Kanban in Industrial Engineering and Software Engineering:

A systematic literature review

Muhammad Ovais Ahmad, Jouni Markkula, Markku Oivo, Bolaji Adeyemi
Dept. of information processing science
University of Oulu,
Oulu, Finland
e-mail: {Muhammad.Ahmad, Jouni.Markkula, Markku.Oivo}@ oulu.fi,
Bolaji.Adeyemi@student.oulu.fi

***Abstract*— There is a growing interest about Kanban in software engineering due to its many advantages, such as, reduced lead-time and improved team communication. Kanban originates from Toyota manufacturing and in 2004 it was introduced to software engineering. However, there is lack of a clear explanation of its principles and practices in software engineering. The objective of this study is to explore Kanban in industrial engineering literature using systematic literature review method. The search strategy revealed 1552 papers, of which 52 were identified as primary studies relevant to our research. From the primary studies, five variations of Kanban were identified together with implementation principles and benefits. These were extracted and summarized for the guidance of practitioners interested in adopting Kanban for software development. The findings of this literature review help researchers and practitioners to gain a better understanding of the Kanban and its use in industrial engineering in order to improve its usage in software engineering.**

***Keywords*- Kanban; software development; systematic literature review; lean, agile, Kanban variants.**

I. INTRODUCTION

In the last decade, lean approach in software development is increasingly popular. The aim of the lean approach is to deliver value to customers more effectively and efficiently through the process of finding and eliminating waste, which is a huge impediment to the productivity and quality offered by an organization [39]. Lean approach was first applied in manufacturing industry, devised at Toyota and originally called the Toyota Production System (TPS). According to Ohno [40] TPS was established based on two concepts. The first is "automation with a human touch", which means when a problem occurs, the equipment stops immediately, preventing defective products from being produced. The second concept is "Just-in-Time," which means in each process produces only what is needed by the next process in a continuous flow.

Kanban is a subsystem of TPS, created initially to control inventory levels and the production and supply of components and raw materials [2][14]. Kanban was created to fulfil specific needs of Toyota Company, i.e., to work effectively under specific production and market conditions. Nowadays, Kanban is not only used in manufacturing industries, but also in software development and services, the health sector, and many more domains [2][13]. Kanban facilitates high production volume, high capacity utilization, and reduced production time and work-in-process [14]. Further, Kanban controls the flow of parts along down-stream processing, which creates a “pull” action with material required.

Kanban entered the software development field in 2004, when David Anderson introduced it in practice while assisting a software development team at Microsoft [1][2][3]. Kanban is used to visualise work, limit work in progress, and identify process constraints to achieve flow and yet focus on a single item at a given time [3]. In general, Kanban aims to bring visibility to work, and to enhance communication, collaboration, and integration between software developers, testers, and support teams, resulting in rapid software development and continuous delivery to the customer [1][2][3][4][5][6][7][41]. In software development, the goal of practising Kanban is to visualise and improve the flow of value by optimizing a cycle time, while respecting work in progress limits [1][11][41].

Kanban in software development uses cards to represent work items. Software practitioners have implemented Kanban techniques using physical materials such as sticky notes on a board. Signals are mostly generated from a software work tracking system [1], for example Agile Zen, and Jira. In software development, Kanban has five core principles: *visualise workflow, limit work in progress, measure and manage flow, make process policies explicit, and use models to recognise improvement and opportunities* [1]. Kanban principles are applied using a board which visualises the flow of activities of the process in various columns. Cards are used for each working item on the Kanban board to show its current state. The flow of work items through the process is optimised by limiting the work in progress in each activity column to a maximum number of items that can be pulled into the column. In this manner, the team effectively visualises their workflow, limits work in progress items in each stage, and monitor the cycle time from start to finish.

Kanban is becoming more popular in software development. A strong practitioner-driven movement has emerged supporting its use [3]. Recent studies [2][41] of it in software development shows both benefits and challenges in the adoption of Kanban. The benefits of using Kanban in software development are: a better understanding of the

 ISBN: 978-1-61208-438-1

whole process, improved team communication and coordination with stakeholders, and improved customer satisfaction [2][3][41]. Currently, Kanban is being increasingly adopted to complement Scrum and other agile methods in software processes. With the growing number of studies on Kanban in software development, some have reported a number of challenges, such as hard to manage work in progress, task prioritization, and misunderstanding of core Kanban principles [2][7][8][9]. Notably, evidence of challenges relating to both its implementation and operationalization exist. As Kanban originated in manufacturing industry, by investigate its features in the industrial engineering field literature we can learn how its basic idea is transformed to software engineering.

The purpose of this paper is to systematically review and analyse the Kanban variations, benefits, characteristics, and implementation principles from its industrial engineering context, and to compare them with Kanban in software engineering; in order to find ways to improve its implementation in the software engineering field. To the best of our knowledge, this is the first study that systematically investigates Kanban in industrial engineering and software engineering and how knowledge of Kanban usage in industrial engineering could improve software engineering.

The rest of this paper is organized as follows. Section II describes the research method employed in this study. Section III, describes the results of the review, describes analysis of the results, and provides discussion. Finally, Section IV reports a conclusion and recommendations for further research.

II. RESEARCH METHOD

We designed this systematic literature review by following the guidelines of Kitchenham and Charters [10]. According to these guidelines, we undertook the review in several steps:

- The development of review protocol,
- The identification of inclusion and exclusion criteria,
- A search for relevant studies,
- Data extraction, and
- Synthesis and reporting of results.

The review protocol was developed jointly by the authors of this paper while also carrying out identification and selection of the primary studies on adherence to the specified protocol. All of the steps of the protocol are described below in this section. The objective of the review was to answer the following research questions:

- How does software engineering Kanban differ from industrial engineering Kanban in terms of characteristics?
- How are variants of Kanban and their characteristics and benefits described in industrial engineering?
- How can industrial engineering Kanban implementation principles be used in software engineering?

A. Data Sources and Search Strategies

Four major databases were selected as literature sources: Scopus, IEEE Xplore, Web of Science, and ACM. The rationale for this choice is that these are the relevant databases having large collections of high quality, peer reviewed conferences and journal papers. Relevant studies were searched in these databases by using the following search strings, which were combined with the OR, operator:

1. Pull system AND Kanban
2. Toyota Production System AND Kanban
3. Kanban AND Inventory system
4. Lean AND Kanban
5. Kanban AND (Implementation OR Benefits OR waste elimination)
6. Just-In-Time OR JIT AND Kanban

"Operation OR production" was added at the end of each search string to focus the search to industrial engineering literature.

B. Selection process and inclusion decisions

In Step 1, the titles, abstracts, and keywords of papers were searched using the above mentioned search terms. In Step 1 search resulted in a total of 1,552 papers. In Step 2, duplicate papers were excluded.

In Step 3, two researchers sat together and went through the titles of all studies that resulted from Step 2, to determine their relevance. The systematic review included peer reviewed qualitative, quantitative and simulation research studies published from 1977 up to 2013. Only studies written in English were included. Additionally, editorials, prefaces, correspondence, discussions, lessons learned and expert-opinion papers were also excluded.

In this step, we excluded studies that were clearly not about Kanban in industrial engineering. However, titles are not always clear indicators of what an article is about. In such cases, the articles were included for review in the next step.

In Step 4, each of the remaining papers was assessed with regard to quality and relevance to our study. In assessing the quality of studies, we developed a checklist outlining the major quality criteria expected from the primary studies. We built the list based on quality criteria adapted from Kitchenham and Charters [10]. In this step, two researchers independently reviewed 257 papers. All disagreements were resolved by discussion that included two researchers before proceeding to the next stage.

The evaluation was based on the following criteria: objective of the study, context description, research design, data collection and analysis, and justification of findings. Of the evaluated 257 studies 52 were finally accepted and included as the primary studies for our research. The rest of the papers were excluded because they did not pass the minimum quality threshold.

The following Table I present the systematic review process, which was carried out through this study to identify the primary studies, as well as the number of papers identified at each stage.

TABLE I. STEPS OF THE STUDY SELECTION PROCESS.

Steps	***No. of studies***
Step 1: Search of the bibliographic databases	1552
Step 2: Removing of the duplicates studies	1365
Step 3: Inclusion based on title and abstract	1138
Step 4: Inclusion based on full text scanning	257
Step 5: Quality evaluation based on full paper reading	52

C. Data extraction, synthesis and analysis

Based on the guidance provided in Cruzes and Dyba [12], we extracted three types of data: Kanban description, different variations of Kanban, and benefits of using Kanban. We used a thematic analysis technique. Coding technique was used manually to identify the relevant text in finally included papers while reading the entire paper. All primary studies were categorized by paying close attention to the type of studies which are as follows: conceptual studies, simulation studies, mathematical approaches, surveys, literature reviews, and case studies. Data from all primary studies were extracted by two authors in consensus meetings.

III. RESULTS

We identified 52 peer reviewed primary studies on Kanban to address our research questions. Most of the studies were published in journals (46 or 89%), while six (11%) were published in conferences. The primary studies were categorized on the basis of study type. Out of 52 primary studies 19 (38%) were conceptual studies, 18 (34%) simulation studies, 3 (6%) mathematical approaches, 1 (2%) survey, 7 (13%) reviews, and 4(8%) case studies. Primary studies which are used in this paper are marked with asterisk "*" in the reference list.

A. Kanban in industrial engineering

It appears the first academic paper describing Kanban was published by Japanese researchers in 1977. The title of the paper by Sugimori, Kusunoki, Cho, and Uchikawa [16] is: "Toyota Production System and Kanban System: Materialization of Just-In-Time and Respect-For-Human System." Describing Just in Time (JIT) as a central element of TPS, Sugimori, Kusunoki, Cho, and Uchikawa [16] suggest that JIT is a method whereby production lead times are greatly shortened in order to allow "all processes to produce the necessary parts at the necessary time and have on hand only the minimum stock necessary to hold the processes together". Regarding JIT, Sugimori, Kusunoki, Cho, and Uchikawa [16] consider the following three defining characteristics:

- Levelling of production
- One piece production and conveyance
- Withdrawal by subsequent processes

Further, Sugimori, Kusunoki, Cho, and Uchikawa [16] elaborate that TPS consists of two unique features. The first feature is JIT, which aims to produce only the necessary quantity products at the given time and keeping the inventory (stock at hand) at a minimum level through Kanban. The second feature is to promote respect-for-humans with the ultimate goal of uncovering workers' full potential, through active participation.

Kanban is a Japanese word meaning card or signboard [16], but it can also be a verbal instruction, a light, a flag, or even a hand signal [17][21]. According to Huang and Kusiak [26] Kanban is also known as a `pull' system in the sense that the production of the current stage depends on the demand of the subsequent stages, i.e., the preceding stage must produce only the exact quantity withdrawn by the subsequent manufacturing stage. Typically a Kanban card has information such as part name and part number, quantity designated (the size of the container) in the production process, input areas and output areas [17][18][21].

The key objective of a Kanban system is to deliver the material JIT to the manufacturing workstations, and to pass information to the preceding stage regarding what and how much to produce [26]. According to Sugimori, Kusunoki, Cho, and Uchikawa [16] reasons to use Kanban are: (1) reduction in information processing cost, (2) rapid and precise acquisition of facts, and (3) limiting surplus capacity of preceding shops or stages.

All the primary studies reported that the original Kanban used at Toyota had the following characteristics:

- Kanban system uses two types of signals. First production signals (authorizes a process to produce a fixed amount of product) and second transportation signals (authorizes transporting a fixed amount of product downstream). We use the code "PS" to denote this characteristic.
- Pulled production: The production is pulled based on the inventory level or the scheduling of the last station. We use the code "PP" to denote this characteristic.
- Decentralised control: The control of the production flow is performed through visual control by the employees of each step of the production process. We use the code "DC" to denote this characteristic.
- Limited work in progress: the inventory level is limited in each workstation, which means, buffer capacity depend on the number of signals. We use the code "LW" to denote this characteristic.

To implement original Toyota Kanban there are six principles discussed by Huang and Kusiak [26] and Sugimori, Kusunoki, Cho, and Uchikawa [16] in primary studies, which are as follows:

- Level production (balance the schedule) in order to achieve low variability of the number of parts from one time period to the next [16]. Production levelling can also be referred to reducing the waste. On a production line, as in any process fluctuations in performance can produce waste. When demand is constant production levelling is easy, but when demand fluctuates two approaches can be adopted to handle it: i) demand levelling; and ii) production

 ISBN: 978-1-61208-438-1

levelling through flexible production [26]. To prevent fluctuations in production, it is important to minimize fluctuation in the final assembly line and make production batches as small as possible.

- Avoid complex information and hierarchical control systems on the company floor which creates confusion and disturbs information flow [16]. In such situations, following the Kanban pull system becomes more complex and difficult [26].
- Do not withdraw parts without a Kanban card. Create a strict environment where the Kanban pull system is followed. Do not allow any associate within the production site to withdraw the parts without the Kanban card.
- Withdraw only the parts needed at each stage. In every stage of production withdraw only the parts which are needed for production at the given stage [16][26]. Do not include any additional parts along with the required parts in a production line.
- Do not send defective parts to the succeeding stages. Sending the defective parts to the succeeding stages will increases rework on same parts along with rejection of finished products [16][26].
- Eliminate waste due to over-production, thus produce the exact quantity of parts.

The primary studies reported a number of original Kanban usage benefits, which are as follows:

- Reduced work in progress and Cycle Time: Sugimori, Kusunoki, Cho, and Uchikawa [16] discuss the automotive industry as consisting of multi-stage processes; generally the demand for the items is unpredictable as the process point is further removed from the point of original demand for finished goods. Preceding processes require having excess capacity, and are liable to have waste by over-producing [16]. By limiting releases Kanban regulates work in progress. By Little's Law [15], this also translates into shorter manufacturing cycle times. Kouri, Salmimaa, and Vilpola [37] reported that limiting work in progress helps to consume resources efficiently at a given time. Further, Marek, Elkins, and Smith [28] explained that controlling work in progress helps reduce amount of reworks and financial losses.
- Smoother Production Flow: By dampening fluctuations in work in progress level, a steadier, more predictable output can be achieved [16][26].
- Improved Quality: Working in short queues challenges tolerance of rework because it will quickly shut down the production line [19][20][28]. Short queues reduce the time between creation and detection of a defect. Consequently, Kanban applies pressure for better quality and provides an environment in which to achieve it [30][34][35][38].
- Reduced Cost: Kanban focusses on limiting work in progress levels which eventually helps to reduce total cost [16][19][20][27][38]. Each reduction in work in progress also causes challenges (such as a setup is too long, worker breaks are uncoordinated) in the form of blocking of a line. Solving these problems by lowering the inventory allows production to proceed. This process was widely described via the analogy of lowering the water (inventory) in a river to find the rocks (problems) [16][30][35]. The end result is a more efficient system with lower costs. Kanban reduces inventory cost, inspection cost, unit product cost, and administrative cost [27]. Furthermore, according to Fearon [22] Kanban helps to identify and prioritize problems and opportunities for improvement, and enhances customer and supplier communications.

The following Table II summarizes the five variations of Kanban from primary studies, based on their similarities (in terms of characteristics) mentioned above; and differences (in terms of operationalization) from the original Toyota Kanban. An operational difference will be based on the following points:

1. Inventory variability: During production, quantity of inventory can be varied. In the original Kanban, the inventory level variation is not systematised although some maximum quantities can change between two different planning periods.
2. Physical cards are not used as signals.
3. Modification of the original concept of using two signals, i.e., production and transportation signal.
4. Visual control: Compare to original Kanban use different visual control to gather and apply information related to inventory level and demand.

TABLE II. STEPS OF THE STUDY SELECTION PROCESS.

Variations of Kanban	***Characteristics (similarities)***	***Operational difference***
Generic Kanban	PP, DC, LW	3
Generalised Kanban Control System	PP, DC, LW	3
Extended Kanban Control System	PP, DC, LW	3
Flexible Kanban System	PP, DC,	1, 4
Electronic Kanban	PP, DC, LW	2, 3, 4

Pulled production (PP), Decentralized control (DC), Limited work in progress (LW), Kanban system (PS)

Generic Kanban was proposed by Chang and Yih in 1994 for non-repetitive production environments [23][24][25]. It used generic signals which do not belong to any one part, and thus can be attributed to any item in a workstation. This system requires a waiting time since there is no arbitrator work in progress between workstations. There are signals that if removed do not initiate the production of new parts automatically, instead they wait for a new requirement [23]. Generic Kanban behaves similarly to the push system except that it is more flexible with respect to system performance and more robust as to the location of the bottleneck [23]. Simulation results showed that the generic Kanban is better than the original Kanban in dynamic environments [23]. The advantage of generic Kanban is that it can be used effectively in environments with unstable demand, as well as in productive environments with variability in processing times [23][24][25].

Generalised Kanban Control System includes the maintenance of buffers to meet the demand instantaneously, and the use of signals to authorize the production and to limit work in progress level [25][33]. The disadvantage of this is the need to define and manage two control parameters per stage, which are the buffer and the number of production order signals [25][33]. The point of difference is that in the Generalised Kanban Control System the transfer of a finished part from a given stage to the next stage and the transfer of demands to the input of this stage may be done independently of one another, whereas in the original Kanban they are done simultaneously [25]. The advantage of Generalised Kanban Control System is that it works effectively when the demand is unstable [25][33].

Extended Kanban Control System is a pulling system proposed for multi-stage manufacturing environments, which works like generalized Kanban with pull production polices, Kanban control system, and base stock control systems [29][32][34]. The difference is that, a work item can only be received in a stage if both production order and free cards are available [29][32][34]. The central idea of this system is that when the demand arrives, it is instantaneously announced to all stations, as in base stock. However, no part is made available without an authorization from the downstream stages [34]. The advantage of Extended Kanban Control System is that it works effectively in an environment with variability processing time [29][32][34].

Flexible Kanban System was introduced to cope with uncertainties and planned/unplanned interruptions [31]. It uses an algorithm to dynamically and systematically manipulate the number of signals in order to offset the blocking and starvation caused by uncertainty (mainly related to demand and processing time) during a production cycle. Gupta, Al-Turki, and Perry [31] summarises Flexible Kanban System as: The idea behind it is to increase the flow of production by judicially manipulating the number of cards. It maintains a minimal number of base level Kanban cards assigned to each station. Extra Kanban cards are added only when needed to improve the system performance and removed as soon as they are no longer needed or when their presence will result in a lowered system performance. That is, we want the extra Kanban cards when the benefits of their presence (e.g., reduced blocking and improved throughput) balance the costs (e.g., increased work in progress and operating costs) [31].

Electronic Kanban is a variation of Kanban with only one modification—the substitution of physical signals by electronic signals [36]. The goal of Electronic Kanban is to introduce an effective means for properly changing the number of cards in the Kanban system [36][37]. One of the most important things in the practical implementation of the system is properly changing the number of cards. Electronic Kanban has many advantages over the original Kanban system in reduced fluctuation and efficient change in the number of cards, faster response to demand change, and effective management of work in progress [36][37]. Electronic Kanban is resulting in improvements in supplier relationships when the systems are used outside the company, by evaluating the supplier's performance instantaneously, and guaranteeing accuracy in acquisition and transmission of amounts [36]. It can be used no matter what the distance between production and operations; it reduces the amount of the company's paperwork, reduces the probability of error associated with signals handling, reduces time to transfer and handle signals, and facilitates new product introduction [36][37].

In the literature, the above variations to Kanban are developed due to competitive industrial environments reflecting unfavourably on the use of the original Kanban system due to the need for greater variety of items, operational standardising difficulties, and demand and processing time instability. Electronic Kanban has more advantages in that it can manage parts ordering and delivery activities more efficiently and effectively than the other variants of Kanban. Additionally, it minimizes operational and logistics issues for a parts supplier or between work stations and complex flow of materials.

B. Kanban learning from industrial engineering to software development

Software development is not a manufacturing activity, because in software development every time we create something new with each development cycle, whereas manufacturing produces the same product over and over again. So, direct mapping of the manufacturing Kanban concept to software development is not logical. In software development, Kanban is used to visualise work, limit work in progress, and identify process constraints to achieve flow and focus on a single item at a given time [3]. The following Table III, compares original Toyota Kanban and software development Kanban characteristics.

TABLE III. COMPARISON OF TOYOTA KANBAN AND SOFTWARE DEVELOPMENT KANBAN CHARACTERISTICS

Kanban Characteristics	***Toyota Kanban [16]***	***Software Development Kanban [1]***
Physical	Yes	Yes
Pull	Yes	Yes
Visual	Yes	Yes
Signal	Yes	Yes
Kaizen	Yes	Yes
Limited work in progress	Yes	Yes
Continuous flow	Yes	Yes
Self-directing	Yes	Yes

 ISBN: 978-1-61208-438-1

Table III, shows that Kanban in software development has all those characteristics which are building blocks of Toyota Kanban and electronic Kanban. Kanban in software development and Toyota Kanban use physical cards to visualise work items and signal the current work in progress. The limited work in progress principle is used to control the work in progress in given time so, as to not exceed the capacity of a system or team. In software development, limiting work in progress, continuous flow, and pull characteristics are not attained by or of themselves. In software development, Kanban focuses more on enabling tasks visual and self-directing; so, as to help the team members become autonomous and improve their own process. To continuously improve the process of continuous flow and to better understand team work in progress limits, daily stand-up meetings are important in communicating information.

In distributed software development teamwork, the electronic Kanban makes it possible to visualise work of remote teams and obtain up-to-date status of projects instantaneously. For software maintenance work, Generic Kanban will work effectively because maintenance teams are dealing with a vast variety of unpredictable and critical tasks, for example, work on isolated or short-time-frame tasks which require quick responses. Generic Kanban has the advantage that it works effectively in environments with unstable demand as well as in productive environments with variability in processing times.

The characteristics of Toyota Kanban in primary studies can be translated in the software development as following:

- Each software development task is represented by a card on Kanban board. Further, these cards signalling the status of activity in the workflow.
- Pulled production: In software development it is refers to trigger the process of producing items only what the customer requested; while restricting to produce the quantity that is required and only when it is needed.
- Decentralised control: In software development, Kanban empower each team member to freely pull the items when their capacity allows. Additionally, everyone is trying to maintain the flow.
- Limited work in progress: In each phase of software development (i.e., development, testing) a limit on work in progress is applied which shows the capacity and signal when it is full or ready to pull upcoming items.

In software development, we can implement the original Toyota Kanban discussed by Huang and Kusiak [26] and Sugimori, Kusunoki, Cho and Uchikawa [16] as follows:

- Kanban is a visual management tool that shares a mental model; visualise the work, the workflow, and the business risks to the whole team and or organisation [1]. According to Al-Baik and Miller [41] Kanban helps in enhancing visual control that facilitated and supported the decision-making process.
- To implement Kanban in software development is to start with what you do now [1][41]. Avoid complex information and hierarchical structure in assigning the tasks. Use the Kanban board and allow the development team to pull the tasks automatically. Further, too much controlling of task assignment should be avoided because it creates confusion, disturb information flow and makes difficult to use pull technique [1][11][41].
- Carefully apply work in progress limits on each stage of software development (i.e., development, testing) [1][41]. Limited work in progress is the means by which we can create a pull system which balances capacity and demand through the value stream [16]. Further, it implies that pull system is implemented to the workflow. In software development, this means that upstream work can be made available, but it is the team member’s responsibility to decide when to take it. The act of pulling the work is a signal for more upstream work to be processed.
- A card needs to be assigned to every customer request and should be visualised on a Kanban board [1][41]. Do not withdraw tasks without a Kanban card. Create a strict environment where the Kanban pull system is followed. Do not allow any team member within the development team to withdraw the tasks without the Kanban card.
- Pull out only the tasks that are of high priority [1][41].
- Do not send partially done tasks. By sending to succeeding stages will increases rework on the same task along with rejection of finished products [1][11][41].
- Eliminate waste due to over-producing by working on the exact tasks which are needed or requested by the customer [1][41].

IV. CONCLUSION

Kanban is a subsystem of the Toyota Production System (TPS), which was created to control inventory levels, production, and supply of components. Kanban entered software development in 2004, when David Anderson introduced it in practice while assisting a software development team at Microsoft.

To learn from industrial engineering examples of Kanban usage, systematic review and analysis of Kanban variations, benefits, and implementation principles was conducted. Searches of the literature identified 1552 studies of which 53 were found to be studies of acceptable credibility and relevance. Most of the primary studies (72%) are conceptual or use simulation techniques to investigate Kanban in industrial engineering.

 ISBN: 978-1-61208-438-1

This study reported the original Toyota Kanban and its five variations from industrial engineering literature. Toyota Kanban which is also adopted in software engineering has the four basic characteristics: pulled production, decentralized control, limited work in progress, and two types of signals (i.e., production signals and transportation signals). The Toyota Kanban and its five variations have similar characteristics fundamentally concerned with signals use and manipulation in terms of number or quantity. For example, electronic Kanban has one modification—the substitution of physical signals by electronic signals.

In software development work, the Toyota Kanban concept is adopted. For distributed software development, electronic Kanban is more suitable. Whereas, based on the Generic Kanban characteristics, it is judged to be more effective for work which has unstable demands and variability in processing times (i.e., software maintenance and support). In maintenance work, customers' requests come on a daily basis, and tasks are constantly prioritised based on severity. The advantage of Generic Kanban is that it works effectively when the demand is unstable. In software development it is used to achieve better process control (keeping continuous flow while limiting work in progress) and better process improvement (makes the flow visible and stimulates Kaizen). Kanban in both industrial engineering and software engineering yields benefits such as smoother production or development flow, reduced cycle time, and improved quality.

Future studies are needed to explore Kanban variation limitations, disadvantages, and challenges in their usage. Further, it is recommended to conduct detailed comparative studies on Kanban variations along with the Kanban used in software development.

ACKNOWLEDGMENT

This research was carried out within the DIGILE Need for Speed program, and partially funded by Tekes (the Finnish Funding Agency for Technology and Innovation) and a Nokia foundation scholarship.

REFERENCES

[1] D. J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business. Blue Hole Press, 2010.

[2] M. O. Ahmad, J. Markkula, and M. Oivo, "Kanban in software development: A systematic literature review". Proceedings of the 39th Euromicro Conference Series on Software Engineering and Advanced Applications (SEAA 2013), pp. 9-16.

[3] M. O. Ahmad, J. Markkula, and M. Oivo, and P. Kuvaja, "Usage of Kanban in Software Companies - An empirical study on motivation, benefits and challenges". Proceeding of the 9th International Conference on Software Engineering Advances (2014). pp, 150-155.

[4] G. Concas, M. I. Lunesu, M. Marchesi and H. Zhang, "Simulation of software maintenance process, with and without a work-in-process limit," Journal of Software: Evolution and Process (2013), vol. 25, pp. 1225-1248.

[5] D. I. Sjøberg, A. Johnsen and J. Solberg, "Quantifying the effect of using Kanban versus Scrum: A case study," Software, IEEE, (2012), vol. 29, pp. 47-53.

[6] J. Prochazka, "Agile support and maintenance of IT services," in Information Systems Development Anonymous Springer, (2011), pp. 597-609.

[7] X. Wang, K. Conboy and O. Cawley, "“Leagile” software development: An experience report analysis of the application of lean approaches in agile software development," J. Syst. Software, (2012) vol.85, pp.1287-1299.

[8] E. Corona, and F.E. Pani, "A review of lean-kanban approaches in the software development". WSEAS Transactions on Information Science and Applications, (2013), vol. 10(1), pp. 1-13.

[9] N. Nikitina, and M. Kajko-Mattsson,"Developer-driven big-bang process transition from scrum to kanban". Proceedings of the International Conference on Software and Systems Process, (2011), pp. 159-168.

[10] B. Kitchenham, and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software Engineering" EBSE Technical Report, EBSE-2007-01.

[11] M. Burrows, Kanban from the inside. Blue Hole Press, Sequim, Washington. 2014.

[12] D. S. Cruzes, and T. Dyba, Research synthesis in software engineering: A tertiary study', Information and Software Technology, (2011) vol. 53, no. 5, pp. 440–455.

[13] Bennet Vallet (2014). Kanban at Scale – A Siemens Success Story, InfoQ. [Online] Available from: http://www.infoq.com/articles/kanban-siemens-health-services 2015.07.03

[14] C.S. Kumar, and R. Panneerselvam, Literature review of JIT-KANBAN system. The International Journal of Advanced Manufacturing Technology, (2007), 32(3-4), pp. 393-408.

[15] J. D. Little, and S.C. Graves, Little's law. In Building Intuition Springer US. (2008), pp. 81-100.

[16] Y. Sugimori, K. Kusunoki, F. Cho, and S. Uchikawa, "Toyota production system and kanban system materialization of just-in-time and respect-for-human system", The International Journal of Production Research, (1977), vol. 15(6), pp. 553-564. *

[17] O. Kimura, and H. Terada, "Design and analysis of pull system, a method of multi-stage production control", The International Journal of Production Research, (1981), vol. 19(3), pp. 241-253. *

[18] G.R. Bitran, and L. Chang, "A mathematical programming approach to a deterministic kanban system". Management Science, (1987), vol. 33(4), pp. 427-441. *

[19] H. Dyck, R.A. Johnson, and J. Varzandeh, "Transforming a traditional manufacturing system into a just-in-time system with kanban". Proceedings of the 20th Conference on Winter Simulation, (1988), pp. 616-623. *

[20] M. Gravel, and W.L Price, "Using the kanban in a job shop environment". The International Journal of Production Research, (1988), vol. 26(6), pp. 1105-1118. *

[21] Jr. Esparrago and R. A, "Kanban. Production and Inventory" Management Journal, (1988), vol. 29(1), pp. 6-10. *

[22] P.A. Fearon, "Inventory controlled environment (I.C.E.) just-in-time at national semiconductor". Proceedings of the Advanced Semiconductor Manufacturing Conference and Workshop, (1993), pp. 34-38. *

[23] T. Chang and Y. Yih, "Generic kanban systems for dynamic environments". The International Journal of Production Research, (1994), vol. 32(4), pp. 889-902. *

[24] T. Chang and Y. Yih, "Determining the number of kanbans and lotsizes in a generic kanban system: A simulated annealing approach". The International Journal of Production Research, (1994), vol. 32(8). *

[25] Y. Frein, M. Di Mascolo, and Y. Dallery, "On the design of generalized kanban control systems". International Journal of Operations and Production Management, (1995), vol. 15(9), pp. 158-184. *

[26] C. Huang, and A. Kusiak, "Overview of kanban systems". International journal of computer integrated manufacturing, (1996), col. 9(3), pp. 169-189. *

[27] L. Sriparavastu and T. Gupta, "An empirical study of just-in-time and total quality management principles implementation in manufacturing firms in the USA". International Journal of Operations and Production Management, (1997), vol. 17(12), pp. 1215-1232. *

[28] R.P. Marek, D.A. Elkins, and D.R. Smith, "Manufacturing controls: Understanding the fundamentals of kanban and CONWIP pull systems using simulation". Proceedings of the 33nd Conference on Winter Simulation, (2001), pp. 921-929. *

[29] J. Bollon, M. Di Mascolo, and Y. Frein, "Unified framework for describing and comparing the dynamics of pull control policies". Annals of Operations Research, (2004), vol. 125(1-4), pp. 21-45. *

[30] E. Kizilkaya, and S.M. Gupta, "Material flow control and scheduling in a disassembly environment". Computers and Industrial Engineering, (1998), vol. 35(1), pp. 93-96. *
[31] S.M. Gupta, Y.A. Al-Turki, and R.F. Perry, "Flexible kanban system". International Journal of Operations and Production Management, (1999), vol. 19(10), pp. 1065-1093. *
[32] G. Liberopoulos, and Y. Dallery, "A unified framework for pull control mechanisms in multi-stage manufacturing systems". Annals of Operations Research, (2000), vol. 93(1-4), pp. 325-355. *
[33] C. Duri, Y. Frein, and M. Di Mascolo, "Comparison among three pull control policies: Kanban, base stock, and generalized kanban". Annals of Operations Research, (2000), vol. 93(1-4), pp. 41-69. *
[34] Y. Dallery, and G. Liberopoulos, "Extended kanban control system: Combining kanban and base stock. IIE Transactions, (2000), vol. 32(4), pp. 369-386. *
[35] K. Takahashi, and N. Nakamura, "Decentralized reactive kanban system". European Journal of Operational Research, (2002), vo1. 39(2), pp. 262-276. *
[36] S. Kotani, "Optimal method for changing the number of kanbans in the e-kanban system and its applications". International Journal of Production Research, (2007), vol. 45(24), pp. 5789-5809. *
[37] I. Kouri, T. Salmimaa, and I. Vilpola, "The principles and planning process of an electronic kanban system". Novel algorithms and techniques in telecommunications, automation and industrial electronics, (2008), pp. 99-104. *
[38] M. Akturk, and F. Erhun, "An overview of design and operational issues of kanban systems". International Journal of Production Research, (1999), vol. 37(17), pp. 3859-3881. *
[39] J.K. Liker, M. Hoseus, "Toyota culture: the heart and soul of the Toyota Way". (2008), McGraw-Hill, New York, USA
[40] O, Taiichi. "Toyota production system: beyond large-scale production". (1988), Productivity press.
[41] O. Al-Baik and J. Miller, "The kanban approach, between agility and leanness: a systematic review" Empirical Software Engineering. (2014), pp. 1-37.