

Distributed Kanban with Limited Geographical Distance: Analyzing Lean Principles Pull, Work in Progress and Kaizen

Raoul Vallon, Stefan Strobl, Martin Ras, Mario Bernhart and Thomas Grechenig
Research Group for Industrial Software, Vienna University of Technology, Vienna, Austria

Keywords: Software Development Methodologies, Distributed Software Development, Kanban, Lean, Software Process Improvement.

Abstract: Although the software development methodology Kanban, which refers and relates to the concepts and ideas of Lean Manufacturing originating in the Japanese automobile industry, was initially developed and used within distributed teams, correlating research is lacking, incomplete and relatively young as a field. This paper addresses the need for research in this field and investigates three specific aspects of Kanban in distributed teams: Pull System, Work In Progress Limit and the concept of Kaizen culture (continuous improvement) narrowed by the distribution, size and life cycle of the team. Our qualitative methodology is based on a case study where empirical data was collected through the use of semi-structured expert interviews. The evaluative strategy is qualitative content analysis. The results of this study show that challenges and complications result from the use of Kanban, but it is effective within distributed teams. The observed challenges are discussed in detail and we conclude with eight recommendations for practicing Kanban in a distributed team as well as indicators for future research directions.

1 INTRODUCTION

In 2001 Kent Beck et al. created the foundations of today's agile practices with the "Agile Manifesto" (Beck et al., 2001). Scrum is probably the most prominent representative of this group of approaches. However, in parallel Poppendieck and Poppendieck (Poppendieck and Poppendieck, 2003) started borrowing ideas from Lean Manufacturing practices and adapting them to the field of software engineering. Based on this work David J. Anderson pioneered the Kanban (Anderson, 2010) methodology in the mid 2000s.

In addition to the evolution of software development processes the typical consistency of a development team also changed fundamentally. Distributed teams, frequently over multiple locations, are already a common reality (Ebert, 2001). However, while there has been significant research on distributed Scrum with multiple case studies and experience reports (Jalali and Wohlin, 2010; Paasivaara and Lassenius, 2011; Kajko-Mattsson et al., 2010; Dorairaj and Noble, 2013; Smite et al., 2010) on both applicability and limitations, this is not the case for distributed Kanban. Vallon et al. have conducted a systematic literature review for the time range of 2010 to 2016 finding only

8 examples of Kanban in a distributed environment, contrasting with 53 representing Scrum (Vallon et al., 2018).

Based on the identified lack of research in the field, the main motivation of the work presented in this paper is exploratory research to find out if and how Kanban can be applied in a distributed organizational environment. As a first step towards that end, the objective of this paper is to analyze lean principles Pull, Work in Progress (WIP) and Kaizen (continuous improvement) using Distributed Kanban with teams working in a limited geographical distance.

2 BACKGROUND

The origins of Kanban can be traced back to the lean production practices implemented at Toyota in the 1940s and 1950s developed by Taiichi Ohno (Ohno, 1988). The basic idea behind the concept is to produce only exactly what is needed at the time when it is needed (Leopold and Kaltenecker, 2013). To achieve this, all forms of waste have to be eliminated.

Based on these foundations and ideas Poppendieck and Poppendieck (Poppendieck and Poppendieck, 2003) developed with their "agile toolkit"

a software development methodology, with the goal of eliminating waste and bureaucracy and establishing a continuous learning and improvement processes through short iterations, regular feedback and frequently delivered product increments. Over-boarding planning and written requirements on the other hand were to be limited (Poppendieck and Cusumano, 2012).

David J. Anderson can be seen as inventor of the modern Kanban software development process, which he successfully implemented at Microsoft (Anderson and Dumitriu, 2005). Anderson describes the process as an evolutionary system to enable change which uses a Kanban based pull system, visualization and further tools. Kanban therefore achieves a context-specific process optimization, minimizing friction and opposition to change by introducing change in small increments (Anderson, 2010).

3 RELATED WORK

Tanner and Dauane (Tanner and Dauane, 2017) performed a study with the focus on application of Kanban principles to alleviate the challenges and problems in communication and collaboration imposed by GSD. A main contribution was the highlighting of the Kanban board as a central tool for communication as well as creating transparency regarding progress, shifting priorities and changing requirements. In addition it was noted that rules for communication had to be formalized.

The three evaluated case studies and five experience reports have contributed the following main findings:

Kanban is seen as a metabolism for naturally fostering change and continuous improvement (Majchrzak and Stilger, 2017). However the introduction of Kanban in general needs the commitment of all involved stakeholders including management as well as an open environment (Prochazka et al., 2011). Teams need the necessary level of empowerment to function properly and to be able to solve challenges on its own. However this can also be a major motivational factor for individual team members (Govindaraj and Tadipatri, 2011). In general obstacles have to be addressed rapidly in order to avoid blockades (Paasivaara et al., 2014). Shorter time to market or release cycles can be the most tangible results of introducing Kanban (Viswanath, 2014).

An electronic Kanban board is a must in a distributed environment. It helps to make progress visible and positively reinforces participation (Tripathi et al., 2015). Centralized, electronic documentation

furthermore facilitates communication and prioritization of work items.

The WIP limit imposes a major challenge when introducing Kanban. It requires both compromise and continuous improvement to adjust to the specific local setting (Tripathi et al., 2015).

Finding a communication strategy that fits both the individuals and the corporate culture is a must (Bocock and Martin, 2011). In a distributed environment the distance needs to be minimized by utilizing technical aids like video-conferencing. However, whenever possible, face to face communication, also in an informal context, are invaluable assets and necessary to build trust.

Waste should be addressed both preemptively (e.g. pre-implementation call) or retro-actively (e.g. Code Reviews). The latter is also a key technique for knowledge transfer. Highly prioritizing code review tasks has the benefit of keeping feedback cycles short therefore reducing the need for context switches (Moe et al., 2015; Bocock and Martin, 2011). Pair programming and actively encouraging team members to take on tasks from other areas can be seen as an investment in a more even knowledge distribution.

4 CASE STUDY

Based on the previous sections, we formulate our research objective, case study design and provide background on the units of analysis.

4.1 Research Objective

As shown in Section 3 available research into Kanban in a distributed setting is limited to one study, three case studies and five experience reports. To achieve the necessary depth, the scope of the research has to be limited in terms of selecting cases and principles of Kanban.

The selection criteria for Kanban teams starts with their location and distribution. Only teams with a distance classification of Onshore-Insourcing-Close-Similar, Onshore-Insourcing-Distant-Similar and Offshore-Insourcing-Near-Small (Šmite et al., 2014) are selected to minimize the effect of travel times, time differences and cultural distance. At least 25% of the team members have to be distributed. The size of a team has to be between 7 and 15 members to eliminate anomalies by sub-optimal team sizes (Rodríguez et al., 2012).

Concerning the Kanban software development process in place the following criteria are applied: A Kanban board has to be present and accessible from

all locations. A pull principle has to be in place. At least one of the process steps needs to have a WIP limit in place. While elements of other agile methodologies might be present, the key characteristics of Kanban need to be preserved. Neither the presence of a Kaizen culture nor the usage of the term Kaizen itself can be strictly enforced. However regular measures for process improvements must take place.

Kanban is defined by a set of six fundamental principles (shown here in abbreviated form) from which the following aspects can be derived (Anderson, 2010; Hammarberg and Sunden, 2014).

1. Visualization
2. Work In Progress
3. Manage Flow
4. Make Policies Explicit
5. Implement Feedback Loops
6. Improve collaboratively, evolve experimentally, using models and the scientific method

The individual aspects are evaluated against the following criteria to decide if they should be in the scope of this research:

1. Distributed teams are confronted and have to solve challenges imposed by distance that are not present in collocated environments.
2. The aspect has to have a direct effect on the work and behaviour of the individual team members.

The aspect Kanban board can be excluded because of the existence of prior research (Tanner and Dauane, 2017). The aspects performance indicators, evaluation of improvement and customer feedback only affect the team indirectly or in a differed manner and can therefore be excluded as well. Having explicit rules rather supports than imposes challenges on distributed teams. This results in the selection of pull principle, WIP limits and continuous improvement (Kaizen culture) as the aspects for in-depth evaluation.

We defined a research question to cover each aspect as well as a set of propositions for further refinement. The propositions have been designed with the possibility of falsification in mind under the following general hypothesis: The Kanban method can be applied for distributed teams in a setting with limited geographic distance without negative effect.

RQ 1: What are the challenges and possible mitigations of applying the **pull principle** in distributed teams?

RP 1.1: In distributed teams the pull principle requires communication when taking over a task from the previous steps.

RP 1.2: The communication over distance does not have a negative effect on the application of the pull principle.

RQ 2: What are the challenges and possible mitigation of defining **WIP limits** in distributed teams?

RP 2.1: For setting a WIP limit distributed teams have to find their own equation.

RP 2.2: The application of a WIP limit on individual process steps depends on the task at hand, the process design and the team composition.

RP 2.3: A change in the composition of the distributed team affects the WIP limits

RQ 3: What are the challenges and possible mitigation of establishing a **Kaizen culture** in distributed teams?

RP 3.1: The establishment of a Kaizen culture does not follow a predefined rule-set.

RP 3.2: Trust between team members, superiors and in the organization itself are fundamental requirements for the establishment of a Kaizen culture.

RP 3.3: Open and direction communication are necessary preconditions for a Kaizen culture.

RP 3.4: The distance in communication does not have a negative effect on a Kaizen culture.

4.2 Case Study Design

The case study in this paper was designed after the guidelines and recommendations by Runeson et al (Runeson et al., 2012). It has both a descriptive as well as exploratory research purpose. We are looking at a total of four cases which have been selected according to the criteria as laid out in Section 4.1.

The goal of this case study is to obtain insights on how the previously mentioned three aspects of Kanban are applied in a distributed setting. In addition perceived challenges and proposed solutions are to be analyzed. Distilled from these findings a set of recommendations for future applications of Kanban in similar settings are to be elaborated.

The case study has been designed as an embedded, single case study with the context of applying selected aspects of Kanban in distributed software development teams. The units of analysis are the individual, distributed software development teams. The case study design and applied research in this paper allows us to falsify the research propositions by finding counter examples. A proof of general applicability of the propositions on the other hand is not possible with the limited data set available.

The data collection is primarily based on the execution of semi-structured interviews representing first degree contact as defined by Lethbridge et al. (Lethbridge et al., 2005). The semi-structured nature of the

interviews was chosen to account for the exploratory part of the research. In addition the Kanban board and here especially the general structure of the board is used as a complementary data source. The interview guideline was developed based on the research questions and propositions. It begins by inquiring about team composition and distribution and how Kanban has been implemented at this organization. In the main part it focuses on the selected aspects of Kanban and goes into detail there. Interviews were conducted in German language, therefore the interview guideline as well as the transcripts were also written in German.

The units of analysis were contacted through one of two means. Direct contact was established through exhibitions, conferences and user groups. Other organizations were reached through contacting them online. Pre-selection here was done by finding companies with job postings for software developers for different locations. A total of 97 organizations were contacted, out of which four were part of the final selection. The overall response rate was 13%.

The data analysis was done applying a structuring technique described by Mayring (Mayring, 2015) to inductively create categories. However the process was adjusted to the individual needs of this study. The transcribed interviews were then coded with a set of a priori defined keywords. Additional key words were added during the analysis as more insights were gained. The coding was therefore done iteratively. Due to the relatively small set of raw data the coding was done entirely manually. The main categories were Kanban process design, continuous improvement, pull principle, WIP limits and communication and trust. Among all five of them there were a total of 36 key words. Figure 1 illustrates the coding process described by example of the Kanban Process Design.

Table 1 gives an overview of the units of analysis.

4.3 Alpha

Alpha specializes in business process digitalization and provides software products and services to achieve their digital transformation goals. Software development is therefore primarily project driven. The composition of the teams also varies depending on the individual projects needs. The main work item is called story and is handled in a 3 level stage-gate-process. The levels are conception, development and acceptance. The current progress is visible through an electronic Kanban board. A short daily standup is conducted each day utilizing a video conferencing solution.

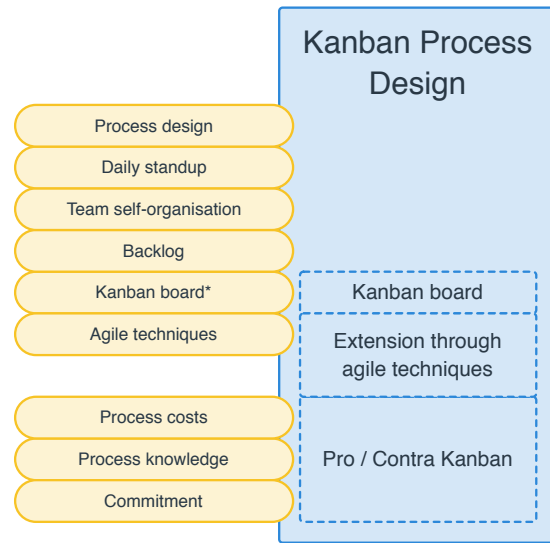


Figure 1: Code words and categories for Kanban process design.

4.4 Beta

Beta is an agency specialized in e-commerce. Through a merger the agency now has five locations across two countries. Teams size and composition is geared towards the current project and therefore varies between five and 11 members. Again a three level process is used with development preceded by an analysis phase and concluded by a delivery stage. Daily stand ups are done using video conferencing, retrospectives on an "as needed" basis. The Kanban board exists in electronic form.

4.5 Gamma

Gamma is a highly specialized system provider combining hardware and software development to provide optimal, integrated solutions to both government and industrial customers. Out of a total of 19 locations across three countries, 2 are mainly focused on software development. Teams are either focused on product development or customization for individual customers, which affects both team composition and life span. The interview was done with a group of two individuals (P1, P2), one heading product development, the other a team lead for two development teams with a team size of 10 to 12. A Zero-Bug-Policy with a dedicated swim lane ensures bugs are prioritized over feature development. The board only exists in electronic form.

Table 1: Overview of the units of analysis.

Unit of Analysis	Role Interviewee	Team Size	Locations (Countries)	Distribution
Alpha	Agile Coach, Scrum Master	4 to 9	2 (2)	variable
Beta	CEO, Business Unit Manager	5 to 11	5 (2)	variable
Gamma	P1: Divisional Head; P2: Team Lead	10 to 12	2 (1)	30% : 70%
Delta	Agile Coach, Scrum Master	6 to 11	2 (1)	variable

4.6 Delta

Delta provides individualized solutions in the fields of e-commerce and digital marketing. There are an average of three times distributed over two locations in one country with a varying team size of six to 11. A preparation phase with detailed analysis of the work items precedes development. Team members individually subdivide work items into tasks in the electronic Kanban board. This allows team members with specialized skills to work in their area of expertise. Both daily stand ups and retrospectives are held.

5 RESULTS

Based on the responses from our units of analysis we will now try to falsify our research propositions and answer the research questions. Subsequently we discuss the threats to validity.

5.1 Research Propositions

RP 1.1: All interviewees confirmed the need for thorough and detailed documentation, specification and requirements analysis to make the pull principle work in practice. This avoids additional round trips in direct communication as well as down times through blockades. This reduces the overall communication overhead and avoids misunderstandings. However the additional effort and cost for preparing this information should not be disregarded. Structuring work items into fine grained tickets in the electronic issue tracking system helps create transparency in terms of progress and problem areas. It also facilitates communication through comments and enables traceability in the form of linking the discussion to a specific issue. Other observed methods of communication were instant messaging, phone calls and video calls. This proposition was not falsified by our observations.

RP 1.2: The pull principle requires communication when taking over a task from the previous step. Communication is facilitated through instant messaging and video telephony as well as screen sharing sys-

tems. The electronic Kanban board enables traceability and makes transparent who was responsible for which step. The negative effect of distance on communication has been proven by the empirical data. A good flow of communication can only mitigate these effects. This proposition has therefore been falsified.

RP 2.1: All units of analysis set their WIP limit through mathematical equations. Alpha and Beta only take the team size as a variable into account. Gamma also tries to reflect pair programming and Delta gives its teams more liberty by not imposing a limit on sub tickets. With the collected data the proposition was not falsified.

RP 2.2: All of the interviewed organizations limit the amount of tasks that are actively being developed. Only two units (Gamma and Delta) also limit subsequent process steps in quality assurance to ensure issues identified by quality assurance are rectified in a timely manner. One subject also indicates that WIP limit are dynamically adjusted if a specific work item requires such a step. The proposition can therefore not be falsified.

RP 2.3: Long term changes in the teams composition are reflected in the WIP limit in all cases. Short team absences like sickness or holidays and additions (e.g. reflecting a specialized need) do not trigger a change in the WIP limit. Therefore the proposition can not be falsified.

RP 3.1: All units of analysis conduct regular retrospectives as well as daily, short exchanges. Alpha and Delta see flat hierarchies as the main enabler for a Kaizen culture. Gamma sees benefits in an open and informal atmosphere and company culture. Beta has a slightly different view with two dedicated roles agile coach and development lead at each location responsible for detecting and mitigating process related respectively technical issues. Apart from meeting on a regular basis no rule-set was elicited, therefore this proposition was not falsified.

RP 3.2: All units of analysis have measures in place to nurture trust. New hires are supported by a multi-week on-boarding process (Alpha, Gamma, Delta). Delta has designated, local points of contact for every employee to facilitate personal communication. All cases furthermore enable face to face

meetings of all contacts. This can be in the form of workshops (Alpha, Gamma) at one location or company events (Beta, Delta). Apart from building trust these measures are designed to facilitate communication and sometimes also to incentivize. This proposition was not falsified.

RP 3.3: Direct personal communication is stated as a key enabler for trust and building long term relationships. However since meetings in person are not always possible and do incur significant cost in a distributed environment, additional events are held to mitigate. Beta and Delta do hold regular events presenting news and innovations to their employees. Beta, in addition, enables all employees to participate in events with a focus on technical exchange and sharing of experiences. The data shows that all participants recognize the need to support direct communication. Therefore the proposition cannot be falsified.

RP 3.4: All units of analysis see the need to take additional measures to facilitate direct communication to mitigate a perceived negative effect due to the distributed nature of their team setups. Digital direct communication like instant messaging and video or audio conferencing are not seen as fully equivalent. This proposition can therefore be regarded as falsified.

5.2 Research Questions

RQ 1: The main challenge in applying the pull principle is to provide sufficient amounts of information and detail about each task for the team to be able to do its work. All units of analysis have therefore established a preceding specification and analysis phase with a strong focus on written artifacts which concludes with a joint meeting to present the planned feature to all participants. The flow of information during handover between individual steps is another challenge. Apart from the documentation produced by the initial phase a key mitigation strategy is the consistent use of the electronic Kanban board and the underlying ticketing system to provide a history for each work item. We observed that some units saw beneficial value in letting the team break down a work item into individual tasks by themselves. Apart from a stronger personal ownership of the resulting tasks the structuring was optimized towards the teams distributed composition to minimize communication over distance. However, although the amount of necessary communication can be reduced by above-mentioned measures, direct communication is a key ingredient for successfully applying the pull principle. Eliminating it might rapidly lead to the creation of local silos with all associated issues.

RQ 2: Three main challenges were identified. First the team needs to decide which steps of the process actually need limits to ensure a continuous flow of work items and this avoid bottlenecks. Second the composition of the teams concerning the individual skills is a key factor with respect to the first challenge. If a skill is rare, a higher limit might be needed to ensure flow. If on the other hand a task can be done by the majority of team members (e.g. reviews) a low limit can ensure timely execution of certain tasks. Third a correlation with established work patterns (e.g. pair programming) needs to be reflected when setting the WIP limit. On a related matter the granularity of the work items also has a direct effect on the limit. Lower granularity requires lower limits, higher granularity might warrant more generous limits. As all these challenges are interdependent, a formula and fine-tuning through experimentation is necessary to achieve good results in each individual case.

RQ 3: Open and direct communication and trust can be seen as preconditions for establishing a Kaizen culture. Distance impedes both and therefore countermeasures are necessary. These can range from company events, over technical exchanges to technologically supported communication and additional possibilities for employee training. Retrospectives and daily standups are established tools in distributed software development to foster a Kaizen culture. These tools establish a setting where perceived problems can openly be discussed. However it is important to also facilitate direct follow up communication to make sure identified issues are being understood and addressed properly by all sides.

5.3 Threats to Validity

To ensure internal validity different approaches for analysis have been applied as recommended by Tanner and Dauane (Tanner and Dauane, 2017) by employing a cross-case result analysis, grouping and comparing the data from multiple sources. The external validity is limited by the constraints employed while selecting the cases. Therefore the presented results cannot be fully generalized to all distributed software development settings. In addition using job postings to select the units of analysis might introduce a bias towards settings with a high degree of employee fluctuation. Furthermore three out of four units of analysis are working in the field of contract work resulting in relatively short project duration and therefore frequent team transformations. Cultural differences have been mostly eliminated by the choice of limitations in distance. Finally although recom-

mended otherwise (Mayring, 2015), the interviews and subsequent coding had to be conducted by a single person due to limitations of the research team setup at the time of execution.

6 RECOMMENDATIONS

The answers to our research questions in the previous section have provided possible solutions to a series of challenges. From these results we can derive the following recommendations:

(1) Carefully analyze and thoroughly document upcoming work items. This will ensure proper flow through the process stages by pull principle and avoid additional, late communication overhead. Comprehensive documentation is a welcomed side product.

(2) Allow self organizing teams. If the team has the ability to break down work items into tasks by itself and decide their distribution, the amount of transitions can be reduced. This requires well qualified and empowered team members.

(3) Apply WIP limits cautiously. Only apply limits where needed, primarily factoring in team size, but also observing established practices and patterns of work distribution.

(4) Utilize technology to disseminate information. Communication in a distributed environment is hard. Technology can help overcome some of these obstacles. A mix of an electronic Kanban board and synchronous as well as asynchronous communication technology is highly recommended.

(5) Enable the team to communicate outside of the process context. Targeted measures to facilitate communication outside of the day to day development context in the form of informal or special topic events will lower the barrier for initiating communication and help foster trust.

(6) Enable continuing education as part of continuous improvement. Employees will contribute more when they get the time and opportunity to critically reflect on established practices by obtaining outside views. Done in groups educational measures can also be seen as events outside of the development context.

(7) Apply retrospectives modestly and purposefully. Depending on the maturity of the team and their individual drive to continuous improvement retrospectives might be needed regularly or just occasionally.

(8) Regard the corporate culture as the counterpart to continuous improvement. We observed an open culture and flat hierarchies as the primary enablers for continuous improvement. Both should encourage the

team members to speak up and address problems as they arise.

7 CONCLUSION

Kanban as proposed by Anderson has roots in a distributed environment. However established literature hardly takes note of distributed settings even though the current global setting increasingly demands globally distributed software development setups. Distance on a cultural, temporal as well as geographical level impose additional challenges in these scenarios. The main research goal of this work is the investigation of applied Kanban in a distributed environment limited to the aspects of pull principle, Work in Progress limits and Kaizen culture. Based on extensive research of the current state-of-the-art in distributed Kanban, a case study was designed and executed with a total of four units of analysis. The main source of empirical data were a series of semi-structured interviews.

All three selected aspects need good communication and established trust as a basis. Most challenges as well as the resulting recommendations revolve around this topic. Mitigating action like additional team events, a sophisticated on-boarding or additional communication technology are necessary to counteract the negative effects of distance. Empowering the teams and individual members is another crucial aspect to successfully applying Kanban in a distributed environment. A Kaizen culture of continuous improvement, although not commonly used as a term, is consistently present and understood as one of the main motivations and drivers behind applying a lean development approach.

Overall the case study has shown that Kanban and specifically the selected aspects of pull principle, WIP limits and Kaizen culture are being successfully applied in a distributed environment. However the challenges around communication, omnipresent in software development projects, are amplified by team distribution and therefore need additional attention to mitigate negative effects.

One direction of future research is to eliminate the self-imposed limitations introduced in this work to ensure wider applicability. Additional cases and quantitative studies are required to further validate and refine the results of this case study. Furthermore two cases have indicated observations that Kanban projects are perceived to be more cost-effective than projects applying Scrum. Research is needed to find out if there is actual correlation or if this is a side effect of the specific projects selected for a Kanban de-

velopment methodology.

REFERENCES

- Anderson, D. J. (2010). *Kanban: successful evolutionary change for your technology business*. Blue Hole Press, e-book edition.
- Anderson, D. J. and Dumitriu, D. (2005). From Worst to Best in 9 Months: Implementing a Drum-Buffer-Rope Solution in Microsoft's IT Department. Barcelona.
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., and Fowler, M. (2001). Manifesto for Agile Software Development.
- Bocock, L. and Martin, A. (2011). There's Something about Lean: A Case Study. In *2011 Agile Conference*, pages 10–19.
- Dorairaj, S. and Noble, J. (2013). Agile Software Development with Distributed Teams: Agility, Distribution and Trust. Nashville, Tennessee, USA. IEEE.
- Ebert, C. (2001). *Global Software and IT: A Guide to Distributed Development, Projects, and Outsourcing*. Wiley.
- Govindaraj, S. and Tadipatri, S. (2011). Use of Kanban in Distributed Offshore Environments. *Cutter IT Journal*, 24:29–33.
- Hammarberg, M. and Sunden, J. (2014). *Kanban in Action*. Manning.
- Jalali, S. and Wohlin, C. (2010). Agile Practices in Global Software Engineering - A Systematic Map. Princeton, New Jersey, USA. IEEE.
- Kajko-Mattsson, M., Azizyan, G., and Magarian, M. K. (2010). Classes of Distributed Agile Development Problems. Orlando, Florida. IEEE.
- Leopold, K. and Kaltenecker, S. (2013). *Kanban in der IT - eine Kultur der kontinuierlichen Verbesserung schaffen*. Hanser München, 2. Aufl. edition.
- Lethbridge, T. C., Sim, S. E., and Singer, J. (2005). Studying Software Engineers: Data Collection Techniques for Software Field Studies. *Empirical Software Engineering*, 10(3):311–341.
- Majchrzak, M. and Stilger, L. (2017). Experience Report: Introducing Kanban Into Automotive Software Project. *e-Infomatica Software Engineering Journal*, 11(1):41–57.
- Mayring, P. (2015). *Qualitative Inhaltsanalyse: Grundlagen und Techniken*. Beltz, Weinheim, 12 edition.
- Šmite, D., Wohlin, C., Galviņa, Z., and Prikładnicki, R. (2014). An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19(1):105–153.
- Moe, N. B., Cruzes, D. S., Dybå, T., and Engebretsen, E. (2015). Coaching a Global Agile Virtual Team. In *2015 IEEE 10th International Conference on Global Software Engineering*, pages 33–37.
- Ohno, T. (1988). *Toyota Production System: beyond large-scale production*. Cambridge.
- Paasivaara, M., Behm, B., Lassenius, C., and Hallikainen, M. (2014). Towards Rapid Releases in Large-Scale XaaS Development at Ericsson: A Case Study. In *2014 IEEE 9th International Conference on Global Software Engineering*, pages 16–25.
- Paasivaara, M. and Lassenius, C. (2011). Scaling Scrum in a Large Distributed Project. Banff, Alberta, Canada. IEEE.
- Poppendieck, M. and Cusumano, M. A. (2012). Lean Software Development: A Tutorial. *IEEE Software*, 29(5):26–32.
- Poppendieck, M. and Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison Wesley.
- Prochazka, J., Kokott, M., Chmelar, M., and Krchnak, J. (2011). Keeping the Spin – From Idea to Cash in 6 Weeks: Success Story of Agile/Lean Transformation. In *2011 IEEE Sixth International Conference on Global Software Engineering*, pages 124–130.
- Rodríguez, D., Sicilia, M. A., García, E., and Harrison, R. (2012). Empirical findings on team size and productivity in software development. *Journal of Systems and Software*, 85(3):562 – 570.
- Runeson, P., Host, M., Rainer, A., and Regnell, B. (2012). *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley.
- Smite, D., Moe, N. B., and Ågerfalk, P. J. (2010). *Agility Across Time and Space*. Springer-Verlag Berlin Heidelberg.
- Tanner, M. and Dauane, M. (2017). The use of Kanban to alleviate collaboration and communication challenges of global software development. *Issues in Informing Science and Information Technology Education*, 14:177–197.
- Tripathi, N., Rodríguez, P., Ovais Ahmad, M., and Oivo, M. (2015). Scaling Kanban for Software Development in a Multisite Organization: Challenges and Potential Solutions. In *Agile Processes in Software Engineering and Extreme Programming*, pages 178–190. Springer International Publishing.
- Vallon, R., Estácio, B. J. d. S., Prikładnicki, R., and Grechenig, T. (2018). Systematic literature review on agile practices in global software development. *Information and Software Technology*, 96:161–180.
- Viswanath, U. (2014). Lean Transformation: How Lean Helped to Achieve Quality, Cost and Schedule: Case Study in a Multi Location Product Development Team. In *2014 IEEE 9th International Conference on Global Software Engineering*, pages 95–99.