Kanban in Software Engineering: A Systematic Mapping Study

Muhammad Ovais Ahmad, Denis Dennehy, Kieran Conboy, Markku Oivo
Empirical Software Engineering in Software, Systems and Services, P.O. Box 4500, University of Oulu, 90014, Finland
Lero Research Centre & Whitaker Institute, NUI Galway, Ireland c College of Business, Public Policy & Law, Lero Research Centre & Whitaker Institute, NUI Galway, Ireland

Abstract

Following a well-established track record of success in other domains such as manufacturing, Kanban is increasingly used to achieve continuous development and delivery of value in the software industry. However, while research on Kanban in software is growing, these articles are largely descriptive, and there is limited rigorous research on its application and with little cohesive building of cumulative knowledge. As a result, it is extremely difficult to determine the true value of Kanban in software engineering. This study investigates the scientific evidence to date regarding Kanban by conducting a systematic mapping of Kanban literature in software engineering between 2006 and 2016. The search strategy resulted in 382 studies, of which 23 were identified as primary papers relevant to this research. This study is unique as it compares the findings of these primary papers with insights from a review of 23 Kanban experience reports during the same period. This study makes four important contributions, (i) a state-of-the-art of Kanban research is provided, (ii) the reported benefits and challenges are identified in both the primary papers and experience reports, (iii) recommended practices from both the primary papers and experience reports are listed and (iv) opportunities for future Kanban research are identified.

Keywords: Kanban, Lean, software engineering, software development

1. Introduction

Rooted in lean manufacturing, Kanban has been used across a range of industries, including aeronautics (Venables, 2005), healthcare (Kim et al., 2009), retail clothing (Tokatli, 2008), human resource (Wijewardena, 2011), and software development (Anderson, 2010). Kanban is a Japanese word meaning 'card or signboard (Sugimori et al., 1977; Anderson, 2010), verbal instruction, a light, a flag, or even a hand signal and is based on a pull system (Kimura and Terada, 1981; Huang and Kusiak, 1996).

The Kanban method has been well received in software engineering, and there is strong anecdotal evidence to suggest that its use is becoming quite prevalent across the community (Anderson, 2013; Dennehy and Conboy, 2016; Nord et al., 2012; Petersen and Wohlin, 2011; Poppendieck and Cusumano, 2012; Power and Conboy, 2015). Annual 'State of Agile' reports show that the use of Kanban increased from 31% to 39% in 2015 and from 39% to 50% in 2016 (VersionOne, 2016, 2017).

Software engineering has been plagued by numerous problems such as (i) a lack of reliability, (ii) poor response to change, (iii) limited agility, and (iv) excessive costs (Anderson, 2010). Kanban is seen as a method to overcome these challenges, allowing teams to respond to dynamic market changes, increase quality, reduce waste, and improve predictability (Abrahamsson et al., 2009; Dybå and Dingsøyr, 2008; Nurdiani et al., 2016; Taibi et al., 2017).

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

Despite the popularity of Kanban in software engineering, this study identifies a number of shortcomings in the Kanban literature in this regard. Firstly, in comparison to manufacturing, where the concept of Kanban has been extensively studied, practiced and matured over time, Kanban in software engineering must operate in an environment that is complex, highly contextual, and socially embedded (Lyytinen and Rose, 2006). To date, research has not sufficiently studied or addressed these characteristics (e.g. Anderson et al., 2011; Cocco et al., 2011; Concas et al., 2013). Secondly, the effectiveness of Kanban has largely been supported by anecdotal evidence and largely by consultancy organisations whose primary business is based on these purported benefits (i.e. Cutter, 2011; Hurtado, 2013; Kniberg and Skarin, 2009; Shalloway, 2010). Thirdly, the three published systematic literature reviews (SLR's) related to Kanban have limitations (i.e. Al-baik and Miller, 2015; Ahmad et al. 2013; Corona and Pani, 2013) as shown in Table 1.

The literature review conducted by Al-Baik and Miller (2015) cited twenty peer-reviewed and seventeen non-peer reviewed articles (i.e. Anderson, 2010; Ladas, 2009; Boeg, 2012; Terlecka, 2012; Kniberg and Skarin, 2009; Zhang, 2010). As the research rigor of these non-peer reviewed articles has not be established, they do not adequately contribute to the accumulative building of knowledge about Kanban. The literature review conducted by Corona and Pani (2013) focused on the features of Kanban products and not its actual use in the real-world context in which Kanban is intended to be used. The literature review conducted by Ahmad et al., (2013) and Al-Baik and Miller (2015) focused on Kanban use only in the context of software development and excluded some Kanban experience reports and empirical studies with no explanations. However, this mapping study includes all Kanban experience reports and empirical studies between 2006 and 2016, which includes the broader areas of the software engineering discipline, namely, software development, software maintenance, software product development, project and project portfolio management and software engineering education.

Table 1: Comparison of previous Kanban SLR's

Comparison element	**Al-Baik and Miller (2015)**	**Corona and Pani (2013)**	**Ahmad et al., (2013)**	**This study**
Purpose	Provides insight into Lean and Kanban concepts, principles and techniques	Discusses tools available for Kanban boards in software development	Identifies the use of Kanban only in software development literature	Kanban in the field of software engineering (e.g. software development, software maintenance, software product, program and portfolio management, software engineering education)
Years included	1990 - 2012	Unknown - 2012 (authors did not specify date)	2004 - 2011	2006 - 2016
Sources of primary studies	Combination of grey and scientific literature • 21 empirical	Selected 14 Kanban tool web sites published on	Scientific literature • 8 empirical studies	Scientific literature • 23 empirical studies

	studies • 13 non-peer reviewed books and doctoral thesis • 8 web articles	http://limitedwipsociety. ning.com	• 9 experience reports • 2 simulation studies	• 23 experience reports

To address this gap in knowledge, the overarching goal of this study is to identify the state-of-the-art of Kanban in software engineering by conducting a systematic mapping study. Conducting a systematic mapping of Kanban in software engineering is important as it can be used to provide a valuable baseline to assist new research efforts (Kitchenham et al., 2010; Petersen et al., 2015). The aims of this systematic mapping study are to:

1. provide a state-of-the-art of Kanban research in software engineering
2. synthesis the claimed benefits and challenges of Kanban in software engineering
3. identify the opportunities for future Kanban research

The paper is structured as follows. Background to Kanban in manufacturing and software engineering is presented. Next, the process (e.g. planning, conducting, reporting) of systematic mapping is presented and limitations of the study are acknowledged. Then, the state-of-the-art of Kanban research is presented. The reported benefits and challenges of Kanban are also analysed and categorised. Followed by discussion and implications for research and practice highlighted. The paper ends with conclusions and directions for future research.

2. Background and related work

This section commences with the origins of Lean and Kanban in manufacturing and explains how these concepts are used together. The evolution of Lean and Kanban in software engineering is then discussed. Related work on Kanban in software engineering is also discussed.

2.1 Lean and Kanban in Manufacturing

Lean, which can be traced back to the 1940s, historically focused on cost reduction (Ohno, 1988), “the elimination of waste” (Naylor et al., 1999; Ohno, 1988; Womack et al., 1990), and “doing more with less” (Towill and Christopher, 2002). Sugimori et al., (1977) published the first academic paper describing kanban and advocated three reasons for its use: (i) reduction in information processing cost, (ii) rapid and precise acquisition of facts, and (iii) limiting surplus capacity of preceding shops or stages. However, the concept of Lean has morphed over time with emphasis shifting from cost and waste to value maximisation (Conboy, 2009). Lean strives to deliver maximum value to the customer by reducing waste, controlling variability, maximizing the flow of information, focusing on the whole process, and not on local improvements (Anderson et al., 2011; Poppendieck, 2002). Lean is a mindset, a mental model of

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study.** ***Journal of Systems and Software*****,** ***137*****, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

how the world works (Poppendieck and Poppendieck, 2013). Lean thinking is guided by five interlinked concepts (Wang et al., 2012):

1. Value: Value as defined by the end customer.
2. Value stream: A map that identifies every step in the process and categorises each step in terms of the value it adds.
3. Flow: Refers to the continuous flow of valuable work in the process.
4. Pull: Customer orders pull product, ensuring nothing is built before it is needed.
5. Perfection: Striving for perfection in the process by continuously identifying and removing waste.

Lean was part of the Toyota Production System (TPS) and is based on two concepts: (i) automation with a human touch and (ii) Just-In-Time (JIT) production (Womack et al. 1990; Ohno 1988). To implement JIT at Toyota, Taiichi Ohno developed Kanban which enabled Toyota to (i) work effectively under specific production and market conditions (Ohno, 1988), (ii) facilitate smooth operation of TPS (Becker and Szczerbicka, 1998; Chai, 2008; Gross and McInnis, 2003; Liker, 2004), and (iii) promote and achieve continuous improvement (Hiranabe, 2008; Shingo, 1989).

The benefits of kanban in manufacturing include: (i) limiting work in progress (WIP), (ii) monitoring and controlling production process, (iii) visual scheduling, (iv) improving flow, (v) responsiveness to changes, (vi) facilitating high production, (vii) preventing overproduction, (viii) improving capacity utilisation, (ix) and reducing production time (Gross and McInnis 2003; Gravel and Price, 1988; Kumar and Panneerselvam, 2007; Ohno, 1988; Zhang et al., 2011).

2.2 Lean and Kanban in Software Development

Lean software development is increasingly being adopted by software teams (Anderson et al., 2011). It is reported that David Anderson was the first to adopt Kanban in 2004 with a software development team at Microsoft, located at Hyderabad, India (Anderson, 2010; Ahmad et al. 2013). However, it was Poppendieck and Poppendieck (2003) who published the first book that adopted Lean principles from manufacturing and applied them to software development, which consists of seven principles: i) eliminate waste, ii) amplify learning, iii) decide as late as possible, iv) deliver as fast as possible, v) empower the team, vi) build integrity, and vii) see the whole. These principles were later refined and are listed in Table 2.

Kanban is described by Anderson (2010, p. 6) as "*Kanban (capital K) is an evolutionary change method that utilizes a kanban (small k) pull system, visualization, and other tools to catalyse the introduction of Lean ideas… the process is evolutionary and incremental*".

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

Table 2: Principles of Lean and Kanban in software

Lean software development (Poppendieck and Poppendieck, http://www.poppendieck.com/	**The Principles of Product Development Flow (Reinertsen, 2009)**	**Kanban Principles (Anderson, 2010)**
• Optimize the whole • Focus on customers • Energize workers • Eliminate waste • Enhance learning • Increase flow • Build quality in • Keep getting better	• Use economically based decision-making • Understand behaviour of queues • Exploit variability • Reduce batch size • Apply WIP (work in progress) constraints • Use cadence, synchronisation and flow control • Use fast feedback loops • Decentralise control	• Visualize workflow • Limit work in progress (WIP) • Measure and manage flow • Make process policies explicit • Use (theoretical) models to recognize improvement opportunities

Kanban enacts the Lean principles, discussed previously, by providing a tool to optimise an outcome for value through a focus on flow management (Anderson, 2010). Each of the five Kanban principles proposed by Anderson (2010) is discussed in the remainder of this section.

Visualise workflow: Work moves through different states (Planned, In Progress, Done) as it moves through the organisation. The Kanban system encourages the visualisation of workflow as work moves through the organisation (Power and Conboy, 2015; Anderson, 2010) by using physical or virtual boards and cards. The cards are used to visually represent work items, enable team members to observe work-in-progress and for the teams to self-organize by assigning their own tasks and to complete work without direction from a manager (Anderson, 2010; Ikonen et al., 2011; Williams, 2012).

Limit work in progress (WIP): Explicit WIP limits are used to manage the quantity of work-in-progress at any given stage in the workflow (Power, 2014). If there is no explicit WIP limit and no signalling to pull new work through the system then it is not a Kanban system (Anderson, 2010).

Measure and manage flow: There are five commonly known techniques that are used to manage flow: (i) value stream maps, (ii) Kanban board, (iii) cumulative flow diagrams (CFDs), (iv) burn-down charts, and (v) line of balance status charts (Anderson, 2010; Petersen et al., 2014; Mujtaba et al., 2010). The quality of flow is measured using four key metrics: queue size, throughput rate, cycle time, and lead time (Power and Conboy, 2015; Reinertsen, 2009). Flow is the hardest concept of Lean to understand as it is concerned with people, processes, and culture (Melton, 2005).

Make process policies explicit: As work moves through different states on the Kanban board, establishing explicit policies, also referred to as 'entry' and 'exit' criteria is required to determine when a work item can be pulled from one state to another (Power, 2014). Explicit policies enable organisations to observe 'cause and effect' when changes are made to the process (Cutter, 2011) and to quantify and balance throughput (Greaves, 2011).

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study.** ***Journal of Systems and Software*****,** ***137*****, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Use models to recognize improvement opportunities: Continuous improvement opportunities can be identified by using models such as Theory of Constraints, and Systems Thinking (Anderson, 2010) as well as frequently using techniques such as using value stream mapping (Zang, 2011).

2.3 Simulating Kanban Principles

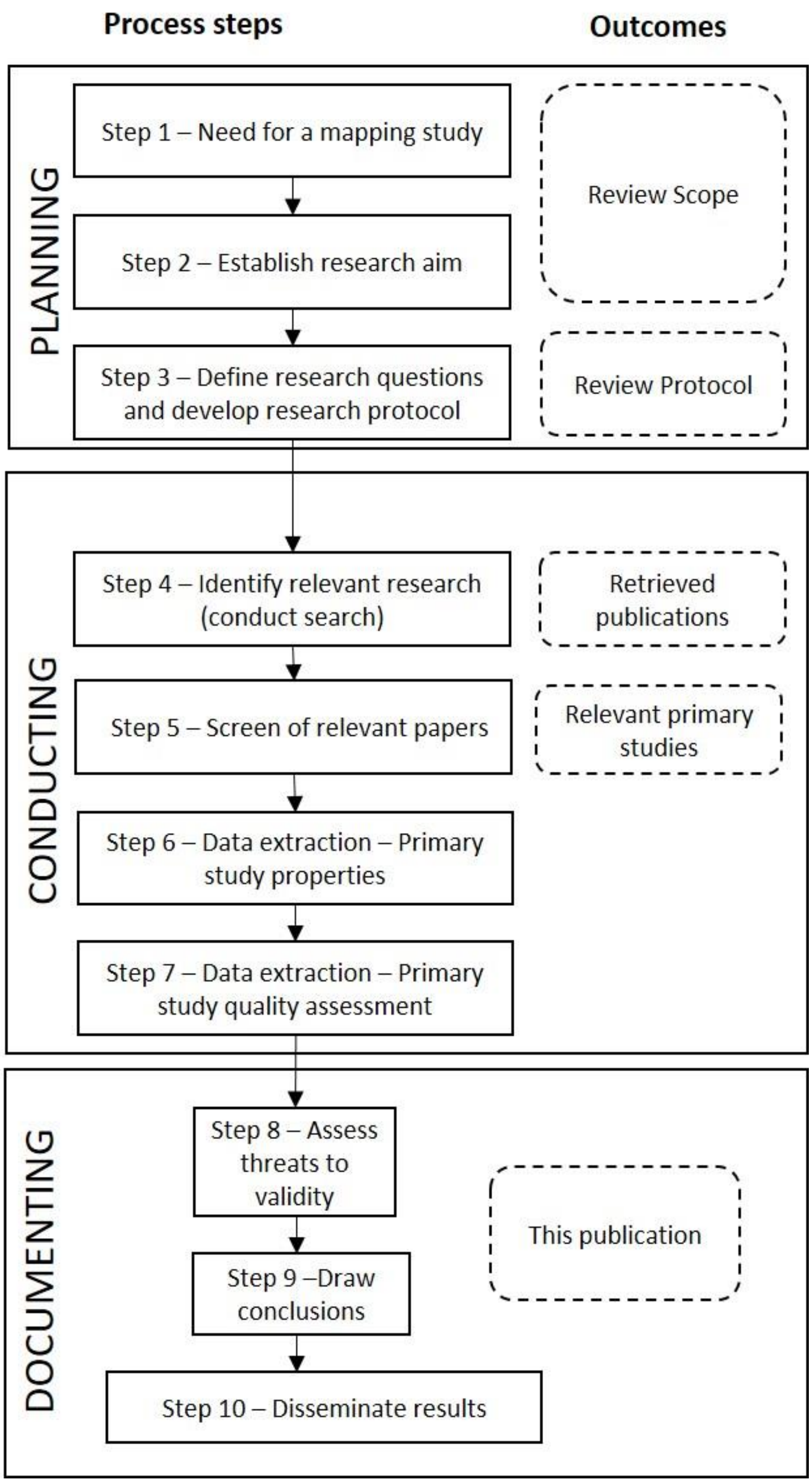
Studies on Kanban using simulation techniques has been conducted to analyse the applicability and effectiveness of Kanban in software development (Anderson et al, 2011; Cocco et al., 2011; Concas et al., (2013). For example, Cocco et al., (2011) analyse the dynamic behaviour of Kanban and Scrum adoption in comparison to a traditional software development process. The reported result was that Kanban helped control and manage workflow effectively while minimising lead-time. Another simulation study by Anderson et al., (2011) highlighted that application of WIP limit results in a constant flow of features and the absence of WIP limits results in a more irregular flow of features. While a simulation study by Concas et al., (2013) revealed that Kanban helps to reduce the average time needed to complete customer requests and WIP limits can increase the efficiency of software maintenance.

Although providing interesting findings on Kanban, these simulation studies were excluded as their focus was too narrow, they focused on the functionality of Kanban itself, and not the wider parameters, these being the social and contextual nature of software engineering that Kanban is intended to be used (c.f. Dennehy and Conboy, 2016; Lyytinen and Rose, 2006; Olerup, 1991; Wastell and Newman, 1993).

3. Research methodology

The section outlines the systematic mapping process adopted in this study, which follows the established guidelines and procedures proposed by Kitchenham et al., (2011) and Petersen et al., (2015). The systematic mapping process is illustrated in Figure 1 and consists of 11 steps across three phases, namely, planning (3 steps), conducting (4 steps), and documenting (4 steps). Each of these three phases and eleven steps are discussed in detail in the remainder of this section.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045



Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study.** ***Journal of Systems and Software***, ***137***, **96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Figure 1: Mapping study steps

3.1 Planning the mapping study

This section, presents steps 1, 2 and 3 that are related to the planning of this systematic mapping study. The motivation to conduct a systematic mapping study is to focus on the "classification and thematic analysis of literature on a software engineering topic" (Kitchenham et al., 2011, p. 640). In this instance, the motivation for conducting a mapping study is to provide a start-of-the-art of Kanban research in software engineering between 2006 and 2016 (Step 1).

The main objectives of this study (Step 2), as previously stated, are to (i) establish the body of knowledge of Kanban by identifying and categorizing the available research on the topic, (ii) identify the most relevant Kanban articles in software engineering, (iii) assess the quality of the existing research in terms of relevance and rigour, (iv) distil the reported benefits and challenges of Kanban in software engineering, and (v) identify the opportunities for future Kanban research. To achieve these broad research objectives, the research questions (Step 3) listed in Table 3 will be answered.

Table 3: Research questions

ID	**Research question**
RQ1	What is the current state of Kanban research in software engineering?
RQ1.1	What number of academic studies on Kanban has been published between 2006 and 2016?
RQ1.2	What are the publication channels used to publish studies on Kanban?
RQ1.3	What do researchers mean when they refer to the term Kanban in software engineering?
RQ1.4	What research methods have been used in studies on Kanban?
RQ1.5	What kinds of contributions are provided by studies on Kanban?
RQ1.6	What is the quality of the published papers?
RQ1.7	What are the knowledge areas of studies on Kanban?
RQ2	What are the claimed benefits of Kanban in software engineering literature?
RQ3	What are the reported challenges faced in the use of Kanban in software engineering?
RQ4	What insights are gained from a review of Kanban experience reports?
RQ4.1	What are the claimed benefits of Kanban in experience reports?
RQ4.2	What are the reported challenges in the use of Kanban in experience reports?
RQ5	What recommendations for Kanban use are provided by empirical studies and experience reports on Kanban?

As RQ1 is a broad research question, seven questions (RQ1.1 - RQ1.7) have been identified as being pertinent in order to answer this question. RQ2 and RQ3 will provide a synthesis of the reported benefits and challenges of Kanban in the software engineering domain. RQ4 – RQ4.2 identify the claimed benefits and reported challenges of Kanban in experience reports. RQ5 examines what recommendations for Kanban use are provided by empirical studies and experience reports on Kanban.

3.2 Conducting the mapping study

This section, presents steps four, five, six and seven of this systematic mapping study.

3.2.1 Search strategy and data sources

In this study, the search string was developed based on the scope of this study, which includes search terms, ‘population’ and ‘intervention’ (Kitchenham et al., 2011). *Population* refers to the application area which is software and *intervention* is Kanban. Software is the expected search that will include all documents with the word "software" in title, abstract or keyword. The search string was “Kanban AND Software”. The rationale for using the term “software” is that, this study will cover studies that discuss software, software development, software engineering or software intensive products, services, and systems. The term Kanban was used to include all Kanban papers. The selected databases and the retrieved papers (Step 4) are listed Table 4.

Table 4: Selected databases and retrieved papers

Database	**Filter**	**No. of retrieved papers**
ACM Digital Library	Only conference papers and journal articles	22
IEEE Xplore	Only conference papers and journal articles	71
ISI Web of Science	Only articles in the following research areas: computer science, software engineering, information systems, engineering	78
Scopus - Sciencedirect	Only conference papers and journal articles in English	211
	Total	382

The selected databases are pertinent to this study as these return the most publications (Dyba et al., 2007; Kitchenham and Brereton, 2013). For each of the four selected databases, using the specified search string retrieves an initial list of studies. Databases with additional functionality of limiting relevance of the studies to specific fields such as software engineering and computer science were used. The records are imported into Microsoft Excel sheet format. The basic input includes meta-data such as (i) title, (ii) author, (iii) year, (iv) publication type, and (v) abstract.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

3.2.2 Primary study selection procedure

Screening of the retrieved publications (Step 5) was achieved by following the best practices proposed by Kitchenham (2004) and Dybå and Dingsøyr (2008). The paper selection process used in this study is illustrated in Figure 2.

The search string used across four the databases (e.g. ACM Digital Library, Scopus, IEEE Software, ISI Web of Science) retrieved 382 publications. Two authors independently analysed the 382 publications in order to (i) remove duplicate papers, (ii) non-English publications, (iii) non-software engineering studies, and (iv) non-peer reviewed scientific papers. The search strategy included the term 'Kanban', which resulted in several hits on papers about Kanban in the manufacturing industry. Those papers were excluded, as the manufacturing industry is outside the focus of this study. This process resulted in 252 publications being excluded and 130 primary studies included.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

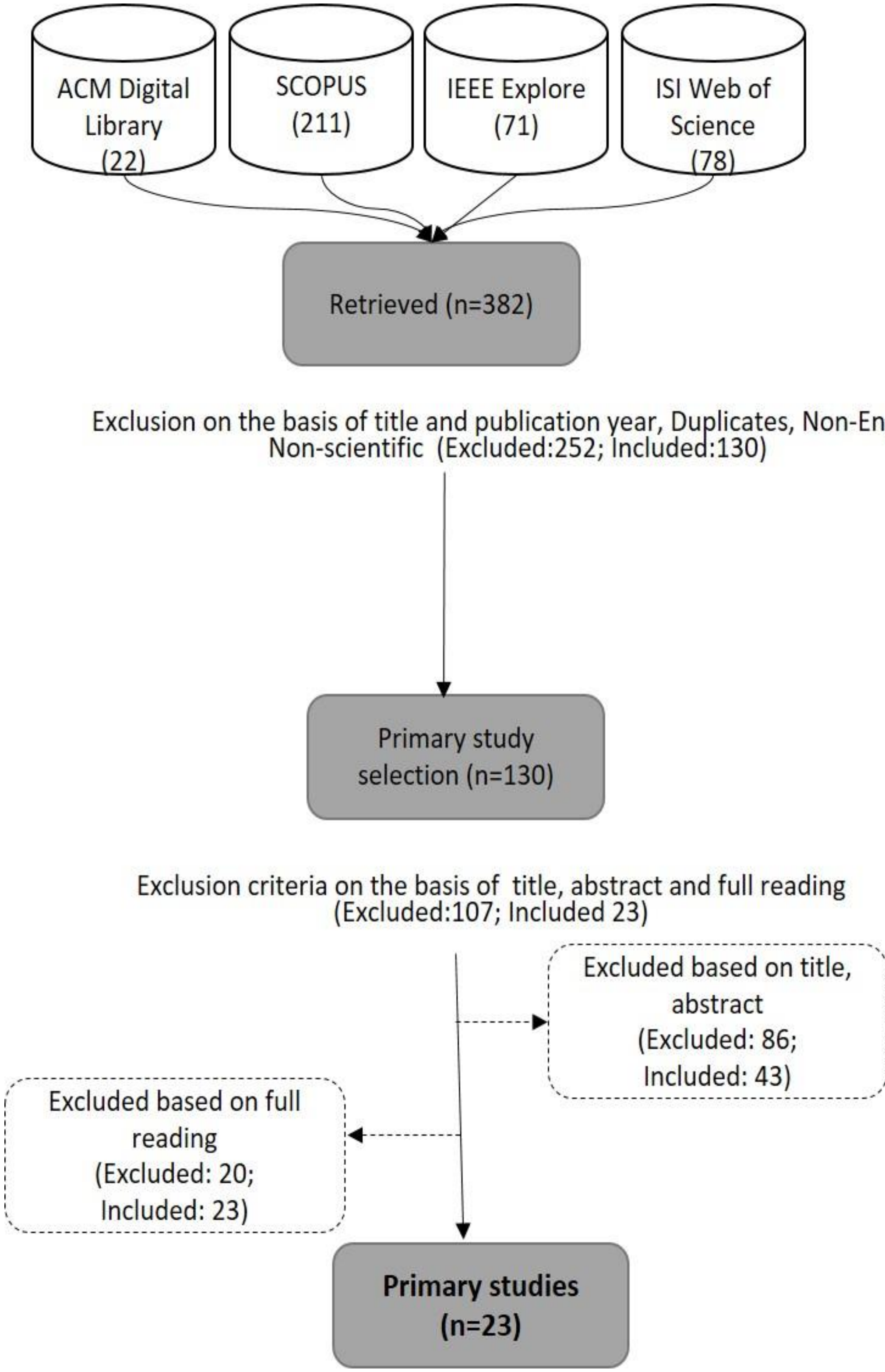


Figure 2: Paper selection process

Next, two authors separately analysed the 130 primary studies over a four-day period at the university of the lead author. During this period, in-depth reviews of each paper were conducted, this required the researchers to read the (i) titles, (ii) abstracts, (iii) introduction, and (iv) conclusion. The outcome of this process produced 23 primary studies, which were then quality assessed using the 11 factor criteria proposed by Dybå and Dingsøyr (2008). The primary studies (P) are listed in Appendix A and are identified by the symbol ∗[P]).

3.2.3 Inclusion and exclusion criteria

Studies were eligible for inclusion in the systematic mapping if they presented empirical data on Kanban usage in software engineering or if a non-empirical study (e.g. systematic mapping study, systematic literature review, experience reports) clear evidence of research rigor. Studies using students or professional software engineers were included. The inclusion criteria used was:

- The study should be written in English
- The study should be published between 2006 and December 2016
- The study directly answers one or more of the research questions of this study
- The study should clearly state its focus on Kanban in the software engineering domain
- The study should describe the elements and the approach used to implement Kanban
- If the study has been published in more than one journal or conference, the most recent version of the study is included.

Studies were excluded if their focus was not specifically Kanban or if they did not provide academic rigour or industry relevance. The exclusion criteria used was:

- Short papers
- Duplicate articles
- Not written in English
- Simulation studies
- Studies not clearly focused on Kanban in the software engineering domain (e.g. industrial engineering, manufacturing and automotive industry)
- Not peer-reviewed scientific papers (i.e. books, book chapters, articles)

3.2.4 Identification of primary studies

The process of identifying primary studies that constitute a mapping study is critical for the success of this study. The search string was built on two key terms, namely ‘Kanban’ and ‘Software’. Nevertheless, the threat of missing relevant articles remains. Use of different terminology in the search string may have biased the identification of primary papers. This is a minor threat as there is no synonym for Kanban and the relatively large volume of retrieved papers (382). The search string was used to search keywords, titles, and abstracts; hence, the search strategy was to retrieve as many documents as possible that were related to Kanban in software engineering and closely related contexts (i.e. software development, information systems development). The titles of the retrieved 382 studies were read and any titles that clearly indicated that it was outside the focus of this study were excluded in this stage. For example the

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study.** ***Journal of Systems and Software*****,** ***137*****, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

search term ‘Kanban’, retrieved studies about Kanban in the domains of manufacturing and industrial engineering. If a title did not clearly reveal application domain of the paper it was included for review in the subsequent steps. At the end of this activity 130 papers remained. Next, two authors read the titles, abstracts and keywords of the remaining 130 papers and their relevance to this mapping study examined. When as abstract appeared to be unclear about the contents of the full paper it was included in the next step. At the end of this activity, 43 relevant papers remained and these were read in full by at least two authors. Analysis of the 43 papers was based on the objective of the study, context description, research design, data collection and analysis, justification of findings and results. In cases where there was disagreement between the first two authors, input was sought from the third and fourth authors. At the end of this activity 23 publications were selected as the final primary studies.

3.2.5 Data extraction and analysis

Once the primary papers were selected, they were subject to in-depth analysis (Step 6). Nonetheless, the analysis of the paper is vulnerable to a validity threat due to researcher bias. To address this threat, researcher triangulation and explicit definitions of the data to be extracted was established. The primary papers were analysed based on study properties (e.g. paper type, method, contributions, domain, pertinence, and publication channel) and study quality (e.g. research rigor and relevance). Each paper was analysed separately by each author and then a combined peer-review conducted. In cases of disagreement, input was requested from author three or four. Finally, one researcher (author one) who had a panoptic vision of the study reviewed each activity of the analysis to ensure consistency in the analysis and consolidation of the results.

3.2.6 Quality Assessment

The quality assessment (Step 7) of the 23 primary papers, applied the 11 factor quality assessment criteria (see Table 5) proposed by Dybå and Dingsøyr (2008) to assess the quality of the 23 primary papers. Each of the criteria was graded on a binary (‘1’ or ‘0’) grade, in which ‘1’ indicates ‘yes’ to the question, while ‘0’ indicates ‘no’.

Table 5: Quality assessment questions (source: Dybå and Dingsøyr, 2008)

No.	**Quality question**
1.	Is this a research paper? (or is it merely “lessons learned” report based on expert opinion)
2.	Is there are a clear statement of the aims of the research?
3.	Is there an adequate description of the context in which the research was carried out?
4.	Was the research design appropriate to address the aims of the research?
5.	Was the recruitment strategy appropriate to the aims of the research?
6.	Was there a control group with which to compare treatments?
7.	Was the data collected in a way that addressed the research issue?
8.	Was the data analysis sufficiently rigorous?
9.	Has the relationship between researcher and participants been considered to an adequate degree?
10.	Is there a clear statement of findings?

11.	Is the study of value for research or practice?

Collectively, these 11 criteria provided a measure of the extent to which the quality of the 23 primary papers could be appropriately assessed. The limit the degree of subjectivity of the assessment, at least two researchers independently assessed the 23 papers. The results of these independent assessments were used to provide a more objective quality assessment of the 23 papers, which are presented in the analysis section of the paper.

4. Results

This section presents the results from the analysis of the 23 primary studies, which is based on the research questions previously mentioned (Section 3, Table 3). The results represent the state-of-the-art of Kanban research in software engineering based on the following (i) publication by year, (ii) publication channel, (iii) Kanban definition, (iv) research method adopted, (v) type of contribution, (vi) reporting quality, (vii) knowledge areas of studies on Kanban, (viii) reported benefits, and (ix) reported challenges.

4.1 RQ 1.1 Publication by year

The aim of this research question is to establish the annual number of academic studies on Kanban within the field of software engineering between 2006 and 2016. Figure 3 lists the number of publications by year of the primary studies over the 10-year period.

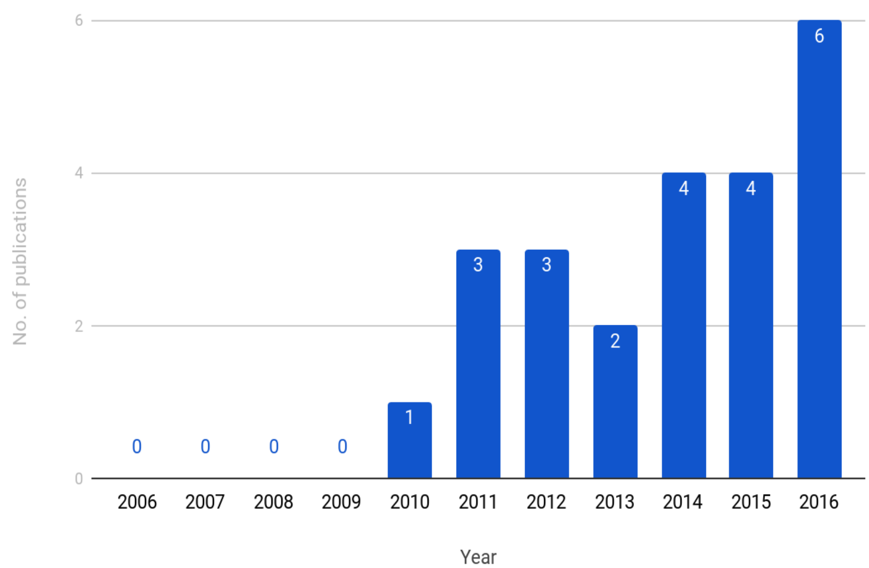


Figure 3: Publication by year

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

This categorisation is valuable as it indicates that although academic studies on Kanban in software engineering remains low, there is a slight increase in interest in recent years. The 23 primary papers were published between 2010 and 2016 and based on the inclusion and exclusion criteria of this study, there are no empirical studies represented between 2006 and 2009.

4.2 RQ1.2 Publication channel

The aim of this research question is to identify the main channels where Kanban studies are disseminated. Table 6 shows that sixteen of the primary papers were published in peer-reviewed journals and eight were published in international conferences.

Table 6: Kanban papers by target journal and conference

Channel	Title	No. of publications	Primary study
Journal (n=8)	IEEE Software	1	P21
	Journal of Systems and Software	1	P9
	Journal of Empirical Software Engineering	1	P7
	Journal of Software: Evolution and Process	1	P3
	International Journal of Engineering Education	1	P16
	IEEE Transactions on Engineering Management	1	P17
	International Journal of Human-Computer Interaction	1	P15
	World Transactions on Engineering and Technology Education	1	P5
Conference (n=15)	International Conference on Agile Software Development	3	P4, P18, P23
	International Conference on Software Engineering and Advanced Applications	2	P6. P13
	Hawaii International Conference on System Sciences	2	P1, P22
	International Conference on Software and Systems Process	2	P19, P20
	IFIP Advances in Information and Communication Technology	1	P11
	WSEAS Transactions on Information Science and Applications	1	P8
	International Conference on Software Engineering Companion	2	P10, P12
	International Conference on Engineering of Complex Computer Systems	1	P14
	International Conference on Global Engineering Education Conference	1	P2
	Total	23	23

Having identified the main publication channels of Kanban research, the next sections identifies the definitions of Kanban used in the journal and conference papers.

4.3 RQ1.3 Definitions of Kanban

The aim of this research question is to identify and analyse the different definitions of Kanban being used in Kanban research. A variety of Kanban definitions have been provided in the primary studies (see Table 7). Fifteen out of the 23 primary studies follow the definition of Kanban as defined by Anderson (2010) - a way to execute Lean principles. Two studies (P19, P20) followed the Kniberg and Skarin (2010) definition of Kanban, two studies (P13, P14) used the definition proposed by Hiranabe (2008), and one study (P17) followed the Kanban definition proposed by Ladas (2008).

Table 7: Kanban definitions in primary studies

Cited Author	Definition	Primary study source
Kanban as defined by Anderson (2010)	Kanban (capital K) as the evolutionary change method that utilizes a kanban (small k) pull system, visualization, and other tools to catalyze the introduction of Lean ideas into technology development and IT operations.	P1, P2, P3,P4, P5, P6, P9, P10, P11, P15, P16, P18, P22, P23
Kanban as defined by Kniberg and Skarin (2010)	Kanban is surmised as (i) visualize the workflow, (ii) limit Work In Progress, and (iii) measure the lead-time.	P19, P20
Kanban as defined by Hiranabe (2008)	A wall showing the current status is sometimes called "Task Kanban" or "Software Kanban". The wall labeled as: "To Do", "Doing", "Done" and limit WIP.	P13, P14
Kanban as defined by Ladas (2008)	Kanban is pull system that visualize and coordinate the work of the software development teams.	P17
Kanban, self defined	A set of concepts, principles, practices, techniques, and tools for managing the product development process with an emphasis on the continual delivery of value to customers, while promoting ongoing learning and continuous improvements.	P7
	We can define Kanban in software process as a pull system with WIP limits and visualized by the Kanban board.	P8
	Kanban is a workflow management method especially suitable for managing continuous software engineering work.	P12

Although the three different definitions of Kanban in the above mentioned studies share the term 'visualisation', there remains a lack of cohesion and consensus around the definition of Kanban. While three primary studies (P7, P8, P12) created their own definition of Kanban without any reference to previous definitions of Kanban.

4.4 RQ1.4 Research methods used in primary studies

The aim of this research question is to categorize available Kanban research according to research method. The diverse research methods used are shown in Figure 4. The focus of this mapping study was on both empirical and theoretical studies of Kanban. Seven of the 23 primary studies on Kanban adopted a mixed methods approach, seven studies used a qualitative method and five studies adopted a quantitative method. Only one study used action research and three were theoretical studies on Kanban using a systematic literature.

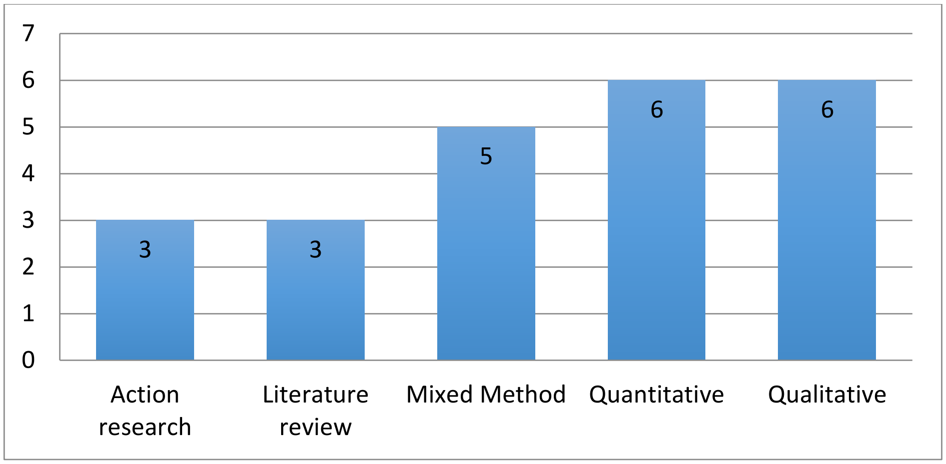


Figure 4: Research methods used in primary studies

A deeper analysis of the research methods was conducted to establish the data gathering techniques used in the primary studies, these are listed in Table 8. Three studies used a literature review technique and three studies reported the use of action research. Six studies adopted a quantitative approach using surveys and descriptive statistics. Semi-structured interviews, single case and multiple case studies were used in 6 studies that adopted a qualitative method. Five primary studies that adopted a mixed method used a combination of surveys, learning diaries, focus groups, semi-structured interviews, single case and multiple case studies.

Table 8: Data collection techniques

Method	Techniques	Primary study	n
Action Research	• Action research	P11, P19, P20	3
Literature review	• Hermeneutics • Monographic • Systematic	P6, P7, P8,	3
Quantitative	• Survey • Descriptive statistics	P2, P4, P5, P10, P16, P21	6
Qualitative	• Semi-structured interviews • Single and multiple case study	P1, P3, P9, P13, P14, P18	6
Mixed method	• Survey • Learning diaries • Focus group • Semi-structured interviews • Single and multiple case study	P12, P15, P17, P22, P23,	5

Table 8 shows that qualitative research (6 studies) and quantitative research are the most popular methods for Kanban research, closed followed by mixed method research (5 studies). These three methods provide rich data on Kanban usage in an environment that is complex, highly contextual, and socially embedded, by using the case study technique. Within the mixed method category, 3 studies (P15, P17, P23) used the combination of survey and semi-structured interviews, one study (P12) used survey and learning diaries, and one study (P22) used interviews and secondary data.

4.5 RQ1.5 Contributions of primary studies

The aim of this research question is to identify and categorise the contributions of published Kanban studies. The contribution of the primary studies is based on six types of contributions proposed by Shaw (2003) and Paternoster et al (2014), namely, (i) framework, method, technique, (ii) guidelines, (iii) lessons learned, (iv) model, (v) tool, and (vi) advice/implication. A description of each contribution type is listed in Table 9.

Table 9: Contribution type (adapted from Shaw, 2003; Paternoster et al., 2014)

Title	Description
Framework/ Method/ Technique	The contribution of the study is a particular framework, method, or technique used to facilitate the construction and management of software and systems.
Guidelines	A list of advice or recommendations based on synthesis of the obtained research results.
Lessons Learned	The set of outcomes directly based on the research results obtained from the data analysis.
Model	The representation of an observed reality in concepts or related concepts after a conceptualization process.
Tool	A technology, program, or application that is developed in order to support different aspects of software engineering.
Advice/Implication	A discursive and generic recommendation based on personal opinion.

The contributions of the 23 primary studies, the research method and the data collection techniques that led to these contributions are listed in Table 10. Fifteen studies made a contribution that can be categorised as 'lessons learned', followed by 'advice or implications' (6 studies), and 'guidelines' (2 studies). Although the contribution type of each paper could be considered to overlap with another contribution type, the categorisations used in this systematic mapping study are based on the contribution type as stated by the authors in each of the 23 primary papers.

Table 10 clearly shows that 'lessons learned' (15 studies) remains the most dominant contribution type of Kanban research, followed by 'advice/implications' (6 studies), and then 'guidelines' (2 studies). However, a limitation of these three types of contributions is that they are context-specific and may not be applicable to other environments. Further, there is frequently a repetition of the lessons learned, implications, and guidelines in these studies, which indicates a lack of cumulative building of knowledge across the respective studies.

Table 10 reveals that 20 studies used case studies, of which 16 were single case study and 4 studies, used multiple case studies. Three studies used literature reviews that were systematic, Hermeneutics, or monographic.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Table 10: Contributions, method and data collection techniques across studies

Primary paper	Contribution type	Research Method	Data collection technique	Analysis technique
1	Lessons learned	Multiple case study	Semi-structured interviews	Thematic analysis
2	Lessons learned	Single case study	Survey	Descriptive statistics
3	Implications	Multiple case study Mixed method	Snowballing Semi-structured interviews	Template analysis
4	Implications	Multiple case study	Survey	Descriptive statistics
5	Lessons learned	Longitudinal	Survey	Descriptive statistics
6	Advice	Literature review	Systematic literature review	Thematic analysis
7	Guidelines	Literature review	Hermeneutics	Content survey
8	Lessons learned	Literature review	Monograph	Thematic analysis
9	Lessons learned	Multiple case study	Semi-structured interviews	Thematic analysis
10	Implications	Longitudinal single case study	Source code repository	Statistical analysis -Erlang-C model
11	Guidelines	Single case study	Action research	Statistical analysis
12	Lessons learned	Single case study	Survey Learning diaries	Statistical analysis
13	Lessons learned	Single case study	Semi-structured interviews	Thematic analysis
14	Implications	Single case study Mixed method	Video and direct observation Thematic, semi structured interviews	Thematic analysis
15	Implications	Multiple case study Mixed method	Semi-structured interviews Survey	Statistical analysis Thematic analysis
16	Lessons learned	Single case study	Survey	Descriptive statistics
17	Lessons learned	Single case study	Direct observations	Statistical analysis
18	Lessons learned	Single case study	Thematic, semi-structured interviews	Constant comparison method
19	Lessons learned	Single case study	Action research	Thematic analysis
20	Lessons learned	Single case study	Action research Direct and participant observations	Statistical analysis
21	Lessons learned	Single case study	Source code repository	Statistical analysis
22	Lessons learned	Single case study	Survey Focus groups	Descriptive statistics
23	Lessons learned	Single case study	Semi-structured interviews	Thematic Analysis

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Table 10 highlights a need for Kanban research to contribute to the categories of (i) frameworks/method/technique, (ii) model, and (iii) tool. This would provide significant practical contributions, as well as widening the academic discourse on Kanban use in software engineering. The quality of primary papers is presented in the next section.

4.6 RQ1.6 Quality of primary papers

The aim of this research question is to establish the quality of published Kanban studies. To achieve this aim, quality of each of the 23 primary papers was assessed independently by at least two authors using the 11-factor framework proposed by Dybå and Dingsøyr (2008). This was followed by in depth discussion and comparison of findings between both researchers. The aggregate of this quality assessment is presented in Table 11.

All of the 23 studies were ranked 1 on the first criterion and all studies provided a clear research aim and all had a form of description of the context in which the research was conducted. However, the research design of one paper was not sufficiently discussed. As three papers were systematic literature reviews, sampling was not applicable. As controlled experiments were excluded from this study, no control group with which to compare treatments was applicable for the primary papers. All 23 primary papers adequately described the ‘data collection’ and ‘data analysis’ and the ‘finding’ and ‘value’ of all papers was appropriate. Twelve papers were not explicit about considering the relationship between researcher and participants (e.g. reflexivity). None of the papers got a full score on the quality assessment and 12 papers were rated with two or three negative answers.

Although the quality ranking of these 23 primary studies may appear high, it is worth noting that the publication channel of the respective studies are a reflection of the high quality of research expected from these channels, which all conduct a peer-review process.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

Table 11: Quality assessment of primary papers

Code ID	Research	Aim	Context	Design	Sampling	Control	Data Collection	Reflexivity	Finding	Value	Total
P1	1	1	1	1	1	0	1	1	1	1	1
P2	1	1	1	1	1	0	1	1	1	1	1
P3	1	1	1	1	1	0	1	1	1	1	1
P4	1	1	1	1	1	0	1	1	1	1	1
P5	1	1	1	1	1	0	1	1	0	1	1
P6	1	1	1	1	0	0	1	1	0	1	1
P7	1	1	1	1	0	0	1	1	1	1	1
P8	1	1	1	1	0	0	1	1	0	1	1
P9	1	1	1	1	1	0	1	1	1	1	1
P10	1	1	1	1	1	0	1	1	0	1	1
P11	1	1	1	1	1	0	1	1	0	1	1
P12	1	1	1	1	1	0	1	1	1	1	1
P13	1	1	1	0	1	0	1	1	0	1	1
P14	1	1	1	1	1	0	1	1	1	1	1
P15	1	1	1	1	1	0	1	1	0	1	1
P16	1	1	1	1	1	0	1	1	0	1	1
P17	1	1	1	1	1	0	1	1	0	1	1
P18	1	1	1	1	1	0	1	1	1	1	1
P19	1	1	1	1	1	0	1	1	1	1	1
P20	1	1	1	1	1	0	1	1	1	1	1
P21	1	1	1	1	1	0	1	1	0	1	1
P22	1	1	1	1	1	0	1	1	0	1	1
P23	1	1	1	1	1	0	1	1	1	1	1
P24	1	1	1	1	1	0	1	1	0	1	1
Total	24	24	24	23	21	0	24	24	12	24	24

4.7 RQ1.7 Knowledge areas of studies on Kanban

The aim of this research question is to categorise studies on Kanban based on key knowledge areas emerging from the papers being studied (c.f. Petersen et al., 2015). Twenty of the 23 primary studies are categorised in the knowledge area of 'software engineering process' and 3 studies in the category of 'software engineering management and economics' (see Table 12).

Table 12: Knowledge areas of Kanban research

Knowledge areas	Description	Primary study source
Software engineering process	Is concerned with work activities accomplished by software engineers to develop, maintain, and operate software, such as requirements, design, construction, testing, maintenance, configuration management, and other software engineering processes.	P1, P2, P4, P5, P6, P7, P8, P9, P11, P12, P13, P14, P15, P16, P18, P19, P20, P21, P22, P23
Software engineering management and economics	Is about making decisions related to software engineering in a business context. It is concerned with aligning software technical decisions with the business goals of the organization.	P3, P10, P17

The scarcity of Kanban research within the three other knowledge areas (e.g. software maintenance, software engineering management, software engineering economics) would indicate that Kanban research in software engineering is currently restricted to project level as Kanban has not yet been scaled to portfolio project level or being used as a tool for decision-making by management.

4.8 RQ2 Reported benefits of Kanban

The aim of this research question is to identify the reported benefits when using Kanban in software engineering. The primary studies reported various benefits associated with the use of Kanban in the context of software engineering. This study distilled 15 types of benefits from the 23 primary studies, categorised them under three broad categories (e.g. process, people, and organisation), and mapped the associated studies to each reported benefit (see Table 13). We acknowledge that these benefits could be mapped to more than one category; however, to avoid complexity they were mapped to the most relevant category.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Table 13: Reported benefits of Kanban

Category	#	Reported benefit	Primary study
Process	1	Improve visibility and transparency	P1, P2, P3, P4, P5, P6, P7, P11, P13, P14, P15, P17, P19, P20, P22, P23
	2	Better control of project activities and tasks	P1, P2, P5 , P9, P10, P11, P13, P15, P19, P20, P22, P23
	3	Identify impediments to flow	P1, P2, P3, P5, P9, P15, P17, P20, P22, P23
	4	Improve workflow	P2, P4, P6, P11, P16, P19, P20
	5	Faster time-to-market	P6, P7, P10, P16, P23
	6	Improve prioritisation of products and tasks	P1, P3, P15, P17
	7	Decrease defects and bugs	P2, P7, P14, P21
	8	Improve quality	P6, P7, P16, P17
	9	A lightweight intuitive method	P14, P15, P16, P17
People	10	Improve communication and collaboration	P1, P4, P6, P7, P9, P14, P17
	11	Improve team motivation	P4, P6, P11, P16, P17, P19
	12	Team building and cohesion	P5, P7, P17, P20, P23
	13	Increase customer satisfaction	P6, P7, P14, P15, P17, P20
Organisation	14	Promoting a culture of continuous learning	P7, P10, P16, P20, P23
	15	Strategic alignment	P3, P5, P7

Process: Eighteen studies reported 9 benefits related to process improvement, and the four most frequently reported benefits were (i) improve visibility and transparency (16 studies), (ii) better control of project activities and tasks (12 studies), (iii) identify impediments to flow (10 studies) (iv) improve workflow (7 studies). Five other benefits reported within the process category are, (v) faster time-to-market (5 studies), (vi) improve prioritisation of products and tasks (4 studies), (vii) decrease defects and bugs (4 studies), (viii) improve quality (4 studies), and (ix) a lightweight intuitive method (4 studies).

People: Fourteen studies reported four benefits related to people. These were (i) improve communication and collaboration (7 studies), (ii) improve team motivation (6 studies), (iii) increase customer satisfaction (6 studies), (iv) team building and cohesion (5 studies), and (v) increase team satisfaction (6 studies). Although only three benefits are reported for this category, that are closely aligned and have a significant positive impact on team cohesion and moral.

Organisation: Eight studies reported two benefits of Kanban that were related to organisation, of which 6 studies reported 'promoting a culture of continuous learning', and 3 studies reported 'strategic alignment' as a benefit of Kanban. The reported benefits for this category are low when compared to the preceding categories and could be linked to challenges related to the category, which are discussed in the next section.

The reported benefits of Kanban in software engineering are predominantly process related (18 studies), followed by people (14 studies), and to a lesser degree organisation (8 studies). While these benefits

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study.** ***Journal of Systems and Software*****,** ***137*****, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

indicate that Kanban provides a range of benefits within the context of software engineering, it is not clear if other supporting techniques and organisation change initiatives contributed to these reported benefits. For example, value stream maps, cumulative flow diagrams (CFDs), and burn-down charts (c.f. Petersen et al., 2014) are also used by software engineering teams, as well as metrics such as cycle-time, lead time, and throughput (c.f. Reinertsen, 2009, Power and Conboy, 2015).

4.9 RQ3 Reported challenges of Kanban

The aim of this research question is to identify the reported challenges when using Kanban in software engineering. Eleven of the 23 primary studies reported 8 key challenges associated with the use of Kanban in software engineering. This study has distilled these challenges and broadly categorised them into three broad categories, namely process, people, and organisation (see Table 14). As highlighted previously, we acknowledge that these challenges could be mapped to more than one category; however, to avoid complexity they were mapped to the most relevant category.

Table 14: Challenges of Kanban usage

Category	#	Challenge	Primary study source
Process	1	Setting up and maintaining Kanban	P4, P6, P9, P12, P17, P18
People	2	Management not ready for new method	P6, P9, P10, P23, P17
	3	Poor understanding of Kanban concepts and practices	P4, P6, P7, P17
	4	Managed communication between teams and customer	P6, P15
Organisation	5	Changing organisational culture	P4, P6, P15, P17, P18, P22
	6	Lack of supporting practices around the use of Kanban	P6, P7, P14, P15, P16
	7	Lack of training	P4, P5, P6, P9, P14
	8	Poor knowledge management	P6

Process: Six studies reported only one process related challenge, ‘setting up and maintaining Kanban’. The relatively low number of studies reporting this as a challenge would suggest that Kanban is suited to software engineering but organisations need to allocate appropriate time for software teams to iteratively design and maintain Kanban, and to embed this process within operations..

People: Eight studies reported three people related challenges, of which (i) ‘management not ready for new method’ was the most frequently reported challenge (5 studies), followed by (ii) ‘poor ‘understanding of Kanban concepts and practices’ (4 studies), and (iii) ‘managed communication between teams and the customer’ (2 studies). These challenges could explain why setting up and maintaining Kanban is challenging as software teams lack the appropriate supported for guided and self-directed learning. These challenges also highlight the ‘lack of readiness’ by management to adopt Kanban, this in turn would suggest that teams adopting or piloting Kanban lack the support of management, which is a greater challenge than the actual adoption of Kanban itself.

Organisation: Eleven studies reported four key challenges of Kanban related to the organisation, namely (i) changing organisational culture (6 studies), (ii) lack of supporting practices around the use of Kanban

(5 studies), (iii) lack of training (5 studies), and (iv) poor knowledge management (1 study). Although only four key challenges were reported for this category. This category received the highest degree of reported challenges to Kanban in software engineering.

The reported challenges of Kanban in software engineering are predominantly related to the organisation (11 studies), followed by people (8 studies), and to lesser degree process (6 studies). This finding is very interesting as it indicates that not only did the category 'organisation' encounter the highest number of challenges, this being four (11 studies), but only two benefits were reported (8 studies). In contrast, only one process related challenge was reported (6 studies), but nine benefits were reported (18 studies) for this category. While the people category falls in the middle, with three reported challenges (8 studies) and four reported benefits (14 studies).

4.10 RQ4 Insights gained from Kanban experience reports

This section provides insights from 23 Kanban experience reports between 2006 and 2016. Experience reports were excluded from the systematic mapping study, as they lack research rigor and are context-specific, which makes the findings difficult to interpret and generalize. However, we acknowledge that Kanban experience reports are appealing to Kanban practitioners because these reports act as a source of reference for practitioners. For example, Neely and Stolt (ER 23) report on their transition to continuous delivery with Kanban at Rally Software and the benefits realised (e.g. greater control and flexibility over feature releases, fewer defects, easier on-boarding of new developers, and increased confidence of team members). Maassen and Sonnevelt (ER19) report on using Kanban for IT maintenance and operations at a European insurance company and the benefits realised (e.g. improved understanding and cooperation between developers and testers working on different technologies). In software development the Kanban board describe workflow well and helps to modify tasks or update the Kanban board without waiting for the next iteration (ER3). Organisations are also leveraging Kanban to visualize HR work, entire IT project portfolios, and set constraints on projects by setting WIP limits at project level (ER2, ER6, ER7) in a Finnish broadcasting company (ER2), Kanban board work as a roadmap to visualize all the activities to management and helps them to make decisions more realistically. Wijewardena (ER11) reports on the adoption of Kanban at a human resource department of a mid-sized, offshore, software development company (Exilesoft) and reported benefits such as increased visibility to work and improved workflow. Additionally, Kanban facilitates management to take joined decisions and look for improvement opportunities (ER2). Other reports on Kanban by established practitioners (e.g. Anderson and Roock, 2011; Leffingwell 2010; Shalloway 2011) make claims that Kanban is the easiest tool to use for project portfolio management, and it enables managers to make appropriate decisions about tasks based on business value (Shalloway, 2011).

The 23 experience reports were analysed in order to draw insights that may not have been identified in the primary papers. The remainder of this section presents the insights gained from the review of Kanban experience reports published between 2008 and 2016 (see Appendix B). Table 15 shows that publication of experience reports on Kanban peaked between 2010 and 2013 but such reports have since declined.

Table 15. Experience reports on Kanban per year

Experience report ID	Year	n
ER1	2016	1
ER2	2015	1
ER3	2014	1
ER4, ER5, ER6, ER7	2013	5
ER8, ER9, ER10	2012	3
ER11, ER12, ER13, ER14,	2011	4
ER15, ER16, ER17, ER18, ER19	2010	5
ER20, ER21	2009	2
ER22	2008	1
Total		23

A deeper analysis of the 23 experience reports (see Table 16) reveals that 15 experience reports focused on Kanban use in software development environments, four reports focused on Kanban in software maintenance and four reports focused on Kanban use in software portfolio project management.

Table 16. Domain of Kanban implementation

Experience report	Domain of Kanban implementation	n
ER4, ER5, ER6, ER7, ER8, ER9, ER12, ER15, ER16, ER17, ER18, ER20, ER21, ER22	Software development	15
ER1, ER13, ER14, ER19	Software maintenance	4
ER2, ER3, ER10, ER11	Software project portfolio management	4
	Total	23

The reported benefits of Kanban in experience reports are presented in the next section.

4.11 RQ4.1 Reported benefits of Kanban from experience reports

The experience reports identified nine benefits of using the Kanban; these are listed in Table 17 and the associated experience report. Six of the reported benefits are related to process, of which there are two dominant benefits, namely (i) visibility facilitates and support the decision-making process (n=15 reports), and (ii) developing continuous improvements strategies and better workflow (n=15 reports). To a lesser degree, better understanding of the entire development process (n=6 reports) and increasing the predictability in the delivery of the final products and more precise estimate of the work (n=6 reports) were reported. Reducing cycle time and lead time (n=4 reports) and better workload balance (n=2 reports) were also reported benefits.

Table 17: Summary of Kanban benefits from experience reports

Category	Benefits	Experience reports
Process	Visibility facilitates and support the decision-making process	ER1, ER2, ER6, ER7, ER10, ER11, ER12, ER14, ER15, ER16, ER17, ER19, ER20, ER22, ER23
	Developing continuous improvements strategies and better workflow	ER1, ER2, ER5, ER6, ER8, ER9, ER10, ER14, ER15, ER16, ER18, ER19, Er20, ER21, ER23
	Better understanding of entire development process	ER6, ER7, ER12, ER15, ER16, ER23
	Increasing the predictability in the delivery of the final products and more precise estimate of the work	ER3, ER6, ER7, ER12, ER13, ER22
	Reducing cycle time and lead time	ER10, ER12, ER17, ER21
	Better workload balance	ER5, ER23
People	Ensuring skills development and cohesiveness of teams	ER6, ER7, ER10, ER11, ER12, ER13, ER14 , ER23
Organization	Facilitate coordination and impose self-organization	ER1, ER2, ER12, ER13, ER10, ER14, ER21, ER23
	Driving and facilitating organizational change management	ER6, ER12, ER9, ER11, E13, ER20, ER23

In terms of organisation related benefits, two benefits were reported, namely, facilitate coordination and impose self-organization (n=8 reports), and driving and facilitating organizational change management (n=7 reports). One benefit was reported that related to people, this being, ensuring skills development and cohesiveness of teams (n=8 reports). The reported challenges in Kanban use are presented in the next section.

4.12 RQ4.2 Challenges in Kanban use from experience reports

Eight challenges of Kanban use are reported in the experience reports (Table 18). Four challenges related to the organisation, of which 11 reports highlighted that Kanban requires integration with existing agile

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

techniques, which can be complicated, expensive, and time-consuming. Other challenges reported were, changing organisational culture (n=4 reports), lack of specialised skills and training (n=4 reports), and Kanban implementation requires deeper understanding of Lean (n=1 report).

Two reported challenges related to process, namely, lack of guidelines for Kanban implementation guidelines (n=4 reports), and assessing performance-using metrics such as lead-time (n=1 report). Two reported challenges related to people, these being, motivating staff to adopt new practices (n=7 reports) and task switching and unpredictable flow of work (n=1 report).

Table 18: Reported challenges of Kanban usage from experience reports

Category	**Challenges**	**Experience report ID**
Process	Lack of guidelines for understanding Kanban and its implementation	ER1, ER12, ER9, ER21
	Assessing performance using metrics (e.g. lead -time)	ER21, ER14
People	Motivating staff to adopt new practices	ER1, ER4, ER6, ER14, ER19, ER20, ER21
	Task switching and unpredictable flow of work	ER14
Organisation	Kanban requires integration with existing agile techniques, which can be complicated, expensive, and time-consuming.	ER1, ER3, ER7, ER8, ER12, ER13, ER14, ER19, ER20, ER21, ER22
	Changing organisational culture	ER16, ER17, ER19, ER20
	Lack of specialised skills and training	ER5, ER6, ER17, ER21
	Kanban implementation requires deeper understanding of Lean	ER22

Having identified the reported challenges of Kanban usage from experience reports, the findings of the final research question are presented in the next section.

4.13 RQ5 Recommendations for Kanban use from empirical studies and experience reports.

As previously stated, this study is unique as it summarizes the recommendations for Kanban use based on empirical studies and experience reports. Fourteen recommendations were identified in the 23 primary papers and 10 recommendations identified in the 23 experience reports. These are listed in Table 19.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Table 19: Recommendations for Kanban use in practice

Primary papers	**Experience reports**
1. Kanban helps to visualize tasks but visualization alone does not replace concrete actions or guarantee success. 2. When a task not progressing it is better to use pair programming technique. As a result, work can be complete efficiently and led team members to work on diverse tasks and broaden their working domain area. 3. Encourage team members to provide feedback to each other. 4. All relevant stakeholders including senior management should agree the WIP limits. 5. Enforce WIP limit strictly, it will help team members to focus on and control their work. 6. The proactive role of team leaders is essential when using Kanban. 7. Cultivate a culture of continuous delivery as it enables teams to be more proactive when high-priority work comes in, rather than waiting for an iteration to complete. 8. Keep daily stand-up meetings regular as this provides up-to-date information about work to all stakeholders; mitigate knowledge loss and facilitates knowledge flow. 9. Make Kanban transition incremental rather than a radical implementation. 10. Educate staff about new software approaches through specialised training. 11. Organization's readiness to the process transition needs to be assessed prior to determining the transition strategy and designing the process transition. 12. Prioritization of tasks can be based upon its value, urgency, importance, and cost of delay or resources	1. Sufficient time is essential for process transition. Allow teams to sufficient time and effort to reflect on problems and come up with an action plan that would improve their process. 2. Identify a dedicated team to pilot Kanban and then build on this learning experience 3. Share the successes and failures of Kanban throughout the organization. 4. Organizations should take the Kanban transition as a serious challenge, and find means such as agile coaching in order to help teams and managers in process transformation. 5. Organizations should create an internal change team that could help focus on sustaining a continuous improvement culture that is supported by management. 6. Empower teams to lead. 7. It is better to synchronize Kanban with other agile processes. 8. First in first out (FIFO) queue process helps to keep track of each defect or maintenance tasks as it enters the development process and teams can see how long it takes to fix a defect, which in turn helps to achieve better predictability. 9. Systematic use of PDCA cycle, A3 problem solving technique and 5 why root cause analysis helps to identify problems and provide improvement opportunities for the entire organisation.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

available. 13. Various tools can be used for performance measurement (i.e. CFDs, burn-down charts). 14. Organisations should clearly communicate software process policies to all stakeholders.	

Table 19 highlights the value that experience reports provide to practitioners using Kanban and those that intend to use it. Although the academic rigor of such reports may not be equivalent to academic written papers, experience reports do provide rich insights and business value to practitioners who work in complex real world environments.

5. Discussion

By using a systematic mapping method, we identified, classified, and analysed 382 studies on Kanban in software engineering were published between 2006 and 2016. Of these, 23 studies were identified as primary studies, as the reported research was found to be within the criteria of this study - acceptable academic rigour, credible, and relevant. Interest in Kanban research has slightly increased in recent years.

Qualitative research (6 studies) and quantitative research (6 studies) were the most popular method of the primary studies. The combination of survey and interview was the dominant techniques used in mixed method and interviews were the dominant techniques used in qualitative research. While these methods do provide very rich and in depth data (Adam and Healy, 2000), the maturity of the studied cases was not explicit (e.g. when was Kanban initially adopted and how frequently was Kanban used). In addition, there were no longitudinal research studies on the adoption Kanban. Yet, the realities of adoption within organisations are that adoption decisions are generally made at the organisation, departmental, or workgroup levels, rather than at the individual level (Orlikowski, 1993; Fichman and Kemerer, 1997). In addition, adoption of methods such as Kanban is not a binary activity that occurs in a short time frame, but rather a number of adoption phases (Gallivan, 2001), namely, (i) initiation, (ii) adoption, (iii) adaptation, (iv) acceptance, (v) routinisation, and (vi) infusion, which can take a number of months or years to achieve.

In terms of identifying the type of contribution that research on Kanban has made over the past 10 years, the primary papers were categorised using the frameworks adapted from Shaw, (2003) and Paternoster et al., 2014). The majority of primary studies (18) provided a contribution type that were categorised as ‘lessons learned’, followed by ‘advice or implications’ (3 studies), and ‘guidelines’ (2 studies). While these findings provide support for organisations considering adopting Kanban, there remains a stubborn lack of empirical studies that provide more practical support in the form of a (i) framework/method/technique, (ii) model, or (iii) tool, which can complement Kanban adoption.

Using the 11 factor quality assessment framework proposed by Dybå and Dingsøyr (2008), the aggregated reporting quality of the primary studies were of a high standard, specifically in the categories of ‘data collection’, ‘data analysis’, ‘finding’ and ‘value’. Further evidence of the quality of primary studies is reflected in the publication channels (e.g. journals and conferences).

A concern identified in this study is the lack of primary studies (12) that did not explicitly explain how they addressed threats to validity (c.f. Petersen et al., 2015; Wohlin et al., 2012; Runeson and Höst, 2009; Kitchenham et al., 2002). Further, of the 11 primary studies that did discuss how threats to validity, it was not always clear what framework (e.g. Petersen et al., 2015; Wohlin et al., 2012; Kitchenham et al., 2002) were used to mitigate these threats or if all elements of a specific framework were followed.

The primary studies were categorised into two thematic knowledge areas, software engineering process (20 studies) and software engineering management and economics (3 studies). As Kanban has traditionally been associated with operational activities (process), it is not unusual to have a dominant application domain of Kanban, and subsequently Kanban research within the knowledge area of software engineering process.

In terms of providing a definition of Kanban, this mapping study identified a lack of cohesion across studies as seven definitions were used. This raises a concern, that in the long term, Kanban studies in software engineering could lack a tradition of cumulative building of knowledge (c.f. Fitzgerald and Adam, 2000), which resonates with the issue of ‘fragmented adhocracy’, and we know from existing research (c.f. Conboy, 2009; Banville and Landry, 1989; Hirschheim and Lyytinen, 1996) has overshadowed related disciplines. This lack of cohesion was witnessed with the concept of ‘agility’ and addressed by Conboy (2009) who adopted a ‘first principles’ approach to the development of a contemporary and universally accepted definition of agility in the context of software development.

Fifteen reported benefits of Kanban in software engineering were identified of which nine were process related, four were people related, and two were organisation. In contrast, nine benefits were reported in the experience reports, of which six were process related, two were organisation, and one people related.

Shared benefits reported from the empirical and experience reports included, (i) increased visibility, (ii) improved work flow, (iii) faster time to market, and (iv) team building and cohesion. While these benefits suggest that Kanban is well suited to the complex and highly contextual nature of software engineering at project level, there remains a limited reported benefit of Kanban for project portfolio management.

Eight challenges were reported in the primary studies, of which one was process related, four were people related, and four were organisation related. Setting up and maintaining Kanban (6 studies) and changing organisational culture (6 studies) were the most reported challenges in the primary studies. Eight challenges were also identified in the experience reports, of which two were process related, two were people related, and four were organisation related. Although changing organisational culture was reported in these reports (4 reports), the most frequently reported challenge (11 reports) was that Kanban requires integration with existing agile techniques, which can be complicated, expensive, and time-consuming. The next most reported challenge was motivating staff to adopt new practices (7 reports).

As acknowledged previously, the categorising of the sixteen reported challenges of Kanban could be mapped to more than one category. Nevertheless, the challenges have implications for practice. Specifically the lack of readiness by management to adopt Kanban could be a symptom of deeper organisational issues. For example, the organisational culture is not conducive to individual and team learning, a culture of blame exists, or the organisation has not established a process for analysing, describing, and integrating method rationale (c.f. Agerfalk and Wistrand, 2003). From a practice perspective, organisations should ensure that the Kanban method is considered within this wider method portfolio. Therefore, before measuring the benefits of Kanban at a team level, it is important to determine whether the method itself is suitable in that instance and if so, enactment of Kanban practices need to be implemented by both software and management teams.

In terms of the recommendations for practice, fourteen recommendations were identified in the 23 primary papers and 10 recommendations identified in the 23 experience reports. A common theme between both types of studies was the emphasis on allowing time for the adoption of Kanban to become embedded in the organisation by creating a culture of organisational learning. To achieve such learning, organisations need to shift from a culture of 'error-free learning' to a culture of 'double loop' and 'triple loop' learning (c.f Argyis, 1976; Roper and Petit, 2002), where piloting of Kanban (c.f. Ahmad et al., 2016) is encouraged, lessons are learned and communicated across projects and to project portfolio level. Failure to communicate lessons learned from piloting Kanban can result in an organisation experiencing 'learning disabilities' (c.f. Schein, 1996) which occur when a new method of learning does not diffuse or become embedded in the organisation, this then gets in the way of second order learning (i.e. an individual project may learn new methods but these methods do not diffuse to other groups within the organisation).

6. Validity threats and limitations of the study

There are always threats to the validity of a study (c.f. Petersen et al., 2015; Wohlin et al., 2012; Runeson and Höst, 2009; Kitchenham et al., 2002). This section discusses these threats and outlines the strategies used to mitigate their effects, as well as the limitations of this study. In order to evaluate the validity of this study, the authors have used the validity framework presented by Wohlin et al., (2012) which addresses (i) construct validity, (ii) external validity, (iii) external validity, and (iv) conclusion validity.

Construct validity relates to obtaining the right measures for the concept being studied (Petersen et al., 2015; Wohlin et al., 2012; Runeson and Höst, 2009). To reduce this threat, a data collection process was designed (Figure 1) to objectify paper selection (e.g. inclusion and exclusion) and data extraction (Figure 2) from the 23 primary papers to support the recording of data. To further mitigate this threat, author three and four were experienced in mapping studies and acted as external reviewers to validate the research protocol. Hence, this threat has been significantly minimised. *External validity* relates to the extent to which the study results are generalisable (Petersen et al., 2015; Wohlin et al, 2015). In order to know what degree the results of a study can be generalised, it is extremely important to describe the research context (Petersen and Wohlin, 2009; Kitchenham et al., 2002). This threat is minimized in this study as a rigorous research methodology that followed guidelines by Petersen et al., (2008) and extracting data regarding the methodology (e.g., data collection procedures) was conducted following guidelines by Petersen et al., (2015) and Dybå and Dingsøyr (2008). *Internal validity* relates to causal relationships and ensuring that it is not a result of a factor that was not measured or the researcher had no control over. As the aim of the study was not to establish a statistical causal relationship on Kanban, it is not considered a threat to this study. *Conclusion validity* relates to bias of the researchers in the interpretation of that data. While this risk cannot be eliminated, it was reduced by taking following actions: (i) four researchers were involved in the analysis of the primary papers, (ii) a full 'audit trail' from retrieving 382 papers to identifying 23 primary papers was maintained, (iii) as highlighted previously, the 43 relevant papers were each read in full by at least two authors, and (iv) and the conclusions drawn from analysis of the 23 primary papers involved all four authors.

These four validity threats resonate with publication bias, which refers to the issue that research outcomes that are positive are more likely to be published than negative outcomes (c.f. Unterkalmsteiner et al., 2012). In this instance, its effect is minimal because the aim of the study is to present a state-of-the-art of research on Kanban. Nevertheless, we acknowledge that publication bias could have affected our results regarding the benefits and challenges of using Kanban. Publication bias can also affected by the sources of the data in a study and its publication channel. The four databases (e.g. ACM Digital Library, IEEE Xplore, ISI Web of Science, and Scopus - Sciencedirect) were used, as these sources are known to return the most publications and have been used in similar types of literature mapping exercises in software engineering (e.g. Dyba et al., 2007; Kitchenham and Brereton, 2013). Although the results of this mapping study are limited by scientific studies published in these databases, they covered a wide range of software engineering literature and closely related contexts (i.e. software development, information systems development). In addition, non-peer reviewed scientific studies, book, book chapters, short papers, experience reports, and assimilation studies were excluded. The raison d'etre for excluding these publications is (i) the data can be anecdotal, (ii) a lack of research rigor, and (iii) simulation studies do not reflect the human and contextual nature of software engineering in which Kanban is used.

7. Conclusion and directions for future research

This systematic mapping study provides a structured understanding of the state-of-the-art of Kanban research in software engineering. This was achieved by identifying 23 primary studies out of 382 related Kanban articles over a ten years period (2006 – 2016) and analysed them with respect to (i) frequency of publication by year, (ii) publication channels, (iii) research method, (iv) contribution type, (v) quality, (vi) knowledge area, (vii) definitions of Kanban, (viii) benefits, and (ix) challenges. In addition, 23 experience reports on Kanban published during the same period were analysed and insights in terms of benefits, challenges and recommendations for Kanban were extracted.

A clear finding emerging from this systematic mapping study is the need to (i) increase the number of rigorous academic studies on Kanban, (ii) be explicit about the validity threats to that study and how these were mitigated, and (iii) build on cumulative knowledge.

Although the benefits of Kanban identified in this study outweigh the challenges, Kanban by itself does not guarantee success as it is a relatively basic flow tool that needs to be supported by additional practices (Ikonen et al., 2011). Research on Kanban in software engineering remains largely unexplored, thereby offering the research community the opportunity to provide a contemporary perspective to Kanban and indeed valuable contributions to the knowledge base. For example, it is well accepted that a software method or technique cannot be studied in isolation (Conboy, 2009; Ebert et al., 2012; Fitzgerald et al., 2002; Kitchenham et al., 2002; Lyytinen and Rose, 2006; Petersen and Wohlin, 2009). This indicates a need, not just to study Kanban as a single method to improve the flow of work, but to include other commonly known complementary techniques that are used to manage the workflow, namely, value stream maps, cumulative flow diagrams, burn-down charts, and line of balance status charts (c.f. Petersen et al., 2014). Key metrics such as cycle-time, lead time, and throughput (c.f. Reinertsen, 2009, Power and Conboy, 2015) are also used to manage software engineering projects. Future research on Kanban could also explore how Kanban could be integrated with contemporary business intelligence and analytics software that can collect, analyse, and communicate real-time data to project teams and management teams. There is also a scarcity of knowledge on the temporal elements of Kanban and indeed within the wider discipline of software engineering. Finally, while the opportunities to conduct high quality research of Kanban in software engineering are limitless, its theoretical development will be limited if future research on Kanban does not adopt a tradition of cumulative building of knowledge.

Acknowledgements

This research was performed within the DIMECC (Digital, Internet, Materials & Engineering Co-Creation) Need for Speed program and was partially funded by Tekes (the Finnish Funding Agency for Technology and Innovation).

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

Appendix A: Primary studies

P1. Ahmad MO., Kuvaja P., Oivo M. and Markkula J., 2016. Transition of software maintenance teams from Scrum to Kanban. In System Sciences. 49th Hawaii International Conference on Information Systems: 5427-5436.

P2. Ahmad MO., Liukkunen K., and Markkula J., 2014. Student perceptions and attitudes towards the software factory as a learning environment. International conference on Global Engineering Education: 422-428.

P3. Ahmad MO., Lwakatare LE., Kuvaja P., Oivo M., and Markkula J., 2016. An empirical study of portfolio management and Kanban in agile and lean software companies. Journal of Software: Evolution and Process.

P4. Ahmad M O., Markkula J., and Oivo M., 2016. Insights into the perceived benefits of kanban in software companies: practitioners' views. International Conference on Agile Software Development: 156-168.

P5. Ahmad MO., Markkula J., and Oivo M., 2014. Kanban for software engineering teaching in Software Factory learning environment. World Transactions on Engineering and Technology Education: 12(3): 338-343.

P6. Ahmad MO., Markkula J., and Oivo M., 2013. Kanban in software development: A systematic literature review. 39th EUROMICRO Conference on Software Engineering and Advanced Applications: 9-16).

P7. Al-Baik O., and Miller J., 2015. The kanban approach, between agility and leanness: a systematic review. Journal of Empirical Software Engineering: 20(6): 1861-1897.

P8. Corona E., and Pani FE., 2013. A review of lean-kanban approaches in the software development. Transactions on Information Science and Applications: 10(1):1-13.

P9. Dennehy, D. and Conboy, K., 2016. Going with the flow: An activity theory analysis of flow techniques in software development. Journal of Systems and Software.

P10. Fitzgerald B., Musiał M. and Stol KJ., 2014. Evidence-based decision making in lean software project management. 36th International Conference on Software Engineering Companion: 93-102. ACM.

P11. Harzl, A., 2016, May. Combining FOSS and kanban: An action research. In IFIP International Conference on Open Source Systems: 71-84.

P12. Heikkilä VT., Paasivaara M. and Lassenius C., 2016. Teaching university students Kanban with a collaborative board game. 38th International Conference on Software Engineering Companion: 471-480.

P13. Ikonen M., Kettunen P., Oza N., and Abrahamsson P., 2010. Exploring the sources of waste in kanban software development projects. 36th EUROMICRO Conference on Software Engineering and Advanced Applications: 376-381.

P14. Ikonen M., Pirinen E., Fagerholm F., Kettunen P. and Abrahamsson P., 2011. On the impact of kanban on software project work: An empirical case study investigation. 16th International Conference on Engineering of Complex Computer Systems: 305-314.

P15. Law E L C., and Lárusdóttir M K., 2015. Whose experience do we care about? analysis of the fitness of scrum and kanban to user experience. International Journal of Human-Computer Interaction: 31(9): 584-602.

P16. Mahnic V., 2015. From Scrum to Kanban: introducing lean principles to a software engineering capstone course. International Journal of Engineering Education.

P17. Middleton P., and Joyce D., 2012. Lean software management: BBC Worldwide case study. IEEE Transactions on Engineering Management: 59(1): 20-32.

P18. Tripathi N., Rodríguez P., Ahmad MO., and Oivo M., 2015. Scaling kanban for software development in a multisite organization: challenges and potential solutions. International Conference on Agile Software Development, Springer International Publishing: 178-190.

P19. Nikitina N., Kajko-Mattsson M., and Stråle M., 2012. From scrum to scrumban: A case study of a process transition. International conference on software and system process: 140-149.

P20. Nikitina N., and Kajko-Mattsson M., 2011. Developer-driven big-bang process transition from Scrum to Kanban. International conference on software and systems process: 159-168.

P21. Sjøberg DI., Johnsen A. and Solberg J., 2012. Quantifying the effect of using kanban versus scrum: A case study. IEEE software: 29(5): 47-53.

P22. Rodríguez P., Partanen J., Kuvaja P. and Oivo M., 2014. Combining lean thinking and agile methods for software development: A case study of a Finnish provider of wireless embedded systems detailed. 47th Hawaii International Conference on System Sciences: 4770-4779.

P23. Senapathi M., Middleton P., and Evans G., 2011. Factors affecting effectiveness of agile usage–insights from the BBC Worldwide case study. International Conference on Agile Software Development, Springer Berlin Heidelberg: 132-145.

Appendix B: Kanban experience reports

ER1: McCalden, S., Tumilty, M., and Bustard, D. (2016). Smoothing the transition from agile software development to agile software maintenance. In International Conference on Agile Software Development. Springer International Publishing: 209-216

ER2: Laanti, M., and Kangas, M. (2015). Is agile portfolio management following the principles of large-scale agile? Case study in Finnish Broadcasting Company Yle. In IEEE Agile Conference. 92-96.

ER3: Parker, M. E. F., and del Monte, Y. F. (2014). The agile management of development projects of software combining scrum, kanban and expert consultation. In OSS. 176-180.

ER4: Raju, H. K., and Krishnegowda, Y. T. (2013). Kanban pull and flow-a transparent workflow for improved quality and productivity in software development. IET,Fifth International Conference on Advances in Recent Technologies in Communication and Computing. 44 – 51.

ER5: Laanti, M. (2013). Agile and wellbeing--stress, empowerment, and performance in scrum and kanban teams. IEEE 46th Hawaii International Conference on System Sciences. 4761-4770.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

ER6: Hui, A. (2013). Lean change: enabling agile transformation through lean startup, kotter and kanban: an experience report. In IEEE Agile Conference. 169-174.
ER7: Wang, X., Conboy, K., and Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. Journal of Systems and Software. 85(6), 1287-1299.
ER8: Fernandes, C. (2012). There and back again: from iterative to flow and back to iterative. In IEEE Agile Conference. 103-110.
ER9: Terlecka, K. (2012). Combining Kanban and Scrum--lessons from a team of sysadmins. In IEEE Agile Conference. 99-102.
ER10: Mazzanti, G. (2012). Agile in the Bathtub: developing and producing bathtubs the agile way. In IEEE Agile Conference. 197-203.
ER11: Wijewardena, T. (2011). Do you dare to ask your HR manager to practice kanban? the experience report of an offshore software company in Sri Lanka introducing agile practices into its human resource (hr) department. In IEEE Agile Conference. 161-167.
ER12: Polk, R. (2011). Agile and Kanban in coordination. In IEEE Agile Conference. 263-268.
ER13: Greaves, K. (2011). Taming the customer support queue: a kanban experience report. In IEEE Agile Conferenc. 154-160.
ER14: Seikola, M., and Loisa, H. M. (2011). Kanban implementation in a telecom product maintenance. In 37th Euromicro conference on software engineering and advanced applications. 321-329.
ER15: Rutherford, K., Shannon, P., Judson, C., and Kidd, N. (2010). From chaos to kanban, via scrum. Agile Processes in Software Engineering and Extreme Programming. 344-352.
ER16: Birkeland, J. O. (2010). From a timebox tangle to a more flexible flow. In XP conference. 325-334.
ER17: Taipale, M. (2010). Huitale–a story of a Finnish lean startup. Lean Enterprise Software and Systems. 111-114.
ER18: Greening, D. R. (2010). Enterprise scrum: scaling scrum to the executive level. In IEEE 43rd Hawaii International Conference on System Sciences. 1-10.
ER19: Maassen, O., and Sonnevelt, J. (2010). Kanban at an insurance company (are you sure?). Agile processes in Software Engineering and Extreme Programming. 297-306.
ER20: Willeke, E. R. (2009). The Inkubook experience: a tale of five processes. In IEEE Agile Conference. 156-161.
ER21: Shinkle, C. M. (2009). Applying the dreyfus model of skill acquisition to the adoption of Kanban systems at software engineering professionals (SEP). In IEEE Agile Conference. 186-191.
ER22: Kinoshita, F. (2008). Practices of an agile team. In IEEE Agile Conference. 373-377.
ER23: Neely, S., and Stolt, S. (2013). Continuous delivery? easy! just change everything (well, maybe it is not that easy). In IEEE Agile Conference. 121-128.

Bibliography

Abrahamsson P., Conboy K., and Wang X., 2009. Lots done, more to do: the current state of Agile systems development research. European Journal of Information Systems 18(4): 281–284.
Ågerfalk PJ., and Wistrand K., 2003. Systems development method rationale-a conceptual framework for analysis. 5th International Conference on Enterprise Information Systems: 185–190.
Ahmad MO., Markkula J., and Oivo M., 2013. Kanban in software development: A systematic literature review. 39th EUROMICRO Conference on Software Engineering and Advanced Applications: 9-16.
Ahmad MO., Kuvaja P., Oivo M. and Markkula J., 2016. Transition of software maintenance teams from Scrum to Kanban. 49th Hawaii International Conference on Information Systems: 5427-5436.

Ahmad MO., Lwakatare LE., Kuvaja P., Oivo M., and Markkula J., 2016. An empirical study of portfolio management and Kanban in agile and lean software companies. Journal of Software: Evolution and Process.

Anderson D., Concas G., Lunesu MI., and Marchesi M., 2011. Studying lean-kanban approach using software process simulation. International Conference on Agile Software Development, Springer Berlin Heidelberg: 12–26.

Anderson D., 2010. Kanban: successful evolutionary change for your technology business. Sequim, Washington Blue Hole Press.

Anderson D., 2013. Lean software development. Lean Kanban University (LKU), Seattle.

Anderson D., and Roock A., 2011. An Agile evolution: why Kanban is catching on in Germany and around the world. Cutter IT Journal: 24: 6–17.

Al-Baik O., and Miller J., 2015. The kanban approach, between agility and leanness: a systematic review. Journal of Empirical Software Engineering: 20(6): 1861-1897.

Argyris C., 1976. Single-loop and double-loop models in research on decision making. Administrative science quarterly: 363-375.

Banville C., and Landry M., 1989. Can the field of MIS be disciplined? Communications of the ACM. 32: 48-60.

Becker M., and Szczerbicka H., 1998. Modeling and optimization of Kanban controlled manufacturing systems with GSPN including QN. International Conference on Systems, Man, and Cybernetics.1: 570-575.

Boeg J., 2012. Priming Kanban: A 10 step guide to optimizing flow in your software delivery system. 2nd edition. Aarhus Trifork, Chronografisk Margrethepladsen A/S Copenhagen.

Conboy K., 2009. Agility from first principles: reconstructing the concept of agility in information systems development. Information Systems Research: 20(3): 329-354.

Cocco L., Mannaro K., Concas G., and Marchesi M., 2011. Simulating Kanban and Scrum vs. Waterfall with system dynamics. International conference on Agile processes in software engineering and extreme programming: 117–131.

Chai L., 2008. E-based inter-enterprise supply chain Kanban for demand and order fulfilment management. International Conference on Emerging Technologies and Factory Automation: 33–35.

Cutter., 2011. The Viral Growth of Kanban in the Enterprise. Cutter Business Journal: 3-29.

Concas G., Lunesu MI., Marchesi M., and Zhang H., 2013. Simulation of software maintenance process, with and without a work-in-process limit. Journal of Software Evolution and Process: 25(12): 1225–1248.

Corona E., and Pani FE., 2013. A review of lean-kanban approaches in the software development. Transactions on Information Science and Applications: 10(1):1-13.

Dennehy, D. and Conboy, K., 2016. Going with the flow: An activity theory analysis of flow techniques in software development. Journal of Systems and Software.

Dybå T., and Dingsøyr T., 2008. Empirical studies of agile software development: a systematic review. Information and Software Technology: 50(9): 833–859.

Ebert C., Abrahamsson P., and Oza N., 2012. Lean software development. IEEE Software 5: 22–25.

Fichman RG., and Kemerer CF., 1997. The assimilation of software process innovations: An organizational learning perspective. Management science: 43(10):.1345-1363.

Fitzgerald B., Russo NL., and Stolterman E., 2002. Information systems development: methods in action: McGraw-Hill Education.

Gallivan MJ., 2001. Organizational adoption and assimilation of complex technological innovations: development and application of a new framework. ACM SIGMIS Database 32: 51-85.
Greaves K., 2011. Taming the Customer Support Queue. International conference on agile software development, Springer Berlin Heidelberg: 54–160.
Gross JM., and McInnis KR., 2003. Kanban made simple: demystifying and applying Toyota's legendary manufacturing process. New York, AMACOM.
Gravel M., and Price WL., 1988. Using the Kanban in a job shop environment. The International Journal of Production Research, 26(6): 1105–1118.
Huang C., and Kusiak A., 1996. Overview of kanban systems". International journal of computer integrated manufacturing: 9(3): 169-189.
Hiranabe K., 2008. Kanban applied to software development: from Agile to Lean. URL: http://www.infoq.com/articles/hiranabe-lean-agile-Kanban. Cited 2017/05/05.
Hurtado J., 2013. Open Kanban - an open source, ultra-light, Agile and Lean method. URI: http://www.agilelion.com/agile-kanban-cafe/open-kanban. Cited 2017/02/06.
Hirschheim R., Klein HK., and Lyytinen K., 1996. Exploring the intellectual structures of information systems development: A social action theoretic analysis. Accounting, Management and Information Technologies. 6: 1-64.
Radatz, J., Geraci, A. and Katki, F., 1990. IEEE standard glossary of software engineering terminology. IEEE., 1990. Standard Glossary of Software Engineering Terminology: 70.
Ikonen M., Pirinen E., Fagerholm F., Kettunen P. and Abrahamsson P., 2011. On the impact of kanban on software project work: An empirical case study investigation. 16th International Conference on Engineering of Complex Computer Systems: 305-314.
Kimura O., and Terada H., 1981. Design and analysis of pull system, a method of multi-stage production control. International Journal of Production Research: 19(3): 241-253.
Kitchenham BA., Budgen D., and Brereton OP., 2011. Using mapping studies as the basis for further research–a participant-observer case study. Information and Software Technology: 53(6): 638-651.
Kitchenham B., Pretorius R., Budgen D., Brereton OP., Turner M., Niazi M. and Linkman S., 2010. Systematic literature reviews in software engineering–a tertiary study. Information and Software Technology: 52(8): 792-805.
Kitchenham B., Charters S., 2007. Guidelines for performing systematic literature reviews in software engineering. Keele University, UK.
Kitchenham B., and Brereton P., 2013. A systematic review of systematic review process research in software engineering. Information and software technology. 55(12): 2049-2075.
Kim CS., Spahlinger DA., Kin JM., Coffey RJ., and Billi JE., 2009. Implementation of lean thinking: one health system's journey. The Joint Commission Journal on Quality and Patient Safety 35(8): 406–413.
Kniberg H., and Skarinm M., 2010. Kanban and Scrum-making the most of both. Enterprise software development series C4Media, Publisher of InfoQ.com.
Kumar CS., Panneerselvam R., 2007. Literature review of JIT-KANBAN system. International Journal of Advanced Manufacturing Technology 32(3-4): 393–408.
Laanti M., and Kangas M., 2015. Is Agile portfolio management following the principles of large-scale Agile? Case study in Finnish Broadcasting Company Yle. International conference on agile software development, Springer Berlin Heidelberg: 92–96.
Ladas C., 2009. Scrumban-essays on kanban systems for lean software development. Lulu. com.
Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045

Leffingwell D (2010) Agile software requirements: Lean requirements practices for teams, programs, and the enterprise. Boston, MA Addison-Wesley Professional.

Liker JK., 2004. The Toyota way: 14 management principles from the world's greatest manufacturer. New York McGraw-Hill Education.

Lyytinen K., and Rose GM., 2006. Information system development agility as organizational learning. European Journal of Information Systems. 15: 183-199.

Lyytinen K., and Rose GM., 2006. Information system development agility as organizational learning. European Journal of Information Systems 15: 183-199.

Mujtaba S., Feldt R., and Petersen K., 2010. Waste and lead time reduction in a software product customization process with value stream maps. Australia Conference Software Engineering: 139–148.

Maassen O., and Sonnevelt J., 2010. Kanban at an insurance company (are you sure?). International Conference on Agile Software Development. Berlin Heidelberg, Springer: 297–306

Melton T., 2005. The benefits of lean manufacturing: what lean thinking has to offer the process industries. Chemical Engineering Research and Design. 83: 662-673.

Nurdiani I., Börstler J., and Fricker SA., 2016. The impacts of agile and lean practices on project constraints: a tertiary study. Journal of Systems and Software: 119(C): 162–183.

Neely S., Stolt S., 2013. Continuous delivery? easy! just change everything (well, maybe it is not that easy). International conference on Agile software development: 121-128.

Nord RL., Ozkaya I., and Sangwan RS., 2012. Making architecture visible to improve flow management in lean software development. Software. IEEE 29: 33-39.

Naylor JB., Naim MM., and Berry D., 1999. Leagility: integrating the lean and agile manufacturing paradigms in the total supply chain. International Journal of production economics. 62: 107-118.

Ohno T., 1988. Toyota production system: beyond large-scale production. New York, CRC Press.

Paternoster N., Giardino C., Unterkalmsteiner M., Gorschek T., and Abrahamsson P., 2014. Software development in startup companies: A systematic mapping study. Information and Software Technology: 56(10): 1200-1218.

Olerup A., 1991. Design approaches: a comparative study of information system design and architectural design. The Computer Journal: 34(3): 215-224.

Orlikowski WJ., and Yates J., 2002. It's about time: temporal structuring in organizations. Organization science: 13(6): 684-700.

Petersen K., Vakkalanka S., and Kuzniarz L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology: 64: 1-18.

Petersen K., and Wohlin C., 2011. Measuring the flow in lean software development. Journal of Software Practice and Experience. 41(9): 975–996.

Petersen K., Feldt R., Mujtaba S., and Mattsson M., 2008. Systematic mapping studies in software engineering. International conference on Evaluation and Assessment in Software Engineering: 8: 68-77.

Petersen K., and Wohlin C., 2009. A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. Journal of systems and software. 82(9): 1479–1490.

Petersen K., Roos P., Nyström S., et al. 2014. Early identification of bottlenecks in very large scale system of systems software development. Journal of Software: Evolution and Process. 26: 1150-1171.

Poppendieck M., and Poppendieck T., 2007. Implementing Lean software development: from concept to cash. Boston MA, Addison-Wesley.

Poppendieck M., and Cusumano M., 2012. Lean software development: a tutorial. IEEE Software 29(5): 26–32.

Poppendieck M., and Poppendieck T., 2009. Leading Lean software development: results are not the point. Boston MA, Pearson Education.

Poppendieck M., and Poppendieck T., 2003. Lean software development: an agile toolkit: Addison-Wesley Professional.

Power K., and Conboy K., 2015. A metric-based approach to managing architecture-related impediments in product development flow: an industry case study from cisco. IEEE/ACM 2nd International Workshop on Software Architecture and Metrics: 15-21.

Reinertsen DG., 2009. The principles of product development flow: second generation Lean product development. Redondo Beach CA Celeritas Publishing.

Runeson P., and Höst M., 2009. Guidelines for conducting and reporting case study research in software engineering. Journal of Empirical software engineering: 14(2): 131.

Roper L., and Pettit, J., 2002. Development and the learning organisation: an introduction. Journal of development in practice. 12(3-4): 258-271.

Shinkle CM., 2009. Applying the dreyfus model of skill acquisition to the adoption of kanban systems at software engineering professionals. International conference on Agile software development. 186–191.

Shalloway A., 2011. Demystifying Kanban. Cutter IT Journal. URI: http://www.netobjectives.com/ files/ resources /articles/ Demystifying-Kanban.pdf. Cited 2017/05/08.

Shalloway A., 2010. The real differences between Kanban and Scrum. URI: http://www.netobjectives. com/ blogs/real-differences-between-kanban-and-scrum. Cited 2017/03/09.

Shaw M., 2003. Writing good software engineering research papers. 25th International Conference on Software Engineering: 726-736.

Shingo S., 1989. A study of the Toyota production system: from an industrial engineering viewpoint. New York, Productivity Press.

Sugimori Y., Kusunoki K., Cho F., and Uchikawa S., 1977. Toyota production system and kanban system materialization of just-in-time and respect-for-human system. International Journal of Production Research: 15(6): 553-564.

Taibi, D., Lenarduzzi, V., Janes, A., Liukkunen, K., and Ahmad, MO. (2017). Comparing requirements decomposition within the scrum, scrum with kanban, xp, and banana development processes. In International Conference on Agile Software Development, 68-83.

Terlecka K., 2012. Combining Kanban and Scrum - lessons from a team of sysadmins. International conference on agile software development: 99–102.

Tokatli N., 2008. Global sourcing: insights from the global clothing industry—the case of Zara, a fast fashion retailer. Oxford Journals: Journal of Economic Geography 8(1): 21– 38.

Towill D., and Christopher M., 2002. The supply chain strategy conundrum: to be lean or agile or to be lean and agile? International Journal of Logistics. 5: 299-309.

Unterkalmsteiner M., Gorschek T., Islam AM., Cheng CK., Permadi RB., Feldt R. , 2012. Evaluation and measurement of software process improvement—a systematic literature review. IEEE Transactions on Software Engineering: 38(2):398-424.

Vashist R., McKay J., and Marshall P., 2011. How well do we understand boundary practices? empirical evidence from a practice of business analysts. European Conference on Information Systems: 158.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.**
https://doi.org/10.1016/j.jss.2017.11.045

Versionone., 2017. The 10th annual state of agile survey. annual state of agile survey. URI: https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2. Cited 2017/05/08.

Versionone., 2016. The 10th annual state of agile survey. Annual State of Agile Survey. URI: http://stateofagile.versionone.com/ Cited 2017/06/05.

Venables M., 2005. Boeing: going for lean [lean manufacturing]. The Institution of Engineering and Technology, IEE Manufacturing Engineer. 84(4): 26–31.

Wang X., Conboy K., and Cawley O., 2012. Leagile software development: an experience report analysis of the application of Lean approaches in Agile software development. Journal of System and Software. 85(6): 1287–1299.

Wastell D., and Newman M., 1993. The behavioral dynamics of information system development: a stress perspective. Accounting, Management and Information Technologies. 3(2): 121-148.

Williams L., 2012. What Agile teams think of Agile principles. ACM Magazine of Communications: 55(4): 71–76.

Womack JP., Jones DT., and Roos D., 1990. The machine that changed the world: the story of lean production: how Japan's secret weapon in the global auto wars will revolutionize western industry. New York, Rawson Associates.

Wijewardena T., 2011. Do you dare to ask your HR manager to practice Kanban? The experience report of an offshore software company in Sri Lanka introducing Agile practices into its human resource department. International conference on agile software development: 161–167.

Zang JJ., 2011. A never ending battle for continuous improvement. International conference on agile software development, Springer Berlin Heidelberg: 282–289.

Cite: **Ahmad, M. O., Dennehy, D., Conboy, K., & Oivo, M. (2018). Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software*, *137*, 96-113.** https://doi.org/10.1016/j.jss.2017.11.045