Journal of Applied Economics and Business

OPEN ACCESS ARTICLE

USAGE OF KANBAN METHODOLOGY AT SOFTWARE DEVELOPMENT TEAMS

Nevenka Kirovska[1], Saso Koceski[2]

Faculty of Computer Science, University Goce Delchev, Stip, Macedonia

[1]nevenka.21081@student.ugd.edu.mk, [2]saso.koceski@ugd.edu.mk

Abstract

The purpose of this paper is to present the Kanban methodology and its practical usage within a software development environment. Kanban is primary a lean manufacturing concept and its application in other areas is continuously growing due to its proven successfulness. Applying Kanban in software development environments is yet young but increasingly growing concept that is becoming more and more popular. Traditional methodologies are going down on the popularity scale as new methods and practices come on board. Agile methods are proven to be very successful in today's modern software development. Usage of Kanban in software development teams brings many advantages and improvements. Kanban methodology also offers specific metrics for monitoring and studying in stochastic terms which means increased effectiveness and collaboration. The practical implementation of this concept will be conducted by using the Kanban web based application called KanbanMAK within an IT company. The results of this implementation showed increased effectiveness, better collaboration and overall improvement in the software development process.

Key words:

Kanban Methodology; Agile Methodology; Software Development; Lean.

INTRODUCTION

In order to understand the usage of Kanban in a software development process we must first explore its original usage. The Kanban concept was originally a part of the JIT (Just-In-Time) production system of Toyota in the 1950s. "Just-In-Time" means making "only what is needed, when it is needed, and in the amount needed." Kanban roughly means signal card that is used to trigger manufacturing action. At Toyota, when a process refers to a preceding process to retrieve parts, it uses a kanban to communicate which parts have been used (Taiichi, 1988).

Taiichi Ohno is the name that is behind this concept and in his book, Toyota Production System, he highlights the JIT concept and automation with a human touch (autonomation) as the two pillars of the Toyota Production System with Kanban being the tool used to operate the system (Taiichi, 1988). The Kanban system has also been called the "Supermarket method" because Taiichi Ohno's inspiration came from the American grocery stores.

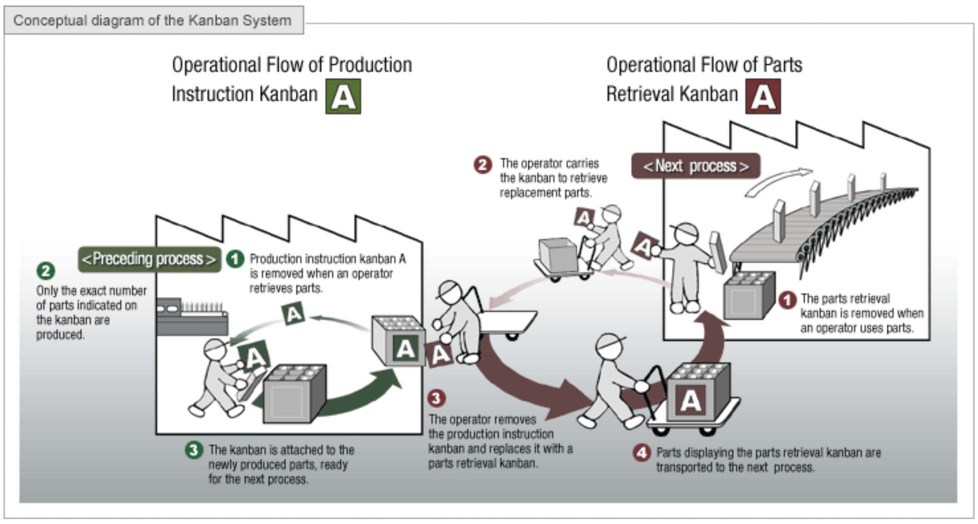
Conceptual diagram of the Kanban System
Operational Flow of Production Instruction Kanban A
Operational Flow of Parts Retrieval Kanban A
<Preceding process>
1 Production instruction kanban A is removed when an operator retrieves parts.
2 Only the exact number of parts indicated on the kanban are produced.
3 The kanban is attached to the newly produced parts, ready for the next process.
<Next process>
1 The parts retrieval kanban is removed when an operator uses parts.
2 The operator carries the kanban to retrieve replacement parts.
3 The operator removes the production instruction kanban and replaces it with a parts retrieval kanban.
4 Parts displaying the parts retrieval kanban are transported to the next process.

FIG. 1 CONCEPTUAL DIAGRAM OF THE KANBAN SYSTEM

The Figure 1 shows the conceptual diagram of the Kanban system. Here we have two kinds of Kanban in a meaning of visual signals (Production Instruction and Parts Retrieval Kanban). These two kinds of Kanban are being used for managing parts in the production process (Toyota Global, Web page, August 2015).

Kanban is not just a concept related to JIT production. It is also a *lean* technique. The term lean was originally mentioned in the book "The machine that changed the world" from the authors J. P. Womack and D. T. Jones briefly meaning *elimination of waste*. The authors used the term as a synonym for the Toyota's automotive industry. Furthermore, in software engineering, this term was originally used in the book "Lean Software Development" (Poppendieck & Poppendieck, 2003). In software engineering the principles that this concept is based on are similar to the one in the production and they are:

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build quality in
7. See the whole

The beginnings of Kanban involvement in software development are connected to David J. Anderson (Anderson, 2010) when he was invited by Microsoft to assist in one of their small teams for achieving better visualisation of the working flow and limiting the workflow.

This paper goal is to present the usage of Kanban in the process of software development and it is structured in this order: section 2 describes the comparison between traditional software development methods on one side and Kanban on the other, section 3 presents the implementation of Kanban in software development and the benefits from it and the last section describes the results gained from evaluating that implementation.

TRADITIONAL SOFTWARE DEVELOPMENT METHODS AND AGILE METHODS

In the last decade, there have been many studies, researches and publications about agile software development methodologies (ASDM). The popularity of these methodologies come from the disadvantages and shortage of traditional software development methods (TSDM) in today`s dynamics and fast changing environment. The formal origin of this agile methodologies date from February 2001, where a group of experts (Agile Alliance) sat down and discussed the problems and disadvantages of the current software development methodologies. The result of that meeting was a document containing the principles and basics of agile methodologies called *Agile Manifesto* (Beck et al, 2001). The messages resulting from this document are:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

According to Beck et al, (2001) the Agile movement is not anti-methodology, but a way to restore credibility to the word methodology, a chance to restore a balance. They embrace modelling, but not in order to file some diagram in a dusty corporate repository. They also embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes. Planning is conducted, but recognition of the limits of planning in a turbulent environment is achieved.

Agile methods are a subset of iterative and evolutionary methods (Larman & Basili, 2003). They are based on the same agile principles but the differences among them come from the different aspects of the software development life cycle they focus. Popular agile methods include: XP, Scrum, Kanban, Dynamic System Development

Method (DSDM), Adaptive Software Development (ASD), Feature Driven Development (FDD), Unified Process (UP), Crystal, etc. Agile methods focus on the code rather than the design. They are based on an iterative approach for software development and are intended to deliver working software quickly and evolve this quickly to meet changing requirements. The aim of agile methods is to reduce overheads in the software process (e.g. by limiting documentation) and to be able to respond quickly to changing requirements without excessive rework.

Many studies have been made comparing ASDM and TSDM with an emphasis on the advantages and disadvantages that came from ASDM implementation. Apart from taking references and reviews, Bhadoriya et al, (2014) conducted a research in a software firm that uses ASDM (Specifically Extreme Programming) in their software development work. The research aim was to find the benefits and difficulties in the transition from traditional to agile software development and the changes from both the team and customer's point of view. The benefits reported were: quick adoption of clients changing requirements, saved time and money for not following the whole systems development life cycle (SDLC), improved and reviewed software in every phase of development because of the customer involvement and so on. Beside the benefits, there were few drawbacks reported by team members. Inconsistency of requirements makes difficulties in project scheduling. As the project requirements change it is absolutely necessary to have experienced and senior resources in the team or at least experienced resources mentoring new resources. Bhadoriya et al, (2014) also reported that team members had decreased time for software development which is some way increasing their programming ability to deliver software properly in time.

Even though agile methodologies are used for small and medium-sized projects, there are also studies and researches that analyse the implementation of ASDM in large projects. Papadopoulos (2015) conducted a case study about adopting the agile framework on large, distributed projects and reported improved quality and employee satisfaction while building the end product, requirement changes and improved better performance of the agile software development than the traditional methodologies. The case study also demonstrated that adopting the agile framework is not straightforward and large companies with a long history of using traditional processes need to carefully plan this activity, trying to avoid common problems observed when attempting to adopt the agile methodologies.

APPLYING KANBAN IN SOFTWARE DEVELOPMENT

Software development is not a production or a manufacturing activity (Reeves, 1993). Software engineers create different things every time, whereas manufacturing produces same things over and over again. When we talk about a software development using Kanban we are not using the term signal cards for "pulling" work but instead we are talking about *work items.* The term *virtual* cards can also be used because here we don't have a physical card that is passed on to mark limits in the Work-In-Progress.

The formal beginnings of Kanban involvement in software development are connected to David J. Anderson (Anderson, 2010) when he was invited by Microsoft to assist in one of their small teams for achieving better visualisation of the working flow and limiting the workflow. The result turned out to be five successfully proven principles for Kanban implementation. The process of successful adoption starts from adopting the basic foundation principles:

- Start with what you have now – that is your current process.
- Agree to pursue an evolutionary approach to change and improvement
- Respect the current roles and responsibilities of the team/ organization

Based on these principles, the next thing to implement are the five core Kanban principles (Anderson, 2010) which were commonly observed in organizations that experienced success in using the Kanban method.

1. Visualize the Work and the workflow that it follows
2. Limit Work-In-Progress (WIP) using a virtual Kanban system
3. Manage Flow
4. Make Management Policies Explicit
5. Use Models and the scientific method and Improve collaboratively

The term "virtual Kanban board" was used by Anderson to make the Work-In-Progress visible while identifying constrains and limiting the Work-In-Progress to one item. In traditional software development, the work was "pushed" over the software development line while here everyone has only one task at a time and the work is "pulled" after it is finished.

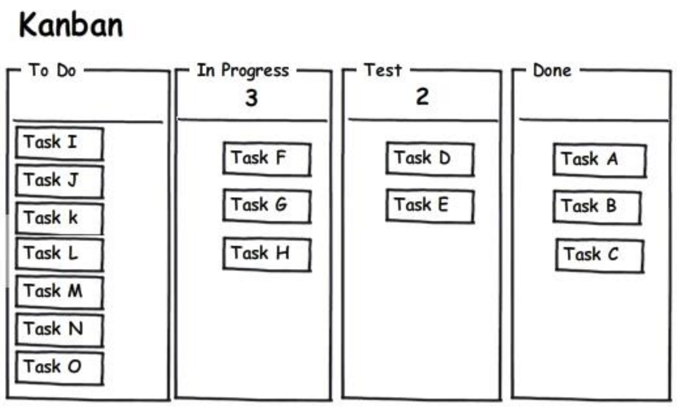

FIG. 2 EXAMPLE OF A KANBAN BOARD

Ahmad et al, (2013) reported lack of scientific researching's regarding Kanban usage in software engineering. Despite that deficiency, their literature review revealed findings about motivation factors for Kanban adoption, benefits and challenges. Even though practices, benefits and challenges were presented, there were no studies found regarding detailed and fully explained usage of Kanban in software development.

Ahmad et al, (2014) reported improving team communication and development flow, reducing time to reach the market, increasing productivity and creating transparency in organisation as main motivation factors for Kanban adoption. The empirical study was conducted on the current state of Kanban usage in software companies which extensively use Agile and Lean approaches. The authors used survey and thematic interviews to analyse the motivations of Kanban usage, the benefits obtained from it and faced challenges. The most common achieved benefits of using Kanban were better visibility of work, improved transparency and communication and better control of flow. Along with the benefits, a few challenges were faced. The most common one was lack of experience with Kanban. That means difficulties in managing the WIP limits and prioritization of tasks. The next challenge was about the traditional culture of the organization and how it affected adoption of Kanban.

Usage and evaluation of KanbanMAK web application

KanbanMAK is a custom web-based application. The application was presented and then tested by the software development team members. The application was tested on medium-sized projects with 1-10 team members. The application test period usage lasted for 14 months and included around 23 projects. After the implementation, the evaluation phase started.

The main goal of this analysis was to examine the real usage of the application and the benefits from it. Two methodological approaches were used, including:

- Automatic database analysis and
- Web-surveys with employees.

Analysis results and conclusions

The conducted web survey included 45 team members from which 88,8% gave their answers. The results from this survey showed that there are significant improvements of the daily development process, with increased collaboration and better estimation of working tasks execution. The following graphs show some of the results presented.

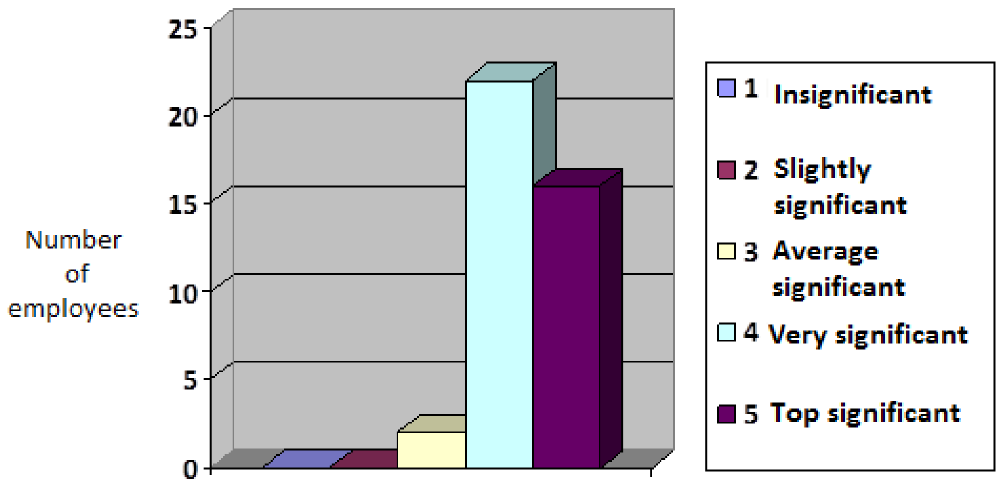

FIG. 3 LEVELS OF IMPROVEMENTS USING KANBANMAK APPLICATION

Figure 3 shows that more than half of the employees gave a rating 4 out of 5 for improvements after the usage of the application. The highest rate was given by 40% of them which is also a satisfactory result.

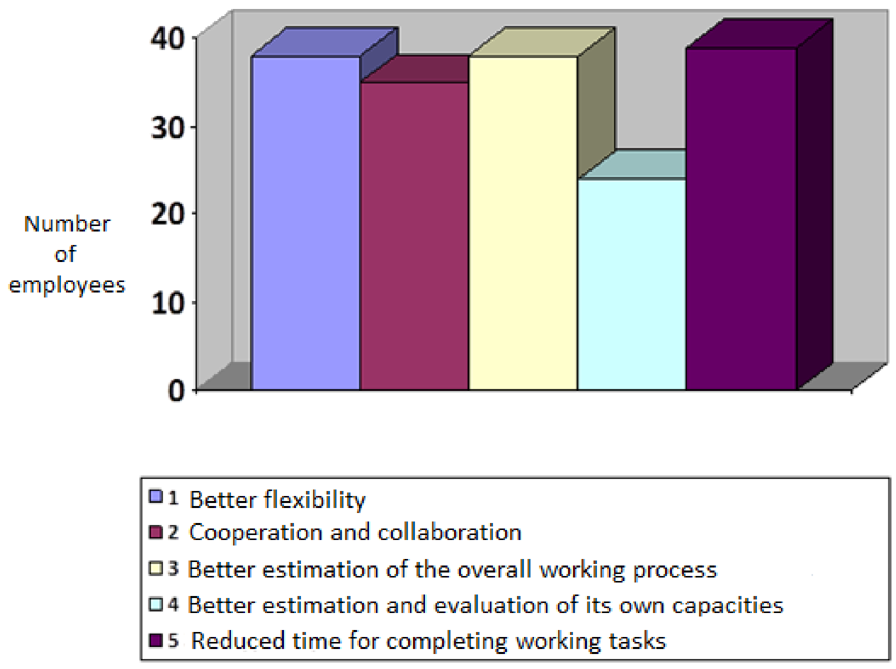

FIG. 4 ADVANTAGES REPORTED WHILE USING KANBANMAK APPLICATION

Most of the proposed advantages were chosen by the employees as gained benefit after using the Kanban web-based application presented in the graph on figure 4. These advantages included: reducing time for completing working tasks, better flexibility, cooperation and collaboration and similar.

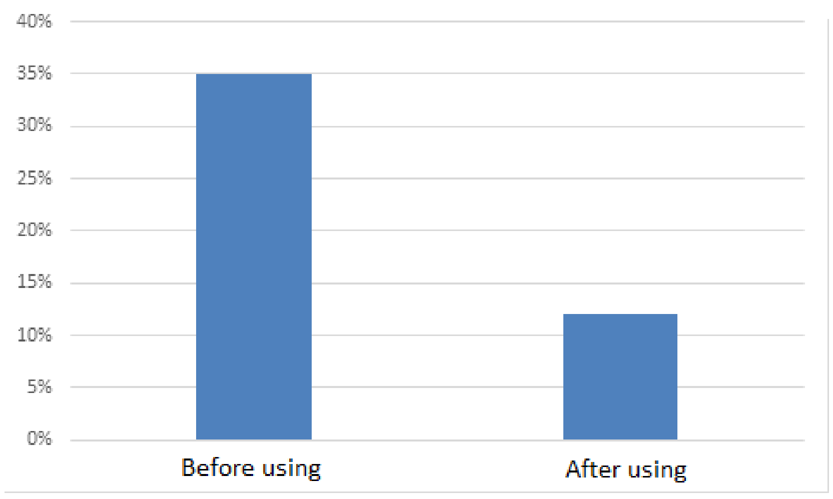

FIG. 5 DIFFERENCE BETWEEN ESTIMATED AND REAL TIME OF WORKING TASKS IN MEDIUM-SIZED PROJECTS

Figure 5 shows a 23% decreasement of the difference between the estimated and real time of working tasks. This means better predictability for estimated delivery and overall customer satisfaction.

Beside the reported advantages and benefits from using the application, there were disagreements and disbelieves for including the new methodology into the daily management processes. There were also employees who didn't wanted to participate in the web survey and considered it to be "too fresh to adopt". This is an interesting fact because it is indeed a new and fresh concept and disagreement in something other than the known traditional approach can be sometimes justified. But all team members should have in mind that everything in the IT world is changing rapidly and they must be prepared to adopt quickly the new challenges that come due to fast changing technical environment.

Taking into consideration all of the results from the survey conducted using the KanbanMAK application and the literature overview, we can list the following advantages:

- Kanban implementation has a few principles and rule which makes it easy to fulfil.
- Applying Kanban in software development means adoption of practices and principle into the existing process for improvement.
- Visualization is the key for identifying bottlenecks and a way to monitor the current state of development process.
- Using Kanban methodology encourages the team to further upgrade and motivate.
- Final development results increases customer trust and satisfaction.

- Improvements in product quality, expenses are decreased and time for delivery is reduced.

One of the most important things about the Kanban methodology is that this methodology is applied to our existing process. This methodology does not create a new way of managing the existing process but find ways to improve the existing process. That is why Kanban is an evolutionary method that promotes gradual improvements to managing processes including development.

CONCLUSIONS

Being originally a lean manufacturing concept, Kanban has gained a great popularity as software development methodology. The traditional, heavyweight methodologies are being substituted by the lightweight agile methodologies as a result of the many advantages they offer.

Each software development project is story for itself and there is no such thing as a perfect single model for all types of software projects. All agile methods have their advantages and disadvantages. What distinguishes Kanban methodology from all the other agile methodologies is the minor changes made when it is applied in the current process. Kanban methodology also offers specific metrics for monitoring and studying in stochastic terms. Predicting the development process increases effectiveness and collaboration within the team.

Beside the reported advantages and benefits from the surveys made, there were disagreements and disbelieves for including the new methodology into the daily management processes. Kanban involvement in software development is still new and fresh concept and there is and will be a need for better understatement and changes in traditionalism for faster adoption.

REFERENCES

Anderson, D. (2010). Kanban – Successful Evolutionary Change for Your Technology Business. Blue Hole Press.

Ahmad, M. O., Markkula, J. & Oivo, M. (2013). Kanban in software development: A systematic literature review. In Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on (9-16). IEEE.

Ahmad, M. O., Markkula, J., Oivo, M. & Kuvaja, P. (2014). Usage of Kanban in software companies: An empirical study on motivation, benefits and challenges (ICSEA), 2014: The Ninth International Conference on Software Engineering Advances.

Bhadoriya, N., Mishra, N. & Malviya, A. (2014). Agile Software Development Methods, Comparison with Traditional Methods & Implementation in Software Firm.

In International Journal of Engineering Research and Technology, 3(7), ESRSA Publications.

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Humt, A., Je®ries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thom, D.: Manifesto for agile software development. Website (2001) http://agilemanifesto.org/.

Larman, C. & Basili, V. (2003). A History of Iterative and Incremental Development, IEEE Computer, 36(6), 47-56.

Papadopoulos, G. (2015). Moving from Traditional to Agile Software Development Methodologies Also on Large, Distributed Projects. Procedia-Social and Behavioural Sciences, 175, 455-463.

Poppendieck, M. & Poppendieck, T. (2003). Lean software development. Boston, Mass.: Addison-Wesley. Taiichi O. Toyota Production System: Beyond Large - Scale Production.

Jack W. Reeves. (1992). What is Software Design? C++ Journal, My favorite article about Software Design Toyota Global (Web page) http://toyota-global.com/ (last access August 2015).