

## Towards continuous value delivery with review meetings in agile methodologies: A structured review



Khalid Khan <sup>1,\*</sup>, Anam Siddiqui <sup>2</sup>, Usman Waheed <sup>3</sup>, Fadzil Hassan <sup>4</sup>, Anh Nguyen Duc <sup>5</sup>

<sup>1</sup>Department of Computer Science, PAF Karachi Institute of Economics and Technology, Karachi, Pakistan

<sup>2</sup>Department of Computer Science, Sir Syed University of Engineering and Technology, Karachi, Pakistan

<sup>3</sup>Department of Computer Engineering, Usman Institute of Technology, Karachi, Pakistan

<sup>4</sup>Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia

<sup>5</sup>Department of Business and IT, University of Southeastern Norway, Notodden, Norway

### ARTICLE INFO

#### Article history:

Received 13 March 2019

Received in revised form

20 July 2019

Accepted 20 July 2019

#### Keywords:

Agile

Continuous value

Scrum

Continuous software engineering

### ABSTRACT

Continuous value delivery (CVD) is the practice to ensure the delivery of user and business value via a fast, reliable, and repeatable process. The change management lies at the core of CVD as the adjustments in value propositions often take place. Review meetings play an important role in change management because the review is a formal assessment with the intention of instituting change, if necessary. A thorough discussion of different agile methods in terms of review meetings as needed. Different papers were studied to extract the data regarding these meetings. It is observed that the sprint review meeting is important to the development team as it is an opportunity for the team to show its work explicitly and get appreciated. Development team spirits are increased by such environments. The sprint review establishes a firm level of sociable comparative competition between scrum teams that keeps everyone focused. It is actually equivalent to a user acceptance test. This paper discusses the importance of review meetings and their impact on CVD in well-known agile methods and using a structured review methodology. It evaluates the processes of review meetings such as planning, adaptability, integration in increment and reevaluation in well-known agile methods in order to find their effectiveness towards achieving CVD. It is found that Scrumban when strengthened with the EVO's increment inspection process, can help in achieving CVD.

© 2019 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

### 1. Introduction

Organizations are investing in finding ways to faster delivery of “valued” software to the customers to cater the competition in software industry (Phillips et al., 2015). The term “value-based software engineering” was coined in Boehm (2003) that questioned the absence of “value” in traditional practices and their concerns were costs, resource allocation, scheduling etc. Some of the practices that can make it happen are Continuous Integration (CI), Continuous Delivery (CDE), and Continuous Deployment (CD) (Shahin et al., 2017). CI presents the procedure of integrating work-in-progress multiple times in a day, whereas CDE and CD shares

the process of automating quick and reliable release of software. In CI, the integration and merger of development work can take place frequently such as multiple times per day which enables software companies to have shorter and frequent release cycle and improve software quality (Fitzgerald and Stol, 2017) whereas in CDE, the aim is to ensure that the application is always in tested and production-ready state (Larman and Vodde, 2016) that reduces deployment risk and faster user feedback. CD practices aim to automate the steady deployments for every change in production or customer environments (Ladas, 2009), as soon as developers commit a change, the change is deployed to production through a deployment pipeline via a push based approach which allows frequent and automated releases. The logical extension to CI and CD, is Continuous Value Delivery (CVD) which can be defined as a practice that ensures the delivery of user and business value via a fast, reliable, and repeatable process. The focus is on Value Realization; hence, all the efforts revolves around

\* Corresponding Author.

Email Address: [khalid.khan@pafkiet.edu.pk](mailto:khalid.khan@pafkiet.edu.pk) (K. Khan)

<https://doi.org/10.21833/ijaas.2019.10.005>

Corresponding author's ORCID profile:

<https://orcid.org/0000-0001-6580-1624>

2313-626X/© 2019 The Authors. Published by IASE.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

generating value proposition at each and every step from planning to results. CVD is a merger of value planning, adoption and measurements. The first step to value planning is to identify the key stakeholders in the business, their perceived business value and its measurements. Adoption is the next step, new released features if remain unused are of no value. Users need to use the new features and actually change their behaviours to realize the value which is not easy. This is change management issue and effective adoption planning; user readiness strategies are required. For the measurement, it not about measuring the usage of certain feature, it's about connecting the dots that can lead to business impact. Measuring the changing behaviour to quantify certain benefits that can be associated with some financial amount shows real improvements. Value reporting is the feedback loop that can be used to adjust value planning efforts.

Agile is a software development approach in which requirements and solutions evolve through the collaborative effort of small cross-functional teams and their customers (end users). These team are self-organized to the major extent and perform adaptive planning and development to achieve early delivery (Conboy, 2009; Matkovic et al., 2018). The focus is on rapid and flexible response to change to deliver continuous value. The term value is difficult to define as different environments interpret value differently depending on what gives business value to them. The general use of the word value ranges from usefulness monetary worth which means that the value of software is assigned by stakeholders outside of the development team, hence become customer oriented. Many of the recent improvement trends that have influenced software development practice have a focus on user and business value. The agile manifesto focuses on customer collaboration and working software to provide customer satisfaction through early delivery (Conboy, 2009). The recent trend of lean start-ups (Ries, 2011) also states that the learning about customer perceived value is a must. AS stated earlier, agile development methods are now focused on value but predicting the value of software isn't a simple task. One way of doing it, is to estimate the value in the form of "benefit points". This is about estimating the value in terms of user stories as done for cost estimation in agile development. Another way to understand the value is through empirical means which is based on the idea of continuous experimentation. In this approach, customers are facilitated with potentially valuable features and value of delivered functionality is determined. This approach can be applied to multiple customers who can be handed over different set of features to determine functionality. In this way, customer perceived value of the various feature of the product can be estimated in quick time (Fagerholm et al., 2014).

A review meeting can be defined as a process involving project personnel, managers, users, customers, or other stakeholders to testify and examine the process/project and decide planning for

upcoming changes in software process model. It is conducted to get the approval of stakeholders on current iteration/project and to get the feedback (Jayatilleke et al., 2018). Reviews can be of different type like architecture review, design review and code review. For all types of reviews, meetings are conducted among "Whole Team" with the aim to improve quality by taking feedbacks resulting satisfaction of customers and the team. All agile methods have their own strategy regarding the review meetings and each methods employs review meetings in a unique manner. Here we are giving a general categorization and high level description of review meetings required to gain the true essence of agile values that consist of speed of delivery and time to market as the key metrics.

- **Inform:** A meeting to inform the team on the progress of something towards a milestone.
- **Plan:** A meeting to contribute to the schedule of some concept and evaluate related risks.
- **Refinement:** A meeting that provide clarity about the scope, cost or time of some concept.
- **Retrospective:** A meeting to improve on some concept that is revisited due to some issue.
- **Investigative:** A meeting to get the reason of a drift in the outcome of a planned process.

CVD is the order of the day and these days it is not possible for software firms to attain competitive advantage without CVD. Agile methods are also paving the path towards CVD. Review meetings play a very important role in change management and are critical to move towards CVD because there are frequent adjustments in value propositions in CVD. It is safe to say that CVD can be ensured with the agile methods that can handle the review meetings in a robust manner. The paper presents a structured review of some well-known agile methods to understand their review meeting processes and in an attempt to correlate review meetings with CVD. In section 2 various well-known agile methods are discussed with respect to review meetings. Section 3 presents the research methodology which includes development of research questions. Section 4 and 5 discusses the research findings and conclusion respectively.

## 2. Review meetings in agile methods

Some organizations strongly emphasize on reviews and feedbacks and consider it as asset for their team and customer. Reviews conducted late in lifecycle are often misguided. While considering reviews and meetings throughout the development lifecycle, quality and satisfaction can be best maintained. Review Meetings are very helpful for inspecting, maintaining and reevaluating project or different modules/increments of projects (Brereton et al., 2007). Agile methodology is best known for satisfaction about change management and requirement analysis. Some well-known agile

methods are discussed below in the context of review meetings.

Scrum is a framework that is iterative and incremental for projects /product /application development. It works in cycles of development called Sprints (Sutherland and Schwaber, 2010). A sprint review meeting takes place at the end of the sprint to examine the increment and adapt the product backlog if needed. During this meeting, the scrum team and stakeholders collaborate about what was done in the sprint. This is an informal meeting, not a status meeting, and the presentation of the increment is intended to elicit feedback and foster collaboration. Normally, a four-hour time-boxed meeting is done for one-month sprints. For shorter sprints, the event is usually shorter. The scrum master schedules the meeting and product owner discusses the current state of product backlog. The whole team discuss next things to do and as per market what is the most valuable thing to do. At the end, the timeline, budget, potential capabilities, and marketplace for the next expected release of the product is discussed.

Scrumban is a hybrid of Scrum and Kanban. It is a management framework that emerges when teams employ scrum as their chosen way of working and use the kanban method as a lens through which to view, understand and continuously improve how they work (Ladas, 2009). Scrumban hold daily meetings but there are no Sprint or release planning meetings and retrospectives in Scrumban. Scrumban embraces on-demand planning. Scrum teams must estimate the time the assigned work takes to meet the commitments of a sprint as scrumban does not have a time constraint. Instead, estimating becomes apparent over time as the team accomplishes more tasks (Reddy, 2015).

Nexus is a framework for developing and sustaining scaled software which is built upon

scrum. The Nexus sprint review is held at the end of the sprint to provide feedback on the integrated increment that the Nexus has built over the sprint and to adapt the product backlog if needed (Bittner et al., 2017). The result is a revised product backlog nexus sprint retrospective which is a formal opportunity to inspect and adapt itself and create a plan for improvements to be enacted during the next sprint to ensure continuous improvement.

Evolutionary Value Delivery (EVO) suggests 20 steps of value delivery with tangible value being provided to stakeholders at every step (Johansen and Gilb, 2005). In EVO, for each iteration there is a re-evaluation of solutions which give way the highest value to cost ratio, guided by feedback and estimates. It needs active stakeholder participation to steer the project. Each iteration is client-driven and is based of adaptive planning.

LeSS is a lightweight framework for scaling Scrum to more than one team. A Diverge- Converge meeting pattern is used in LeSS for review. A bazaar is used in diverge period. In a large room which has multiple areas, items are shown and discussed by team representatives. Stakeholders visit areas of interest. During the converge period, stakeholders sum up their opinions from the bazaar (Larman and Vodde, 2016). A component of items may be inspected on a common computer projector.

These well-known agile methods differ with each other in many ways but the important factors are roles and iterations. User roles are assigned to different stakeholders who are directly linked with the project and have some responsibilities whereas an iteration, is a time frame during which development specific activities takes place. By considering the above mentioned factors, here we present a comparison of various well-known agile methods in Table 1.

**Table 1:** Comparison of roles and iterations in well-known agile methods

Agile Methods	Roles	Iterations
Scrum	Product Owner, Scrum Master, Team	Duration: Two weeks. Type: Release Iteration
Scrumban	Team, as per need roles	Duration: One to Four weeks. Type: Development, Release, Maintenance, Bugs
Nexus	Product Owner, Scrum Master, Nexus Integration Team Members	Duration: Based on the dependencies inherent in the Product Catalog. Type: Release, Maintenance, Bug
EVO	Project Manager, Architect, Stakeholder, all other development team members	Duration: One to Three weeks. Type: Not Defined.
LeSS	Scrum Master, Product Owner, Team	Duration: Two hours to two days. Type: Development, Release, Maintenance, Bugs

### 3. Research methodology

The objective of this structured review is to identify the role of review meetings in establishing an association of various well-known agile methods with CVD. To examine different well-known agile methods, past researches, case studies, practices have been investigated so that accurate result can be established. After identifying the requirements, keywords were searched for literature review. The inclusion/exclusion criteria were established and research questions were developed. Answers were

found out from the literature review and finding were presented at the end.

#### 3.1. Search keyword strategy

Search terms were formed by using the Boolean expression 'OR' and combining major look out terms using 'AND.' Data is collected by using general terms. Agile AND Review Meeting AND Continuous Value Delivery AND (Scrum OR Scrumban OR Nexus OR Evo OR LeSS) AND (Planning OR Retrospective OR Increment OR Feedbacks). The parameters for agile methodologies in review meetings were extracted

from textbooks and different research papers. Google scholars is the primary source for research which extracted data from various databases including Scopus, IEEE Explore, Wiley online library, ICSR, Science digest, Springer Link, World Scientific and Digital library.

### 3.2. Inclusion and exclusion

After searching various research papers, articles and text books inclusion and exclusion needed to be applied so that only relevant studies would be searched for review. Various research papers were studied, and their count was estimated to be 22.

#### 3.2.1. Inclusion criteria

- Empirical studies using the agile methodologies.
- Empirical study comparing the agile and conventional models

- Empirical study having agile methodologies and Review meetings.
- Empirical study using planning, retrospective, increments, meetings in agile methodologies

#### 3.2.2. Exclusion criteria

- Studies without empirical results of agile methodologies.
- Review studies without empirical result.
- White papers
- Unrecognized conference papers.

### 3.3. Research questions

Five Research questions were developed to understand the role, process and impact of the review meetings. These research questions were shown in [Table 2](#).

**Table 2:** Research questions were developed to understand the role, process and impact of the review meetings

Research Question	Objectives
RQ1: How Review meetings are planned?	To identify the steps involved in the planning phase for review meeting
RQ2: How to inspect increment in review?	To identify the procedure of inspecting the last sprint and to design increments in review meeting
RQ3: Define Reevaluation in review?	To identify the process of reevaluation (when to do and what to be done) in review meeting
RQ4: How feedbacks are conducted in review?	To identify the method of taking feedbacks in review meeting
RQ5: Define types of meeting in review?	To identify the need and investigate the types of review meeting

## 4. Research Findings

Various authors have worked on the topic of review meetings in context of agile methods. In [Vranic and Laslop \(2016\)](#) and [de Gea et al. \(2015\)](#), authors discussed the requirement engineering perspective and have showed the importance of review meetings and discussed various types. The work was focused on the interaction of teams during meetings and the collaboration among team members. To investigate this, the transcripts of daily meetings at two companies were analyzed with intention to understand interaction of team members. In [Hossain et al. \(2009\)](#) a case study based analysis is presented with the aim of revealing the categories of communication at daily meetings to recognize how team members interact at these meetings. The parameters most of these researches based on, are, team interactions, decision making, risk mitigation, daily review, and interviews. Authors showed the importance of scrum meetings, its processes and practices in overall industry and produced a very informative study on risks in scrum for using GSD projects ([Srinivasan and Lundqvist, 2009](#)). They talked about risk mitigation and factors that emphasize on scrum planning and meetings in GSD projects. In [Paetsch et al. \(2003\)](#), authors have discussed the difference among traditional requirement engineering process and formal agile methods by focusing on review meetings, process of daily stand up meetings, and interviews sessions with customers. In this review, we are more focused on adaptability, integration in increment,

reevaluation, planning phases and types of meetings in different agile methods to evaluate their effectiveness towards achieving CVD.

#### 4.1. Planning for review meetings (RQ1)

Planning considers the overall project plan and the results of the earlier cycle and sets out a plan for the next week discrete development time. The planning meeting is less open than the review, as it is more concerned with internal team activities rather than disseminating information to wide audience. Planning steps taken up in different agile methods are given below:

- **Scrum:** Due to tracking and response to change, short-range plans adapt at every 24 hours. Daily scrum meetings (~15 minutes) are conducted to give accurate estimates, plans, and proper tracking ([Sutherland and Schwaber, 2010](#)).
- **Scrumban:** Scrumban holds daily meetings but there are no sprint or release planning meetings and retrospectives in scrumban. Scrumban embraces on-demand planning ([Reddy, 2015](#)).
- **Nexus:** Right representatives from each scrum team meet to discuss and review the refined product backlog. They select product backlog items for each team. Each scrum team then plans its own sprint, interacting with other teams as appropriate ([Bittner et al., 2017](#)).
- **EVO:** The plan consists of promising solution that are included on weekly basis and expressed by using an Impact Estimation (IE) table. The



solutions were evaluated with respect to value for clients versus cost of implementation (Johansen and Gilb, 2005).

- **LeSS:** Sprint planning consists of two parts. In sprint planning one, selection of ready items offered by the product owner are focused, covering up remaining questions, and definition of the sprint goals. A plan of work to get to 'done' for each item is created in second sprint planning (Larman and Vodde, 2016).

#### 4.2. Increments in review (RQ2)

After every iteration/ sprint a review is conducted to inspect incremented version and to update the product backlog if needed. Increments handling in different agile methods are given below:

- **Scrum:** At the end of each iteration, reviews are conducted but only for newly implemented features. This provides feedback early enough so that integration can be done in the next iteration.
- **Scrumban:** As scrumban holds daily meetings and also embraces on-demand planning, the increments are adjusted according to selected planning method (Ladas, 2009).
- **Nexus:** The nexus integration team is accountable for ensuring that a "Done" (combined work completed) is produced at least once every sprint (Bittner et al., 2017; Shahin et al., 2017).
- **EVO:** Continuous Integration was introduced with which the developers get their work out onto the test servers every week (Johansen and Gilb, 2005).
- **LeSS:** The output of every sprint is called a potentially shippable product increment. The work of all the teams must be integrated before the end of every sprint and the integration must be take place during the Sprint (Larman and Vodde, 2016).

#### 4.3. Reevaluation and retrospectives in review (RQ3)

Agile methods have a frequent reevaluation of plans with emphasis on face-to-face communication and sparse use of documents. Discussion on how these plans are handled by different agile methods is given below:

- **Scrum:** Teams had some experience with a clear development process and that they can influence it. The iteration planning meetings are started again with a retrospective. Most contributions were about how they develop application and how they can actually improve (Sutherland and Schwaber, 2010).
- **Scrumban:** Scrumban hold daily meetings, but there are no sprint or release planning meetings and retrospectives in Scrumban. Scrumban embraces on-demand planning (Ladas, 2009; Reddy, 2015).
- **Nexus:** The nexus sprint retrospective is a formal opportunity for nexus to inspect itself and produce a plan for improvements for the next sprint to

ensure continuous improvement. The nexus sprint retrospective occurs after the nexus sprint review and prior to the next nexus sprint planning (Bittner et al., 2017).

- **EVO:** It is expected that in each iteration there is a re-evaluation of solutions which yield the highest value to cost ratio, guided by feedback and estimates (Johansen and Gilb, 2005).
- **LeSS:** At the end of the sprint, all the teams have their individual retrospectives in which they also brainstorm about larger obstacles that are impeding them and put them on the organizational improvement backlog. The overall retrospective is a new meeting in LeSS which discusses cross-team, organizational and systemic problems within the organization (Larman and Vodde, 2016; Shahin et al., 2017).

#### 4.4. Feedbacks in review (RQ4)

Feedback loops are the driving factors in agile methods as it directly improves the team's productivity (Srinivasan and Lundqvist, 2009). Feedback handling in different agile methods is given below:

- **Scrum:** By the end of each sprint, stakeholders are shown what has been built. Feedbacks are obtained that can be included in the next sprint. Working product at the end of the sprint is really "done," if the completely tested and potentially shippable code is included (Sutherland and Schwaber, 2010).
- **Scrumban:** The feedback mechanisms in scrumban are team standup, service delivery, operation review, and strategy review (Ladas, 2009).
- **Nexus:** The nexus sprint goal is an objective set for eachsprint. The nexus should demonstrate in a sprint that the functionality is "Done" at the nexus sprint review to receive the next goal after the stakeholder feedback (Bittner et al., 2017).
- **EVO:** Feedbacks are conducted in EVO to get the highest value priority list from customer (Johansen and Gilb, 2005; Shahin et al., 2017).
- **LeSS:** At the end of sprint, the sprint review is an inspect-adapt point. Customers and stakeholders examine what has been built during the sprint and discuss feedbacks (Larman and Vodde, 2016).

#### 4.5. Types of meetings in review (RQ5)

Meeting types explains the overall process required by the method. The meeting types of different agile methods are given below:

- **Scrum:** Five types of meeting take place: daily standup, sprint planning, sprint review, sprint retrospective and product backlog refinement (Sutherland and Schwaber, 2010)
- **Scrumban:** In scrumban, meetings are optional. They can be avoided entirely or agreed upon on a regular or on demand basis (Reddy, 2015).

- **Nexus:** Three types of meeting take place: nexus daily standup, nexus sprint review and nexus sprint retrospective (Bittner et al., 2017).
- **EVO:** A combination of short and long term definition meetings periodically taking place (Johansen and Gilb, 2005).
- **LeSS:** Five types of meeting take place: sprint planning one, sprint planning two, daily scrum, sprint review and sprint retrospective (Larman and Vodde, 2016; Shahin et al., 2017).

#### 4.6. Comparative analysis

The comparative analysis of agile methods confirms that these methods employ different strategies of review meetings. We tried to find the answer of each research question in light of the literature review. Though, each method has its strengths and weaknesses on the basis of which it can be good or bad in different circumstances but here our focus was more towards CVD. As depicted in Table 3, the selected agile methods have shown the highest level of flexibility for the reviews and it is a fact that CVD needs lots of review, as the perspective of value for the stakeholders may change which is true especially for customers.

**Table 3:** Methods identified against the research question

Research Questions: Activities	Selected Agile Method
RQ1: Planning	Scrumban
RQ2: Increments Inspection	EVO
RQ3: Retrospective	Scrumban
RQ4: Feedbacks	Scrumban
RQ5: Meeting Types	Scrumban

Findings acquired after research depicts that Scrumban is comparatively a better model in different aspects while conducting Reviews. For RQ1 where Planning was concerned, Scrumban is supposed to be the best choice as the planning meeting is more concerned with internal team activities rather involving external environments, sometimes long term planning or frequent plannings may create disruption while transmitting information, also additional information may lead towards ambiguities among internal team, Scrumban, unlike other models does not support release planning meetings and retrospectives, it rather believes on demand planning which ultimately lift confidence of internal team as team has series of tasks to focus on rather planning for the format. To inspect incremented version (RQ2) it is found that more or less Nexus, LeSS and Scrum focuses on integration and inspection at the end of each sprint which sometimes create liability for upcoming sprint while EVO believes in Continuous Integration that sums up the entire sprint activity every week thus completing targeted and tested work at end of each week.

RQ3 was about how reevaluation of plans are catered in agile methods. Scrum, Nexus, EVO and LeSS reevaluate at start of next sprint or end of ongoing sprint. Sometimes reevaluation just deals with formal assessments that on particular, at that

time does not required or a team has something more important than reevaluate the trend. Scrumban has comparatively better option as it believes in on demand reevaluations without following trend. Feedbacks (RQ4) are the most promising factor ensuring for continuous excellence and Scrumban has different activities that ensure continuous feedback from stakeholders and team. Rather conducting feedbacks at end of sprint and assuming it a reevaluation for next sprint like other models, Scrumban conducts team standups, service delivery, operation review and strategy review and strengthen the sprint right on time. Similarly, Types of meetings (RQ5) each model has its own style of meeting with different agenda and different requirements and these meetings have fixed culture. Scrumban allows a complete denial from meeting IF there is no any need. It is a foremost feature of Scrumban that provides confidence and trust on individuals.

#### 5. Conclusion

This paper presents a detailed discussion on different agile methods in terms of review meetings. It is noted that the sprint review meeting is important to the development team as it is an opportunity for the team to show its work openly and get acknowledgment from the stakeholders. Development team morale is increased by such meetings. The sprint review establishes a firm level of sociable comparative competition between scrum teams that keeps everyone focused. It is actually equivalent to a user acceptance test.

The processes of review meetings are evaluated in five distinct areas and Scrumban came on top as compared to other methods such as scrum, EVO, LeSS and nexus. However, due to continuous integration, EVO came on top in the area of increment inspection. It is clear that scrumban is the better option in the agile methods for CVD but CVD cannot work with the scrumban's way to increment inspection. Hence it is concluded that to move towards CVD with agile methods, scrumban should be strengthened with EVO's increment inspection process. As a future work, we intend to propose a hybrid mechanism focused on CVD by merging scrumban and EVO.

#### Compliance with ethical standards

#### Conflict of interest

The authors declare that they have no conflict of interest.

#### References

- Bittner K, Kong P, Naiburg E, and West D (2017). The nexus framework for scaling scrum: Continuously delivering an integrated product with multiple scrum teams. Addison-Wesley Professional, Boston, USA.

- Boehm B (2003). Value-based software engineering: Reinventing. ACM SIGSOFT Software Engineering Notes, 28(2): 1-3. <https://doi.org/10.1145/638750.638775>
- Brereton P, Kitchenham BA, Budgen D, Turner M, and Khalil M (2007). Lessons from applying the systematic literature review process within the software engineering domain. Journal of Systems and Software, 80(4): 571-583. <https://doi.org/10.1016/j.jss.2006.07.009>
- Conboy K (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. Information Systems Research, 20(3): 329-354. <https://doi.org/10.1287/isre.1090.0236>
- de Gea JMC, Nicolás J, Fernández-Alemán JL, Toval A, Ebert C, and Vizcaino A (2015). Commonalities and differences between requirements engineering tools: A quantitative approach. Computer Science and Information Systems, 12(1): 257-288. <https://doi.org/10.2298/CSIS131201001C>
- Fagerholm F, Guinea AS, Mäenpää H, and Münch J (2014). Building blocks for continuous experimentation. In the 1<sup>st</sup> International Workshop on Rapid Continuous Software Engineering, ACM, Hyderabad, India: 26-35. <https://doi.org/10.1145/2593812.2593816>
- Fitzgerald B and Stol KJ (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123: 176-189. <https://doi.org/10.1016/j.jss.2015.06.063>
- Hossain E, Babar MA, Paik HY, and Verner J (2009). Risk identification and mitigation processes for using scrum in global software development: A conceptual framework. In The 16<sup>th</sup> Asia-Pacific Software Engineering Conference, IEEE, Penang, Malaysia: 457-464. <https://doi.org/10.1109/APSEC.2009.56>
- Jayatileke S, Lai R, and Reed K (2018). Managing software requirements changes through change specification and classification. Computer Science and Information Systems, 15(2): 321-346. <https://doi.org/10.2298/CSIS161130041J>
- Johansen T and Gilb T (2005). From waterfall to evolutionary development (Evo): How we rapidly created faster, more user-friendly, and more productive software products for a competitive multi-national market. International Council on Systems Engineering (INCOSE), San Diego, USA. <https://doi.org/10.1002/j.2334-5837.2005.tb00781.x>
- Ladas C (2009). Scrumban-essays on kanban systems for lean software development. Lulu Press, Morrisville, USA.
- LeMan C and Vodde B (2016). Large-scale scrum: More with LeSS. Addison-Wesley Professional, Boston, USA.
- Matkovic P, Maric M, Tumbas P, and Sakal M (2018). Traditionalisation of agile processes: Architectural aspects. Computer Science and Information Systems, 15(1): 79-109. <https://doi.org/10.2298/CSIS160820038M>
- Paetsch F, Eberlein A, and Maurer F (2003). Requirements engineering and agile software development. In the 12<sup>th</sup> IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, Linz, Austria: 308-313. <https://doi.org/10.1109/ENABL.2003.1231428>
- Phillips A, Sens M, De Jonge A, and Van Holsteijn M (2015). The IT manager's guide to continuous delivery: Delivering business value in hours, not months. Xebia Labs, Burlington, USA.
- Reddy A (2015). The Scrumban [r]evolution: Getting the most out of agile, scrum, and lean Kanban. Addison-Wesley Professional, Boston, USA.
- Ries E (2011). The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses. Crown Books, Largo, USA.
- Shahin M, Babar MA, and Zhu L (2017). Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. IEEE Access, 5: 3909-3943. <https://doi.org/10.1109/ACCESS.2017.2685629>
- Srinivasan J and Lundqvist K (2009). Using agile methods in software product development: A case study. In the 6<sup>th</sup> International Conference on Information Technology: New Generations, IEEE, Las Vegas, USA: 1415-1420. <https://doi.org/10.1109/ITNG.2009.334>
- Sutherland J and Schwaber K (2010). The scrum papers: Nuts, bolts and origins of an agile process. Available online at: <https://bit.ly/2yS501D>
- Vranic V and Laslop M (2016). Aspects and roles in software modeling: A composition based comparison. Computer Science and Information Systems, 13(1): 199-216. <https://doi.org/10.2298/CSIS151207065V>