



---

# Systematic literature review on the impacts of agile release engineering practices

Teemu Karvonen , Woubshet Behutiye , Markku Oivo , Pasi Kuvaja

[Show more](#)

Share Cite

---

<https://doi.org/10.1016/j.infsof.2017.01.009>

[Get rights and content](#)

---

## Abstract

## Context

Agile release engineering (ARE) practices are designed to deliver software faster and cheaper to end users; hence, claims of such impacts should be validated by rigorous and relevant empirical studies.

## Objective

The study objective was to analyze both direct and indirect impacts of ARE practices as well as to determine how they have been empirically studied.

## Method

The study applied the systematic literature review research method. ARE practices were identified in empirical studies by searching articles for “rapid release,” “continuous integration,” “continuous delivery,” and “continuous deployment.” We systematically analyzed 619 articles and selected 71 primary studies for deeper investigation. The impacts of ARE practices were analyzed from three viewpoints: impacts associated with adoption of the practice, prevalence of the practice, and success of software development.

## Results

The results indicated that ARE practices can create shorter lead times and better communication within and between development teams. However, challenges and drawbacks were also found in change management, software quality assurance, and stakeholder acceptance. The analysis revealed that 33 out of 71 primary studies were casual experience reports that had neither an explicit research method nor a data collection approach specified, and 23 out of 38 empirical studies applied qualitative methods, such as interviews, among practitioners. Additionally, 12 studies applied quantitative methods, such as mining of software repositories. Only three empirical studies combined these research approaches.

## Conclusion

ARE practices can contribute to improved efficiency of the development process. Moreover, release stakeholders can develop a better understanding of the software project's status. Future empirical studies should consider the comprehensive reporting of the context and how the practice is implemented instead of merely referring to usage of the practice. In addition, different stakeholder points of view, such as customer perceptions regarding ARE practices, still clearly require further research.

---

## Introduction

In software development, fast, incremental deliveries involve lightweight, efficient practices for continuous release planning [1] and release engineering [2], [3]. This paper focuses on engineering by synthesizing empirical studies for *agile release engineering* (ARE) practices. To the best of our knowledge, the concept of ARE has not been used before in other scientific papers. We use it in this paper to map the research topic and to incorporate the investigated release engineering practices that are involved in a “*release engineering pipeline*” [3]. Hence, by ARE practices, we mean contemporary software *integration, testing, deployment* and *release* practices that are often applied in modern release engineering [3]. Many of these practices originate from agile software development methodologies such as extreme programming (XP) [4]. From the research point of view, ARE practices derive from the theories of *agile and lean software development* [5], *release engineering* [2], [3], [6] and *continuous software engineering* [7], [8] research disciplines. According to Adams et al. [6], release engineering “deals with all activities in between regular development and delivery of a software product to the end user, i.e., integration, build, test execution, packaging and delivery of software.” Continuous software engineering is an emerging subtopic in software engineering (SE) that is focused on continuous experimentation, innovation and the elimination of discontinuities within and between the developmental, operational and business

strategy functions. Fitzgerald and Stol [8] associate concepts in continuous software engineering with concepts of classic “lean thinking” [9] such as “value and waste,” “flow and batch size,” “autonomation and building-in quality” and “Kaizen and continuous improvement.”

Modern ARE practices are supposed to aid in delivering software faster and cheaper to end users; hence, claims of such impacts should be validated by rigorous and relevant empirical studies. In this systematic review, the objective is to understand the direct and indirect impacts of ARE practices. By investigating the direct and indirect impacts we emphasize the notion that “impacts may be desired already according to the explicit method rationale(s), or they may be unexpected, sometimes even unwanted[10].” In addition, we sought to evaluate primary studies to understand how the impacts of ARE practices have been investigated in empirical studies. The research questions are:

RQ1: What are the direct and indirect impacts of ARE practices?

We break down the main question into three sub-questions as follows:

RQ1a: What are the impacts associated with adoption of ARE practices?

RQ1b: What is the prevalence of ARE practices?

RQ1c: What are the impacts of ARE practices on the success of SW development?

In addition, we define a second main research question to understand how these impacts have been investigated:

RQ2: How have ARE practices been investigated in empirical studies?

In this study, we systematically searched and analyzed empirical studies investigating 1) *continuous integration* (CI) [11], [12], 2) *continuous delivery* (CD) [12], 3) *rapid release* (RR) [13] and 4) *continuous deployment* (CD2) [11], [12]. We analyzed and clustered studies by topic and research approach, and outlined a checklist for analyzing software development capabilities for CD2 in the context of software-intensive products. We applied a systematic literature review (SLR) [14] method that allowed us to critically compare, evaluate and synthesize the primary studies. Our main selection criterion for the primary studies was that they were conducted in real software development contexts. Literature reviews, mapping studies, opinion papers and small-scale experiments with students are not included in our analysis. To the best of our knowledge, a systematic synthesis focusing on the impacts of ARE practices has not previously been undertaken, although some of the practices have been synthesized

either separately or from a different research question point of view, as we explain in more detail in the following section. With this systematic review, we aim to provide a reliable overview of the current state of existing empirical studies for ARE practices that may help in terms of scoping and planning future studies. Our study also helps practitioners to better understand the impacts and capabilities associated with ARE practices. Finally, this paper aims to contribute to the theorizing on software development practices [10] for ARE. The concepts used in this paper (i.e., *learning*, *practice*, *development context*, *rationale*, *impact* and *theory*) conform to definitions used for the Coat Hanger model [10].

---

## Access through your organization

Check access to the full text by signing in through your organization.

Access through **your organization**

---

## Section snippets

### Background

ARE practices aim to support the agile principle of “early and continuous delivery of valuable software [15].” Early and continuous deliveries allow mechanisms for fast feedback and transparency of the development process, allowing stakeholders to continuously review and evaluate the state of the system under development and, if needed, to make adjustments to the priority and content requirements accordingly. CI practice originated from the agile XP methodology. Beck and Andres [4] summarizes ...

### Research design and implementation

We applied an SLR [14], [31] method in our study. The research was conducted in seven key phases: 1) Research planning; 2) Piloting of the search; 3) Searching articles; 4) Inclusion and exclusion of articles; 5) Quality assessment of primary studies; 6) Analysis and clustering of primary studies; and 7) Reporting the results of the SLR. The first version of the research plan and protocol was written and reviewed in a research group comprising all the authors. Later, the plan and protocol were ...

### Clustering and assessment of the primary studies

This section presents the results of the assessment on the rigor of the primary study and its relevance to the industry. In addition, we cluster articles based on their research

approach and main topics in terms of providing an answer to RQ2 (How have ARE practices been investigated in empirical studies?).

After undertaking the inclusion steps presented in the previous section, the articles were read multiple times and analyzed using the NVivo tool [32] to extract data for the synthesis and ...

## Analysis of the impacts of ARE practices

This section elaborates the key findings from the primary studies for providing the answer to RQ1 (What are the direct and indirect impacts of ARE practices?). To structure the results we have specified three sub-questions: RQ1a (What are the impacts associated with adoption of ARE practices?), RQ1b (What is the prevalence of ARE practices?) and RQ1c (What are the impacts of ARE practices on the success of SW development?).

In Section 5.1, we focus on adoption of the practice. This viewpoint ...

## Discussion on research implications

This section continues with the results' interpretation and further discusses possible implications of the study. In the discussion, we also elaborated on existing research gaps and offered approaches for conducting future empirical studies on ARE practices. ...

## Limitations of the study and future research ideas

In literature reviews, the main validity threats and limitations typically lie in the search, selection and analysis of the primary studies. As explained earlier, we performed search, selection and quality evaluations with two authors and cross-checked each decision to avoid the subjective selection of primary studies to be used in the analysis. Furthermore, the analysis and findings of the paper were reviewed by all of the authors. Our search was carried out using the specific concepts of ...

## Conclusion

This paper presented a synthesis of the impacts of ARE practices by analyzing empirical studies that had investigated continuous integration, continuous delivery, continuous deployment, and rapid release. One of our main research objectives was to understand how ARE practices have been investigated in empirical studies (RQ2). Our results indicated that 33 out of 71 primary studies were experience reports that had neither an explicit research method nor a data collection approach specified. ...

## Acknowledgements

This work was supported by TEKES as part of the Need for Speed Project (<http://www.n4s.fi/> ↗) of DIMECC (Digital, Internet, Materials & Engineering Co-Creation). ...

[Recommended articles](#)

## Research data for this article

Open Data

for download under the [CC BY licence](#) ↗



Supplementary Data

(ZIP, 4KB)

[Download data](#)

[Further information on research data](#) ↗

---

## References (99)

T. Päivärinta *et al.*

[Theorizing about software development practices](#)

Sci. Comput. Program. (2015)

D. Ståhl *et al.*

[Modeling continuous integration practice differences in industry software development](#)

J. Syst. Softw. (2014)

B. Kitchenham *et al.*

[A systematic review of systematic review process research in software engineering](#)

Inf. Softw. Technol. (2013)

G.G. Claps *et al.*

[On the journey to continuous deployment: technical and social challenges along the way](#)

Inf. Softw. Technol. (2015)

L. Chen

## Continuous delivery: huge benefits, but challenges too

IEEE Softw. (2015)

G. Ruhe

### Product Release Planning Methods, Tools and Applications

(2010)

B. Adams *et al.*

### The practice and future of release engineering: a roundtable with three release engineers

IEEE Softw. (2015)

B. Adams *et al.*

### Modern release engineering in a nutshell: why researchers should care

K. Beck *et al.*

### Extreme Programming Explained: Embrace Change

(2004)

M. Poppendieck *et al.*

### Lean software development: a tutorial

IEEE Softw. (2012)



View more references

---

## Cited by (31)

### Management of quality requirements in agile and rapid software development: A systematic mapping study

2020, Information and Software Technology

#### Citation Excerpt :

...Despite the fact that Alsaqaf et al. [10] provided interesting insights, their focus on large-scale distributed agile projects and limited number of primary studies (the study is based on 60 primary studies, while our study found 156 primary studies on the topic) limit their results. Regarding the way in which QRs are managed in RSD, it has not been examined in any of the previous secondary studies of RSD [8,23–25]. Compared to other related secondary studies, our study provides more concrete contributions to the literature, including the following:...

Show abstract

## Test Case Prioritization in Continuous Integration environments: A systematic mapping study

2020, Information and Software Technology

[Show abstract](#) ✓

## To the attention of mobile software developers: guess what, test your app! ↗

2019, Empirical Software Engineering

## Current state of research on continuous experimentation: A systematic mapping study ↗

2018, Proceedings 44th Euromicro Conference on Software Engineering and Advanced Applications Seaa 2018

## Continuous experimentation and A/B testing: A mapping study ↗

2018, Proceedings International Conference on Software Engineering

## DevSecOps: A multivocal literature review ↗

2017, Communications in Computer and Information Science



[View all citing articles on Scopus](#) ↗

---

[View full text](#)

© 2017 Elsevier B.V. All rights reserved.





