota Culture: The heart and soul of the Toyota Way. McGraw-Hill, New York, NY, USA.
- [8] Shalloway, A., Beaver, G., and Trott, J. R. (2009). Lean-agile software development: Achieving enterprise agility. Pearson Education, Inc. /Addison-Wesley, Boston, Massachusetts, USA.
- [9] Gross, J. M. and McInnis, K. R. (2003). Kanban made simple: Demystifying and applying Toyota's legendary manufacturing process. AMACOM, New York, NY, USA.

AUTHOR'S PROFILE

Er.Kirtesh Jailia

M.Tech, S.E.

Independent Researcher

kirtesh.jailia@gmail.com

Manisha Jailia

MCA, Sr. Assistant Professor (CS)

Banasthali University, Tonk (Raj.)

manishajailia@yahoo.co.in