SOFTWARE DEVELOPMENT FOR RTOS AIDED BY KANBAN METHOD

Alexandre, T. M.; Almeida Jr, J. R.
Universidade de São Paulo, Departamento de Engenharia de Computação e Sistemas Digitais

*** Corresponding author e-mail address**: t.alexandre@gmail.com

PAPER ID: SIT146

ABSTRACT

The purpose of this paper is to promote the usage of the Kanban method for the development of RTOS (real-time operating system) usually used in avionics, nautical and automobile industries. The paper takes the form of a literature study. Kanban method has been aiding the development of software in several industries such as healthcare, financial, communication manufacturing and retail. There is an enormous potential and opportunity in ROTS industry. All the Kanban benefits observed in other industries may be discovered in RTOS development process as well.

Keywords: Kanban, RTOS, software development.

1. INTRODUCTION

The Kanban method applied to software development has been highlighted in the literature since 2010 with some studies and experience reports being published. In mid-2013, it was still not possible to reliably determine its impacts, however, the method was already demonstrating to deliver customer satisfaction, reduction of reported defects, improved communication, improved visualization of work in progress and some recommendations such as: introducing the method incrementally, accompanying the board with daily meetings and providing appropriate training to all those involved. (Ahmad, Markkula, & Oivo, 2013)

More recently, new studies and experience reports have endorsed these impressions about the method, in addition to listing more recommendations on how to always apply the Kanban board with other complementary tools such as cumulative flow diagrams, metrics report and other graphs. (Ahmad, Dennehy, Conboy, & Oivo, 2018)

In RTOS (*real-time operating system*), Kanban has been applied to eliminate waste by improving the visualization of the process as it highlights the bottlenecks of the production or development system. (Simić, et al., 2021) Many embedded real-time systems are necessary for aeronautics that, from the 1990s, adopted the BFP (*better, faster, cheaper*) paradigm and the Lean to produce your products. (Murman, Walton, & Rebentisch, 2000)

This article is structured as follows: Section 2 presents an overview of Kanban as a software development method. The growth and use of Kanban in the software industry is described in section 3. Some examples of Kanban usage in RTOS are listed in section 4, and finally, in section 5, the conclusion and future work are presented.

2. KANBAN AS A SOFTWARE DEVELOPMENT METHOD

The Kanban method, although it has established principles that characterize it, still needs a unification in relation to the construction of knowledge in the scientific environment. Non-peer-reviewed publications, books and guides vie for attention in the market and experience reports need more rigor in terms of how they address the threats and risks of the method. (Ahmad, Dennehy, Conboy, & Oivo, 2018)

The first publication on the introduction of Lean in software development is attributed to (Poppendieck & Poppendieck, 2003). The Kanban method, proposed by (Anderson D. J., 2010), derives from *Lean* and incorporates other elements

2.1. DEFINITION

Kanban is an evolutionary project tracking method that uses a kanban, (lowercase k), and other tools to visualize a pull system to apply *Lean* ideas to the technology development process as well as information technology operations. (Anderson D. J., 2010)

At its origin, Kanban was based on 3 principles: 1. Visualize the workflow; 2. Limit work in progress; 3. Manage the flow. (Hammarberg & Joakim, 2014)

Visualizing the workflow allows to identify bottlenecks and propose solutions. These proposals lead to small improvement actions that can be continuously implemented and tested.

Limiting work in progress leads to balance the amount of work against the amount of contributors to improve the delivery rate. In this principle, (Anderson D. J., 2010) relied on Little's Law of the simplified queuing theory of (Leon-Garcia, 2008):

$$Q_i = \lambda_i R_i \quad (1)$$

Where Q_i is the average number of entities in the queue, λ_i is the arrival rate and R_i is the average response time.

Translating to the context of software development:

Work in progress = task completion rate *x* average development time.

The average development time is a difficult variable to change because it involves several factors such as the number of people on the team, the capacity of these people, availability, and other productivity factors.

Thus, from (1), the simplest way to increase the rate of task completion is to reduce work in progress.

Flow management consists of keeping activities that deliver value to achieve the customer's purpose in progress without impediments or queues. Anything that does not generate value in the eyes of the customer should be considered as waste. Daily meetings, reassigning more people to the same task that is more technically difficult, dividing complex tasks into tasks that are simpler to understand and faster to execute are some activities related to flow management. (Hammarberg & Joakim, 2014)

2.2. IMPLEMENTATION

One of the ways to implement Kanban is to apply eight steps not necessarily chronological proposed (Anderson D. J., 2010) and detailed by (Burrows, 2014) called *STATIK* - Systems Thinking Approach to Introducing Kanban. The steps are: 1. Adapt the service to the client's purpose; 2. Understand the sources of dissatisfaction with the current system; 3. Analyze demand; 4. Analyze capacity; 5. Model the workflow; 6. Discover the classes of service; 7. Design the kanban system; 8. Publicize and negotiate the implementation of the method with the stakeholders. Each of these steps is detailed in the following sections.

2.3. MAKE THE SERVICE FIT FOR PURPOSE FOR THE CUSTOMER

This step refers to exploring all the satisfaction criteria that are important to the customer. Some criteria to be followed are quality, predictability of deliveries, security, service level agreements, regulatory issues, among others. The clearer the goals and needs of the client are, the greater the chances of success and delivery according to their expectations. (Anderson & Zheglov, 2017) propose two ways to know the client's purpose: 1. Through narratives; 2. Through research and data. The first approach relies on the human ability to tell and interpret stories. The second contains data for measurement and analysis. These approaches are complementary and help to understand the client's purpose and quality criteria for each type of service.

2.4. UNDERSTANDING THE CURRENT SYSTEM’S SOURCES OF DISSATISFACTION

At this stage, dissatisfaction can be observed, both internally (developers, quality team, support team) and externally (customers, suppliers). A technique for collecting dissatisfaction described by (Burrows, 2014) consists of the following steps: 1. Individuals write down their dissatisfaction with a short sentence on a small self-adhesive paper anonymously; 2. The papers are placed on a board and grouped according to their similarity in small groups; 3. The groups are named so as to logically represent the dissatisfaction they represent; 4. Each individual receives 3 to 5 votes to distribute in the cluster. With this technique, individuals can better reflect on the problems they face and begin to propose suggestions for process improvement. From experience, (Burrows, 2014) warns that some dissatisfactions can hide others. For example: Dissatisfaction about lack of time can hide communication, coordination, or quality problems. At this stage, it is recommended to explore needs and not assume hasty solutions.

2.5. ANALYZE CAPABILITY AND DEMAND

In this stage, demand analysis is carried out, which consists of collecting information about each type of task, volume, and arrival rate of each item and understand these values and how important they are to the customer. As for capacity, collect historical data, if available, on delivery times, time for each task, quality, predictability, and regulatory requirements. (Senapathi & Drury-Grogan, 2021)

This analysis should generate quantitative and qualitative data. If possible, use graphs such as *Lead Time*, shown in Figure 1 and *CFDs* - cumulative flow diagram.

.

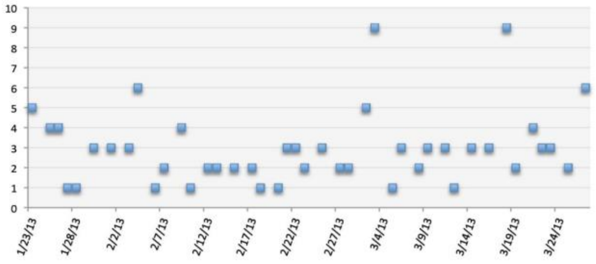

Figure 1 Lead Time. Source: (Burrows, 2014)

2.6. MODEL WORKFLOW

A common approach to this activity is the top-down decomposition. In this approach, 3 parts are chosen that divide the process: "to do", "in progress", "done" and then begins the decomposition of each part in details of the process until a table is built where each column represents a stage of the development process. (Burrows, 2014) The columns in Figure 2 exemplify the breakdown of these parts in some activities.

to do		in progress			done	
backlog	selected	under development	ready to test	in test	waiting deploy	deployed

Figure 2 Model workflow. Source: Adapted from (Anderson D. J., 2010)

2.7. DISCOVER CLASSES OF SERVICE

The Kanban method uses the concept of service classes that are defined based on the business risk that the tasks have. In particular, the cost of delay. (Anderson & Bozheva, 2021) Which classes to determine is a particular design decision. However, there are four main categories that can be cited: 1. Urgent or Priority: tasks that need immediate attention; 2. Fixed date: tasks that have a date that, if not fulfilled, will imply a strong penalty; 3. Intangibles: Tasks with no attached business value, but with medium to long-term value potential; 4. Standard or common: The remaining tasks prioritized according to the value they bring to the customer. (Anderson & Bozheva, 2021)

2.8. DESIGN THE KANBAN SYSTEM

The Kanban board design considers the service classes as horizontal demarcations as shown in Figure 3 and the development steps as columns

Project A	to do		in progress			done	
	backlog	prioritized	under develop-ment	ready to test	in test	waiting deploy	deployed
wip limits >>		3	3	3	2	4	-
Urgent							
Standard							

Figure 3 Kanban board. Source: Adapted from (Anderson D. J., 2010)

The limitation of work in progress (WIP) needs to be imposed through some mechanism (Burrows, 2014) as it is part of one of the principles of the method. The highlighted line in Figure 3 is an example of a work-in-progress control mechanism. The numbers indicate the number of tasks, represented by yellow cards, that can exist in each column that represents a step in the process. So, to pull a new task from the "prioritized" list, it is necessary that the number of tasks under "in development" column be less than three and so on in all fields with constrains.

After a few weeks of follow-up, it will be possible to extract the cumulative flow diagram (CFD).

The CFD is useful for viewing a series of information relevant to the progress of the project, serving as a background for discussions of process improvements, being the most used diagram, making it possible to identify WIP, total development time (lead time) and bottlenecks. in the flow. (Corona & Pani, 2013) The analysis of this graph shows the relationship between WIP and lead time, that is, the throughput rate. In the graph of Figure 4, the slope is the throughput. The height is WIP, and the length is lead Time.

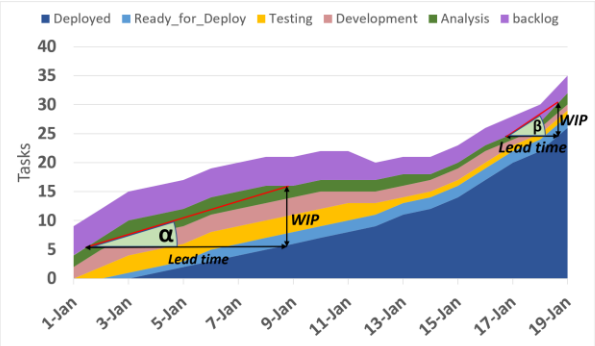

Figure 4 *CFD* Source: Adapted from (Anderson & Bozheva, 2021)

From (1) and the analysis of the CFD graph, it is possible to calculate the throughput dividing the WIP by the Lead Time which is the same as observing the tangent of α and β.

$$throughput = \frac{WIP}{Lead\ time} \qquad (2)$$

From the symmetrical scale of the graph or from (2), it is possible to observe that tan(α) is greater than tan(β) meaning, the delivery rate is higher on days when the WIP is lower. This brings up an important talking point for the team to assess what WIP value makes the flow go faster. (Hammarberg & Joakim, 2014)

2.9. SOCIALIZE THE DESIGN AND NEGOTIATE EXPECTATION

Kanban does not provide a time-based delivery period, but rather, it predicts that the main activities and functionalities that were prioritized by the customer are delivered as soon as possible, which has been shown to be totally feasible. (Birkeland, 2010) Therefore, all those involved, managers, developers and testers must participate in the implementation process by providing feedback and proposing improvements. (Nikitina, Kajko-Mattsson, & Stråle, 2012) It is necessary to build a shared understanding of the purpose, process, and associated problems. Suggest improvement actions based on scientific models and seek agreements by consensus to continually evolve. (Anderson & Bozheva, 2021)

3. THE USAGE OF KANBAN METHOD IN THE SOFTWARE INDUSTRY

In the manufacturing industry, where Kanban has been applied for several decades, (Ikonen, Pirinen, Fagerholm, Kettunen, & Abrahamsson, 2011) it has been observed a resulting facilitation of the application of Lean principles, in a simple and effective way.

In the software industry, Kanban has lower penetration compared to other development methods and practices, despite the fact that academic interest in the method has grown consistently as (Ahmad, Dennehy, Conboy, & Oivo, 2018) show in Figure 5.

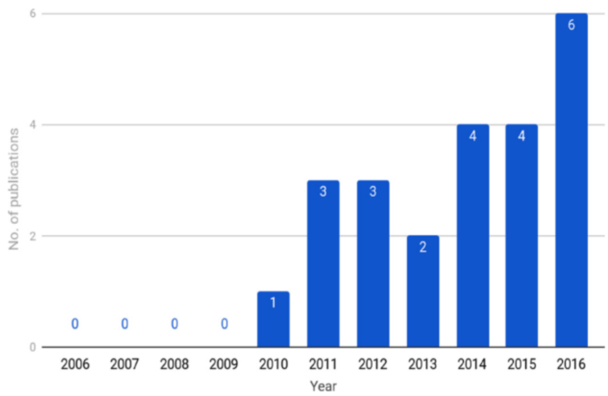

Figure 5 Publications about the Kanban method. Source: (Ahmad, Dennehy, Conboy, & Oivo, 2018)

These 23 articles, 8 in journals and 15 in conference proceedings, were selected from 382 works as primary studies on the Kanban method after a careful filtering in the systematic mapping study by (Ahmad, Dennehy, Conboy, & Oivo, 2018).

In this mapping, the main benefits and challenges of the method were collected. Dozens of studies pointed out the following as benefits of the method: 1. Improved visibility and transparency; 2. Better control of project activities; 3. Identification of workflow impediments. And as challenges: 1. Lack of good practices and understanding of the method and its implementation; 2. Evaluate performance using metrics; 3. Motivate the team to adopt new practices.

Despite the benefits reported by the research carried out, there are disagreements and disbelief in the inclusion of this new method in the daily management process, as the involvement of Kanban in the development process is new and there is still a need for a deeper understanding and changes in traditionalism for a faster adoption. (Nevenka & Saso, 2015)

Kanban also helps to identify and make clear which tasks are the highest priority, which allows managers to decide what needs to be done and how to allocate resources properly, as well as improving visibility of work in progress. (Ahmad, Kuvaja, Oivo, & Markkula, 2016) For (Ikonen, Pirinen, Fagerholm, Kettunen, & Abrahamsson, 2011) Kanban's greatest value for the industry lies in its ability to provide real-time process visualization. However, it should be noted that visualization does not guarantee success.

Although the mentioned challenges, Kanban has been successfully applied in several software companies (Seikola, Loisa, & Jagos, 2011), (Rutherford, Shannon, Judson, & Kidd, 2010), (Taipale, 2010)

4. KANBAN IN RTOS AND HIGH-RISK SYSTEMS

Considering that the kanban board is also a Lean tool, and that this is a tool that works universally across industries (Kaur, Sharma, & Matharou, 2013) and, noting that Kanban has shown results in several sectors such as , for example: Avoiding waste in hospitals (Harris, 2018); reduce production time and balance the use of resources in the electronics industry (Lee-Mortimer, 2008); reducing operating costs and losses in small automotive industries (Rahman, Sharif, & Esa, 2013) is very likely to present results in RTOS as well.

The agile manifesto rules the various agile methods proposed after its declaration and, at first, it is natural that methods that value more individuals and interactions than processes and tools, more working software than documentation, more collaboration with the customer than contract negotiation, more responding to changes than following a plan seems to be far from an industry that traditionally works with well-defined processes, tools, plans and documentation and that works with products that are critical to people's lives. (Chenu, 2009)

In fact, some restrictions and demands imposed by regulatory agencies of critical systems and RTOS applied to high-risk systems, such as, for example, standardized documentation and specific security standards conflict with some principles of the agile manifesto. However, the high demands of this industry have made the interest in agile methodologies grow and some adaptations are being proposed to enable the migration of development to an agile approach. (Gibrail & Tim, 2020)

Experience reports have shown that some agile methods have contributed significantly to this niche as well. Scrum and XP Extreme programming methods, for example, have been adopted and well accepted in embedded software industries. (Salo & Abrahamsson, 2008)

In the context of high-risk software, such as aviation software, the requirement: Avoid the loss of human life, resulted in the establishment of the standard: DO-178 issued by the RTCA (Radio Technical Commission for Aeronautics, Inc) in 1982 used by several certification agencies. Although the document contains a series of guidelines that result in the generation of a certain amount of documentation, it is not prescriptive, in other words, it does not impose a specific software development process. (Wils, Baelen, Holvoet, & Vlaminck, 2006) Currently, revision C (DO-178C) is in effect. This review encompasses several other standards such as DO-330 for certification of development tools and DO-333 which deals with formal development methods.

In the Avionic industry, specifically, this regulation increases software development costs but does not avoid the late discovery of problems and the unpredictability of requirements. (Hanssen, Wedzinga, & Stuip, 2017)

The Kanban method has a maturity model called KMM – Kanban Maturity Model that is compatible with CMMI – Capability Maturity Model Integration, Lean/TPS, Real World Risk Institute and Mission Command/Auftragstaktik. It establishes 7 levels (0-6) that allow the organization to understand its maturity in relation to its internal processes and in relation to stakeholders. The benefits of conquering each one in the levels are summarized below: 0. Oblivious: You can only deliver the products; 1. Emerging: Basic understanding of the workflow, less work overload for employees, more transparency in relation to processes; 2. Defined: Better collaboration, greater customer empathy, basic understanding of demand and capacity, better quality of work delivered. 3. Managed: Shorter lead time, quick balancing of workloads, predictability of deliveries, meeting SLAs (service level agreement) metrics that allow decision making; 4. Quantitatively managed: Risk forecasting, quantitative analyses, full compliance with regulatory agencies, data-driven decisions; 5. Optimizing: Organizational agility, shared reconfiguration of services; 6. Congruent: Resilient to external changes, long-term security. (Anderson & Bozheva, 2021) With this model, it is possible to evolve the adoption of Kanban systematically and formally as a development method and make it clear to regulatory agencies and other external entities how mature the development process is compared to other widely accepted standards like CMMI for example.

5. CONCLUSION AND FUTURE WORKS

The pressure for faster deliveries and better-quality software is also present in the RTOS software market. With the publication of his article, (Wils, Baelen, Holvoet, & Vlaminck, 2006) he took the first step towards a change in the software development paradigm on the road to highly regulated systems that require careful documentation.

Considering the advances in the maturity of the Kanban method and the success of its implementations in the various researched areas, it is noticeable that the method can bring concrete benefits to RTOS industries such as Avionics, Nautical, automobile other critical systems.

For a better acceptance of the method by regulatory actors, other case studies and experience reports would be important and even necessary. Several industries that develop non-regulated and non-vital systems have pioneered new methodologies and experimented with new techniques. It is important that this industry continues to evolve incrementally, and safely as new practices are tested and verified to continue reliable, but at the same time remain competitive and healthy.

References

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, pp. 96-113. doi:10.1016/j.jss.2017.11.045

Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018, Mar). Kanban in software engineering: A systematic mapping study. *The Journal of Systems and Software [s.l], v. 137*, pp. 96-113. doi:10.1016/j.jss.2017.11.045

Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J. (2016). Transition of Software Maintenance Teams from Scrum to Kanban. *2016 49th Hawaii International Conference on System Sciences (HICSS)* (pp. 5427-5436). Hawaii: IEEE. doi:10.1109/HICSS.2016.670

Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development A systematic literature review. *2013 39th Euromicro Conference on Software Engineering and Advanced Applications* (pp. 9-16). Santander, Spain: IEEE. doi:10.1109/SEAA.2013.28

Anderson, D. J. (2010). *Kanban: Successful Evolutionary Change for Your Technology Business. [s.l.]:*. Blue Hole Press. doi:9780984521401

Anderson, D. J., & Bozheva, T. (2021). *Kanban Maturity Model, Coaches' Edition: A Map to Organizational Agility, Resilience, and Reinvention.* Kanban University Press. doi:ISBN: 1732821259

Anderson, D. J., & Zheglov, A. (2017). *Fit for Purpose: How Modern Businesses Find, Satisfy, & Keep Customers.* Blue Hole Press. doi:ASIN: B078JFTVS8

Birkeland, J. O. (2010). From a Timebox Tangle to a More Flexible Flow. *Agile Processes in Software Engineering and Extreme Programming* (pp. 325-334). Berlin: Springer Berlin Heidelberg. doi:10.1007/978-3-642-13054-0_35

Burrows, M. (2014). *Kanban from the Inside: Understand the Kanban Method, connect it to what you already know, introduce it with impact.* Blue Hole Press.

Chenu, E. (2009). Agility and lean for avionics. *Lean, Agile Approach to High-Integrity Software Conference.* Paris.

Corona, E., & Pani, F. E. (2013, 01). A Review of Lean-Kanban Approaches in the Software Development. *WSEAS Transactions on Information Science and Applications*, pp. 1-13.

Gibrail, I., & Tim, S. (2020, 08 01). A case study of agile software development for safety-Critical systems projects. *Reliability Engineering & System Safety*, p. 106954. doi:10.1016/j.ress.2020.106954

Hammarberg, M., & Joakim, S. (2014). *Kanban in Action.* Manning Publications. doi:ISBN 9781617291050

Hanssen, G., Wedzinga, G., & Stuip, M. (2017). An Assessment of Avionics Software Development Practice: Justifications for an Agile Development Process. *International Conference on Agile Software Development*, (pp. 217-231). Trondheim, Norway. doi:10.1007/978-3-319-57633-6_14

Harris, A. M. (2018, 04). Boosting Efficiency with a kanban plan. *Physician Leadership Journal*, p. 50+. Retrieved from https://link.gale.com/apps/doc/A533408878/AONE?u=googlescholar&sid=bookmark-AONE&xid=2b51f2c7

Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation. *2011 16th IEEE International Conference on Engineering of Complex Computer Systems* (pp. 305-314). IEEE. doi:10.1109/ICECCS.2011.37

Kaur, S., Sharma, N., & Matharou, G. (2013, 03 01). Lean manufacturing tool and techniques in process industry. *International Journal of Scientific Research and Reviews*, pp. 54-63.

Lee-Mortimer, A. (2008). A continuing lean journey: an electronic manufacturer's adopting of Kanban. *Assembly Automation*, pp. 103-112. doi:10.1108/01445150810863662

Leon-Garcia, A. (2008). *Probability, Statistics, and Random Processes For Electrical Engineering.* Prentice Hall. doi:ISBN 0131471228, 9780131471221

Murman, E., Walton, M., & Rebentisch, E. (2000). Challenges in the better, faster, cheaper era of aeronautical design, engineering and manufacturing. *The Aeronautical Journal*, pp. 481-489. doi:10.1017/S0001924000091983

Nevenka, K., & Saso, K. (2015, 01). Usage of Kanban Methodology at Software Development Teams. *Journal of applied economics and business*, pp. 25-34.

Nikitina, N., Kajko-Mattsson, M., & Stråle, M. (2012). From scrum to scrumban: A case study of a process transition. *2012 International Conference on Software and System Process (ICSSP)* (pp. 140-149). Zurich, Switzerland: IEEE. doi:10.1109/ICSSP.2012.6225959

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit.* Pearson.

Rahman, N. A., Sharif, S. M., & Esa, M. M. (2013, 01 01). Lean Manufacturing Case Study with Kanban System Implementation. *International Conference on Economics and Business Research 2013*, pp. 174-180. doi:10.1016/S2212-5671(13)00232-3

Rutherford, K., Shannon, P., Judson, C., & Kidd, N. (2010). From Chaos to Kanban, via Scrum. *Agile Processes in Software Engineering and Extreme*

Programming (pp. 344-352). Berlin: Springer.

Salo, O., & Abrahamsson, P. (2008, 03 01). Agile methods in European embedded software development organisations: A survey on the actual use and usefulness of Extreme Programming and Scrum. *Software, IET,* pp. 58 - 64. doi:10.1049/iet-sen:20070038

Seikola, M., Loisa, H.-M., & Jagos, A. (2011). Kanban Implementation in a Telecom Product Maintenance. *2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 321-329). Oulu, Finland: IEEE.

Senapathi, M., & Drury-Grogan, M. (2021, 10 05). Systems Thinking Approach to Implementing Kanban: A Case Study. *Journal of Software: Evolution and Process*, p. 33:e2322. doi:10.1002/smr.2322

Simić, D., Svirčević, V., Corchado, E., Calvo-Rolle, J. L., Simić, S. D., & Simić, S. (2021). Modelling material flow using the Milk run and Kanban systems in the automotive industry. Jan;38(1):. *Expert Systems 38*, 1-15 e12546.

Taipale, M. (2010). Huitale – A Story of a Finnish Lean Startup. *Lean Enterprise Software and Systems LESS 2010. Lecture Notes in Business Information Processing, vol 65* (pp. 111-114). Berlin: Springer.

Wils, A., Baelen, S. V., Holvoet, T., & Vlaminck, K. D. (2006, 06 17). Agility in the avionics software world. *Extreme Programming and Agile Processes in Software Engineering, 7th International Conference, XP 2006. Proceedings. Lecture Notes in Computer Science 4044* (pp. 123-132). Oulu, Finland: Springer. doi:10.1007/11774129_13