Big Data Analytics in Agile Software Development: A Systematic Mapping Study

Katarzyna Biesialska[a], Xavier Franch[a], Victor Muntés-Mulero[b]

[a]*Universitat Politècnica de Catalunya, Barcelona, Spain*
[b]*Beawre Digital S.L., Barcelona, Spain*

Abstract

Context*: Over the last decade, Agile methods have changed the software development process in an unparalleled way and with the increasing popularity of Big Data, optimizing development cycles through data analytics is becoming a commodity. ***Objective: Although a myriad of research exists on software analytics as well as on Agile software development (ASD) practice on itself, there exists no systematic overview of the research done on ASD from a data analytics perspective. Therefore, the objective of this work is to make progress by linking ASD with Big Data analytics (BDA). ***Method***: As the primary method to find relevant literature on the topic, we performed manual search and snowballing on papers published between 2011 and 2018. ***Results***: In total, 65 primary studies were selected and analyzed. Our results show that BDA is employed throughout the whole ASD lifecycle. The results reveal that data-driven software development is focused on the following areas: code repository analytics, defects/bug fixing, testing, project management analytics, and application usage analytics. ***Conclusions***: As BDA and ASD are fast-developing areas, improving the productivity of software development teams is one of the most important objectives BDA is facing in the industry. This study provides scholars with information about the state of software analytics research and the current trends as well as applications in the business environment. Whereas, thanks to this literature review, practitioners should be able to understand better how to obtain actionable insights from their software artifacts and on which aspects of data analytics to focus when investing in such initiatives.

Keywords: Agile software development, Software analytics, Data analytics, Machine learning, Artificial intelligence, Literature review

1. Introduction

Over the last decade, Agile methods have changed the software development process in an unparalleled way. As opposed to traditional, plan-driven models of software development (e.g. waterfall model), where processes are organized in a series of sequentially ordered stages, Agile software development (ASD) entails collaborative development with swift and incremental iterations. As a result, adaptability to frequently changing requirements and a strong emphasis on delivering value to customers represent the crux of ASD and have driven its wide acceptance among software practitioners in the last years. Furthermore, this paradigm shift from plan-driven software development processes to ASD accorded with social and technological advances.

When it comes to applying data-driven approaches to improve the work of software teams, software development practice has been in the midst of major upheaval for some years now. Plan-driven software development approaches follow, as the name suggests, a pre-defined plan. Being reactive in their nature, plan-driven software development processes are less susceptible to immediate feedback than ASD. Hence, their need for timely information is seemingly lower. Consequently, with the increasing popularity of Big Data, which throughout the years has successfully entered the realm of computer science and business environment alike, software practitioners can now efficiently use it to improve software development processes. Improving the productivity of software development teams is one of the most important objectives every software company faces. Optimizing development cycles through data analytics can not only streamline the company's day-to-day business operations but also help the business venture take over its competitors in the longer term. A data-driven approach may enhance the decision-making processes in the company by providing insights, for instance, on the following topics: estimating whether a project is on track with its budget and timeline, predicting the number of tasks that can be accomplished within one cycle, prioritizing software feature releases based on software usage metrics, or advising on the best team composition for a particular project.

A myriad of research exists on software analytics, software metrics, as well as on ASD practice. Hence, a vast number of systematic literature studies summarizing the research literature published on each of those topics alone (e.g. [31, 8, 33, 26, 13, 24]). Systematic literature reviews in ASD focus mainly on topics such as global software engineering, usability, management-oriented approaches [26], and, at the same time, refrain from discussing data-driven approaches to Agile development methods. Therefore, our objective for this work is to make significant progress by linking ASD with data analytics. In this paper, we examine studies on BDA in the context of ASD. By means of a systematic mapping study, we aim at getting a comprehensive understanding of the state of research on data-driven ASD. Therefore, we first examine studies concerning the topic. After performing a thematic analysis, we develop a classification framework explaining different ways how software development teams, and organizations in general, adopt BDA to improve ASD.

The remainder of this paper is structured as follows: Section 2 is meant as an introduction to the topic of data analysis within the realm of ASD. In this section, we also explain our approach to conducting a systematic literature study. In section 3, we discuss how we conducted a systematic literature study – we provide a detailed description of search methods, inclusion and exclusion criteria, and a list of central journals and conferences in our research area. Section 4 describes our results as well as topology and a classification framework proposed in this paper. Section 5 provides a discussion of the results and the limitations of the study. Finally, Section 6 concludes the paper.

2. Background

2.1. Big Data Analytics

Although the phenomenon of Big Data is not new – it has emerged in computer science in the late 1990s – the term itself still causes some confusion as precise boundaries for Big Data are hard to define [20]. In general, Big Data is often described along three dimensions, the so-called *three Vs* – *volume* (large datasets), *velocity* (high-frequency data influx), and *variety* (data from heterogeneous data sources). These three characteristics are frequently extended to include more factors, such as *veracity* (data accuracy and reliability), *variability*, *value* (the potential of data to support decisions) or *visualization* (visual representation of data). Therefore, the whole concept of Big Data can be understood as a process of gathering, storing, analyzing, and extracting knowledge from high-volume and/or complex data, often by means of Machine Learning (ML)/Artificial Intelligence (AI) algorithms. In that context, the term Big Data Analytics (BDA) can be defined as a sub-step in the Big Data process, focused on gaining insights through advanced analytics techniques – "*Big data analytics is the process of using analysis algorithms running on powerful supporting platforms to uncover potentials concealed in big data, such as hidden patterns or unknown correlations*" [27]. Therefore, BDA "*must effectively mine massive datasets at different levels in real-time or near real-time - including modeling, visualization, prediction, and optimization*" [27]. Business analytics for years has been viewed from three complementary perspectives, i.e. *descriptive*, *predictive*, and *prescriptive* [2, 20]. The most recent addition to this group is *adaptive analytics*, which instead of only analyzing the past, responds in real-time to the actions of a user. By keeping humans in the loop, this type of analytics is the embodiment of real-time decision making.

2.2. Agile Software Development

ASD is said to be the answer to multiple limitations of plan-driven approaches [14, 43, 24]. Specifically, frequent changes of the system's requirements, close collaboration between developers and their stakeholders, iterative development delivered in increments are the cornerstone of ASD. The main phases of ASD lifecycle (SDLC), are: the *concept*, *inception*, *construction*, *release*, *production*, and *retirement* phases [9, 3]. The first phase, which is the *concept* phase, is a stage where a project is envisioned. This phase can be considered the planning phase. The *inception* phase entails assembling the team, finding a project sponsor, and securing the funding, framing initial requirements. It resembles the analysis phase. The *iteration/construction* phase is about working products, hence the following activities take place in this phase: development of the software, requirements definition, and customer feedback loop. This phase is the design and implementation phase. In the *release* phase testing plays an important role. This is the testing and integration phase. The *production* phase is related to the maintenance of the software in a production environment. This is the maintenance phase. Finally, the *retirement* phase relates to software decommission. ASD is an iterative process, where each iteration (known as a sprint in Scrum) is expected to create and deliver a working piece of software as soon as possible. Therefore, a typical iteration encompasses requirement, design and development, testing, delivery, and assessment stages, which form a loop, as numerous iterations might be needed to release a working product or a new feature. Notwithstanding breaking software development work into iterations/sprints, the aforementioned ASD phases may vary, also because initially, Agile methods were created with small organizations in mind. With time, however, their large-scale variants have been adopted by large organizations [41, 22], e.g. the Scaled Agile Framework (SAFe) [5], Large-Scale Scrum (LeSS) [4], Scrum-of-Scrums, Nexus, Scrum at Scale, or Disciplined Agile Delivery (DAD) [50]. Large organizations face a challenge related to coordination and communication between several development teams or different organizational units, often globally distributed [41]. Therefore, data analytics in software engineering can increase the ability of organizations to understand how to improve their software development processes.

3. Research Method

A systematic mapping (SM) study (also referred to as a scoping study), similar to a systematic literature review (SLR), serves as a summary of scientific articles published on a particular research topic. In light of this difference, the reason why we decided to perform an SM study of the topic, rather than an SLR, is twofold. First, when performing initial analysis of our research domain and before starting an actual systematic review, we identified the involved domain as a broad research topic, for which an SM study seemed to be a more appropriate way of summarizing academic findings than an SLR. Second, in this article, we intend to make a classification of evidence and summarize the current state-of-the-art research related to applying data analytics to improve ASD practice. Our goal is not to answer very specific questions related to the said domain, but rather provide a general overview.

In this work, we followed a protocol devised exclusively for the purpose of this SM study – its in-depth description is provided in the following subsections. Below we present only a brief overview of steps undertaken by us in the course of conducting this SM study. We started off with defining our research questions (outlined in section 3.2) and a set of research papers that we identified as major contributions in the domain in recent years, that were published in leading journals and conferences in our research area (our search method is discussed in section 3.4). As BDA and ASD are fast-developing areas, we assumed that papers published between 2011 and the beginning of 2018 had cited all seminal pieces on the topic. The assumption is in accordance with findings of Meidan et al. [36] who, in their SM study, showed that the number of papers discussing Agile and Lean development processes almost doubled after 2010. In the same vein, Laanti et al. [34] argue that scientific and quantitative studies on Agile methods were still rare in 2011. Therefore, we based our search of primary studies on manual search in the selected top-tier publication venues, and then we performed both backward and forward snowballing from the selected papers. For papers identified as relevant through the first iteration of backward snowballing and forward snowballing, we reviewed the references appearing in those papers. Next, we applied the second iteration of the forward and backward snowballing technique, which turned out to be the final one as no new study was found after that (i.e. during the third iteration of the method) [17, 52].

3.1. Protocol Development

Similar to other scholars, e.g. [16, 25, 53], our review protocol was divided into separate, yet collectively exhaustive steps, as outlined in the list below:

- **Phase 1: Define**
  - defining the research questions
  - defining search criteria
  - defining search strategy
- **Phase 2: Perform search**
  - identifying similar secondary studies
  - finding a set of relevant publications (primary studies)
- **Phase 3: Conduct review**
  - validating the list of primary studies
  - extracting data
  - analyzing data
- **Phase 4: Document review**
  - synthesizing the findings
  - validating report findings
  - writing the review report

3.2. Research Questions

In order to gain a comprehensive understanding of how BDA is used in Agile development, the objective of this study was broken down into three main RQs. The following major RQs and their sub-questions are answered in this evaluation:

- **RQ1**: What studies discuss BDA and ASD?

  RQ1.1: When the studies have been published?

  RQ1.2: Where and in what types of venues have the studies been published?

  RQ1.3: What is the geographic distribution of the studies and their authors?

  RQ1.4: How are the selected studies distributed between industry and academia?

  RQ1.5: What are the companies discussed in the studies and/or what is their authors' affiliation?

  RQ1.6: What is the distribution of the studies with regard to the research results?

- **RQ2**: What are the approaches to using BDA in ASD?

  RQ2.1: What types of analytics have been used in the ASD domain?

  RQ2.2: What sources of data have been used?

  RQ2.3: What methods, models, or techniques have been utilized in the studies?

- **RQ3**: Is BDA used to improve ASD processes? If so, in what ways?

  RQ3.1: Where BDA is employed throughout the ASD lifecycle? What are the main research topics and sub-topics?

  RQ3.2: What Agile practices, techniques, or engineering practices have been used in the studies?

  RQ3.3: Do the approaches reported in the literature discuss tools? Have the tools been applied in practice?

RQ1: What studies discuss BDA and ASD? RQ1 and its sub-questions provide information on the demographics and the general type of papers selected in this mapping study. Based on that, we were able to aggregate research papers with respect to the quantity and present the distribution of publication over time (RQ1.1), venue (RQ1.2), or according to a location (RQ1.3) to determine trends. Furthermore, we mapped the frequencies of industrial versus academic publications as well as and authors' affiliation (RQ1.4) and companies that took part in case studies discussed in the selected primary studies (RQ1.5). In RQ1.6 we classified papers with respect to research results (i.e. the kind of contribution). This type of analysis was a stepping stone to the next two RQs and allowed us to identify research gaps in the field.

RQ2: What are the approaches to using BDA in ASD? The rationale behind the second RQ (including its sub-questions) is to classify BDA approaches according to the used methods. By approaches we understand types of analytics used in ASD domain (such as *descriptive*, *predictive*) [20] according to which we group studies in RQ2.1. Moreover, RQ2 helps with understanding what type of data triggers the adoption of BDA in ASD, as in RQ2.2 we discussed what data sources the selected studies leveraged. Whereas, RQ2.3 provides information on what kind of methods and techniques are utilized in BDA for ASD.

RQ3: Is BDA used to improve ASD processes? If so, in what ways? By answering RQ3, we were able to determine how BDA helps to improve ASD. In order to make our results more informative to a broad audience of people without specialized experience in any given sub-field, we decided to standardize our findings (where possible) by using the Software Engineering Body of Knowledge (SWEBOK) [15] sub-topics (RQ3.1). Such a thematic classification gave us confidence that: (1) we did not miss any important software engineering sub-field, (2) our work can be compared with studies conducted by other researchers, as the SWEBOK document is a widely accepted document in the software engineering community. In RQ3.2, we utilized a classification somewhat similar to the one used in VersionOne's Annual State of Agile Report [7], analyzing primary studies from three perspectives: Agile practices, Agile techniques, and Agile engineering practices. Furthermore, the last RQ sheds light on the applicability of the concepts discussed in the selected papers. RQ3.3 answers the question of whether the papers proposed tools and, if so, what concepts have been applied in practice.

3.3. Study Design

First, in section 3.5, we investigated whether any previously published secondary study discussed our topic of interest extensively. We concluded that no such study existed and we started work on our SM study. Our work analyzed papers published in 2018 and earlier, but not before 2011. Only central journals and conferences in our research area were evaluated in our work. The reason behind limiting selection results only to certain types of publications, such as those that appeared in selected top-tier venues, is the fact that such publication venues need to publish the most relevant high-quality studies, because otherwise they would not be able to sustain their ranking positions and a reputation for quality. Therefore, our initial question was: Is the paper published in a venue where relevant papers are published? To answer this question, we relied on our domain expertise. It appears that the selection of journals and conferences proposed in our paper is well-aligned with the one presented by Dingsøyr et al. [23] in their summary of a decade of ASD practice. Nevertheless, we would like to highlight that before deciding which journals, conferences, and workshops to include, we performed a broad search by topic across different publishing venues to find as many relevant journals, conferences, and workshops on the topic as possible. Overall, we reviewed publications from 11 journals, 12 conferences, and 7 workshop proceedings. However, only 8 journals, 10 conferences, and 5 workshops published works that we finally selected in our study (see Appendix A). No new venue emerged while applying a snowballing technique to find all relevant studies. The list of excluded venues due to lack of relevant primary studies is available in Appendix A.

3.4. Search Strategy

Before starting work on our literature survey, first, we had to investigate whether any previously published secondary study discussed the topic of interest, and if so, whether such a paper formulated research questions (RQs) similar to ours. Therefore, in order to eliminate the risk of duplicating existing literature review studies, we decided to search within a vast and diverse range of sources using an automatic search technique. Hence, our approach for finding existing literature reviews differed from the main one used in finding primary papers for our SM study. The main reason why we decided to deviate from our main approach was the need to cover all existing literature surveys in the involved domain, including those published in less reputable outlets. To perform an exhaustive search of surveys, we used the following digital libraries: *Thomson Reuters's Web of Science*, *Elsevier's Scopus*, *IEEExplore*, *ACM Digital Library*, *Microsoft Academic*, *Google Scholar* and *Semantic Scholar* (see Appendix A for the results of the search).

Furthermore, in order to have the best chance of finding similar review papers to ours, we also decided to cast a net wide with respect to strings used in our search query, which we describe in detail below. Specifically, for *analytics*, being a recurrent keyword for terms such as *software analytics*, *data analytics* and *big data analytics* (among others), we used an asterisk (*) to make the search wider. Following Kitchenham and Charters's [30] advice on constructing a search string, we defined ours as follows: (*survey* OR *review* OR *"mapping study"* OR *"systematic map" OR "state?of?the?art"* OR *"meta?analysis"*) AND (*"software engineering"* OR *"software development"* OR *"agile *development"*) AND (*"*analytics"* OR *"data-driven"* OR *"big data"*). As usual, later on the search string needed to be adapted to conform to the requirements of the selected digital libraries. For instance, not all electronic databases supported the question mark sign (?), matching exactly one character, or the asterisk character (*), matching zero or more characters. Hence, in such cases, we had to add missing characters or words where applicable.

The main source of noise in our search results was related to various applications of Agile methods to improve data analytics processes. After eliminating irrelevant studies, two literature reviews satisfied our conditions (discussed in section 3.5). None of them addressed the research topic extensively enough; hence, we concluded that our SM study could fill this gap.

Next, according to the next step in our protocol (presented in section 3.1), we used a hybrid search technique to find a set of relevant publications. The detailed description of this stage is presented below:

Step 0. Studies from selected venues: A basic search for all potential candidates. This means taking into account all papers published within the selected publishing venues within the timeframe specified in our mapping study. Manual search technique used.

Step 1. Title and abstract: Papers limited to those that, according to title and abstract, seem to be appropriate for the mapping study.

Step 2. Skimming: Quick read of papers reveals which studies are not relevant.

Step 3. Full Text: Full text read reveals precisely whether already selected papers meet the criteria.

After finding a set of relevant primary studies through manual search, we performed both backward and forward snowballing from the selected papers.

Step 4. 1st Round snowballing: we reviewed the references appearing in those papers. The first round of forward and backward snowballing.

Step 5. 2nd Round snowballing: For papers identified as relevant through the first iteration of snowballing, the second round of forward and backward snowballing was performed. After the second iteration of the forward and backward snowballing technique, no new study was found. Hence, the last iteration, which resulted in new studies included in our work, was the second iteration of snowballing.

Figure 1 shows the number of papers selected after each stage, and below is a description of each step. The full list of selected papers in this mapping study is presented in the Primary Studies section at the end of this paper.

From the study selection phase, we can conclude that

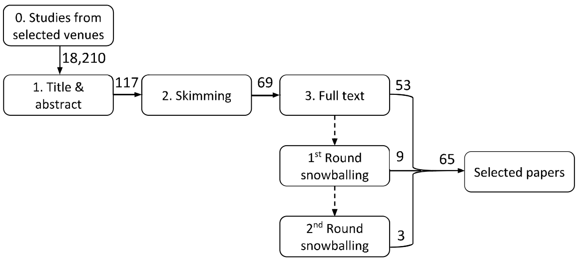


Figure 1: A primary study selection process

we have certainly not included all works published in the involved domain. However, the snowballing method is regarded as a comparable method in terms of the quality of results to other methods used in systematic studies [28]. Furthermore, building on Kitchenham and Charters's [30] argument that for an SM study, quality assessment of primary studies is not necessary, we decided not to use the quality assessment as one of our criteria for the study selection.

3.5. Comparison to Previous Studies

Although a quick look at the literature on software analytics can reveal a number of short articles, summaries of panel discussions, or editorials discussing the topic [37, 38, 39], however, none of these can be considered an extensive literature review of the research area. Therefore, in the course of preparing this SM, we examined a body of literature in order to find secondary or tertiary studies covering our topic of interest. The detailed explanation of our search process can be found in section 3.4.

Although none of the two papers discussed in this section is entirely focused on ASD, we identified them as being the closest to the research topic; hence, a brief description of those works is presented below. The first secondary study, identified as similar to ours in our search for similar works, is by Abdellatif et al. [8]. The authors carried out an SLR in which they selected and analyzed 19 primary studies (out of 135) in the period of January 2000 to December 2014. Those papers were used to identify software practitioners benefiting most from available software analytics studies, understand which software engineering domains are covered by software analytics studies as well as explain which artifacts are consumed by analytics and whether they are linked together somehow. Abdellatif et al. [8] tried to answer the following research questions: (1) "Which software practitioners does the available SA research target?", (2)"Which domains are covered by SA studies?", (3) "Which software artifacts are extracted?", (4) "If different artifacts are used, are they linked together?"

The authors of the SLR found that 90% of all studies targeted developers (47% developers only, 21% developers and project managers, 11% developers, testers and project managers, and 11% developers, portfolio, and project managers). Whereas, studies focused only on project managers constituted 10% and this was the only group identified to be targeted solely apart from developers. The review by Abdellatif et al. [8] showed that most available SA studies fell into one of the following domains: maintainability and reverse engineering (58%), team collaboration and dashboard (21%), incident management and defect prediction (11%), software analytics platform (5%), software effort estimation (5%). Furthermore, the researchers found that 47% of the studies used only one artifact as a data source (predominantly source code). 53% of works used more than one, including data from: issue tracking systems, version control systems, operating system and service transaction logs, project management, team wikis, as well as defect and bug reports (among others). All in all, the study concludes that it was not possible to find too many relevant or mature research in the software analytics field.

Also, Bagriyanik and Karahoca [13] performed an SLR. The study included papers from January 2010 to October or November 2015 (the end month remains unclear: in the abstract, it appeared to be October, whereas in the body, the authors mentioned November). The researchers selected 32 papers out of 326 studies found through Google Scholar. In their work, Bagriyanik and Karahoca tried to answer two main research questions: (1) "In which software engineering areas Big Data and Software Engineering are interacting and to what extent?" and (2) "Which software engineering artifacts are used for Big Data processing? What are the most frequently used artefacts?" [13]. Bagriyanik and Karahoca's RQ1 resembles our RQ3.1 (discussed in detail in section 3.2) as it aimed at finding software engineering areas that benefit from Big Data. The authors concluded that software quality, development, project management, human-computer interaction as well as software evolution and software visualization were the most active research areas in software engineering big data studies. Whereas RQ2 focused on determining software engineering data sources used in Big Data is similar to our RQ2.2. In terms of artifacts, source code, issue related, and operational data was identified as the most frequently used sources of data in the studies. This RQ was also covered in our study (see RQ2.2 in section 3.2). As we will explain later in section 4, the number of publications related to BDA in ASD has increased over the last few years, with the year 2014 being the caesura. As the work covered papers published over a span of 5 years starting from 2010, we can conclude that this literature review certainly requires an update. Furthermore, the two RQs formulated by the authors of [13], although helpful to understand the progress of the research area, do not cover all related factors such as the type of software analytics, methods and techniques used, to name just a few.

In sum, at the time of this writing, none of the previously published secondary studies (i.e. [8, 13]) covered the topic of interest extensively enough by answering research questions similar to those that our work tries to address. Therefore, with this work, we aim to fill this research gap.

3.6. Study Selection Criteria

The same research questions and inclusion/exclusion criteria are used in both search strategies: manual identification of papers in the selected venues and snowballing. The following criteria explain when a study was selected in our survey:

- **Inclusion criteria**
  - **IC1:** studies related to BDA and ASD
  - **IC2:** studies published from 2011 until 2018 (inclusive)
  - **IC3:** primary studies
  - **IC4:** studies published in selected publishing venues
- **Exclusion criteria**
  - **EC1:** studies not accessible in full-text
  - **EC2:** recaps of conferences, panel discussions, editorials
  - **EC3:** workshop papers published before 2014
  - **EC4:** duplicated studies

Although some works related to our research topic are published as grey literature (i.e. presentations at Agile conferences for practitioners, books, blog posts, podcasts), we decided, however, to not include this type of publications. Furthermore, articles published in the special sections of journals featuring the best works presented beforehand at conferences (i.e. articles which are revised and extended versions of the original papers) were considered on a case by case basis. Although, in general, this study reviewed papers published between 2011-2018 inclusive, however, for workshops we applied slightly modified criteria. Namely, workshops usually publish papers describing emerging results and work in progress. However, if considered important by the research community, they are later extended and published as fully-fledged conference or journal articles. Therefore, for that reason, in this work, we do not review workshop papers published before 2014. Although our work could also cover secondary studies, and hence be considered a tertiary study (or a tertiary review), however, we decided not to analyze papers that do not describe new or original experimental work.

In Appendix A, we present the number of selected papers published by major journals or presented at top conferences and peer-reviewed workshops, where papers on the research topic are featured.

3.7. Data Extraction and Synthesis

To answer our Research Questions (RQs), we prepared a form in which we tracked metadata together with other relevant information extracted from each reviewed primary study (see Appendix A).

4. Results

This SM study was performed according to the protocol described in section 3.1. After executing all its steps, we identified 65 studies on the usage of BDA for ASD.

4.1. Study Demographics and Trends (RQ1)

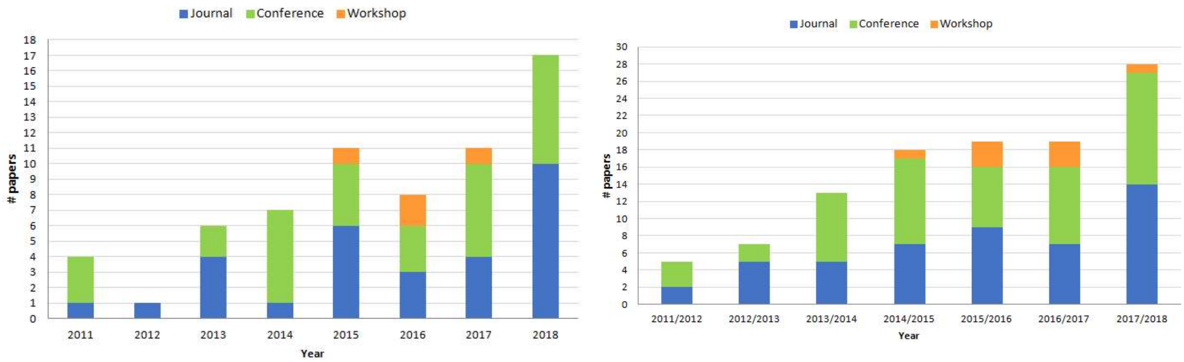
This section focuses on describing the demographic data of the selected primary studies, i.e. the distribution of studies with respect to authors' affiliation, country, publication year, and publication venue, among others.

4.1.1. Publication years (RQ1.1)

The detailed analysis of the distribution of the selected primary studies over the years is presented in Figure 2a. It can be noticed that there was a significant increase in the total number of publications after 2013, which is clearly visible in Figure 2b showing the volume of published research on the topic for two-year periods. This trend suggests a relatively growing research interest in the topic of BDA in the context of ASD, as the number of publications steadily increases every year, with a record-breaking year in 2018, when 17 (26.2%) research papers were published on the topic.

4.1.2. Publication venues (RQ1.2)

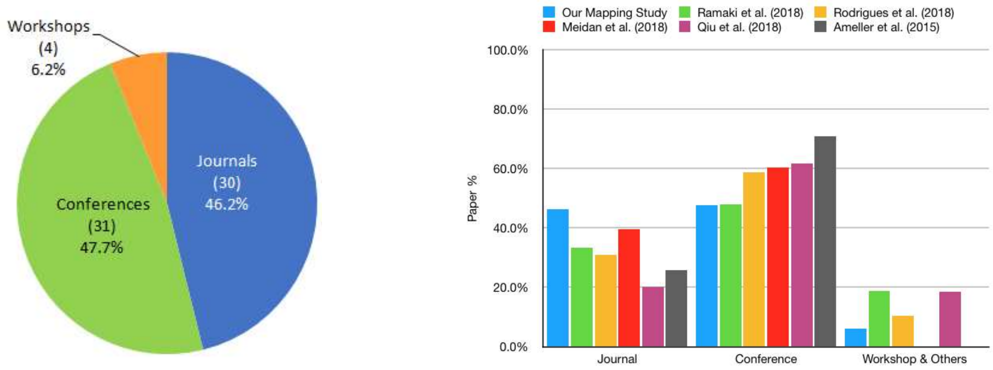
As discussed in section 3.3, the design of our mapping study implied doing a pre-selection of the most important venues in our research area as well as performing a snowballing process to cover relevant papers as diligently as possible. The selected primary studies



(a) per year

(b) per 2-year window

Figure 2: Distribution of studies over time



(a) our study

(b) comparison with other studies

Figure 3: Distribution of studies over venue type

were distributed over 22 publication venues (see Appendix A). We found that works presented at periodically held events (such as conferences and workshops) constituted the majority of studies in our area (see Figure 3a). Journals were, in general, slightly less popular (46.2%) among the scholars. Nevertheless, two of them (i.e., *IEEE Software* and *Information and Software Technology*) individually published the highest number of the selected primary studies, namely 16 (22.6%) split evenly.

To put it in context, we compared our numbers with the ratios reported by other mapping studies in the software engineering domain (see Figure 3b). Concretely, the data was obtained from a sample of 13 papers compiled by Ameller et al. [10] (by taking an average), which we extended with 4 recent SM studies [35, 45, 44, 46]. We observe that, similar to the results reported in the above-mentioned reference mapping studies, conferences combined together with workshops were the main forum (53.8%) for researchers to showcase their research. However, in our case, the difference between journals and conferences combined with workshops is very small. In fact, the percentage of publications in journals is the highest among the analyzed studies.

4.1.3. Publication countries (RQ1.3)

The authors of the 65 selected primary studies were distributed over 24 countries (see Table 1). It is evident that the USA produced the largest number of papers, having over 1.5 as many papers as the runner-up country – Canada, which makes North America the most prolific region followed by Europe. All continents are represented except Africa. Eleven authors (4.9% of all unique authors) represented more than one entity (not necessarily located in the same country), meaning that some numbers require a second reading to make them fit together in a coherent way.

Overall, our study included papers written by 231 distinct authors (see Table 1). The number of distinct au-

Country	# studies	# authors	# distinct authors	Research density index	# distinct collaborators	Collaboration index
USA	16	40	39	9.3	8	8.1%
Canada	10	30	24	5.3	17	17.2%
Sweden	7	19	17	2.5	10	10.1%
Brazil	6	21	21	23.8	3	3.0%
Italy	6	16	15	7.6	12	12.1%
Germany	6	16	14	3.2	4	4.0%
Finland	5	14	14	2.0	8	8.1%
Australia	4	15	7	1.5	6	6.1%
Estonia	4	5	2	0.6	1	1.0%
Netherlands	3	15	15	3.3	0	0.0%
Spain	3	9	9	3.4	5	5.1%
Singapore	3	9	8	1.2	8	8.1%
China	3	8	8	7.3	6	6.1%
Switzerland	3	6	6	1.4	3	3.0%
Poland	3	5	5	2.4	2	2.0%
Hungary	2	7	7	2.6	0	0.0%
New Zealand	2	4	4	1.0	1	1.0%
Chile	1	3	3	7.0	0	0.0%
Norway	1	3	3	0.5	0	0.0%
Peru	1	3	3	-	0	0.0%
Belgium	1	2	2	0.4	2	2.0%
Colombia	1	2	2	35.1	0	0.0%
Ireland	1	2	2	0.4	2	2.0%
Turkey	1	1	1	0.9	1	1.0%
Total	93	255	231	N/A	99	100%

Table 1: Distribution of studies over authors and countries

thors is lower to the accumulated number of authors as it counts only a single appearance of each author per country (there are 31 authors co-authoring more than one paper). Some countries present a significant difference between accumulated and distinct authors, such as Canada and remarkably Australia and Estonia. On the contrary, only one author co-authored more than one study (in fact, only two) in the USA. For each country, we included a custom metric, i.e. the research density index. The index is defined as follows: the number of distinct authors per each country is multiplied by 1000 and divided by the number of researchers per 1 million inhabitants for each analyzed country (in full-time equivalent, FTE) provided by the UNESCO Institute for Statistics [40]. This research density figure reflects the ratio between the actual number of authors publishing in our research domain versus the general proportion of researchers to the total population of a given country. For instance, if the research density index is equal to 1.0, as in the case of New Zealand, it means that the share of researchers who published papers about BDA for ASD is equal to the expected number of researchers for that country. In other words, there is neither under- or over-representation of researchers in that particular field for the analyzed country. At the other end of the spectrum lie South American countries, such as Colombia (35.1) or Brazil (23.8), where we observe a remarkable magnitude of research output in that particular field. Furthermore, the three Nordic countries: Finland, Sweden, and Denmark, followed by Norway, have one of the highest numbers of researchers per 1 million inhabitants in the world. Nevertheless, as seen in Table 1, they have quite different research density indexes for our research topic.

The numbers in the distinct collaborators' column represent the total number of distinct scholars from the country that co-authored a paper written with researchers affiliated with foreign institutions. We defined the multinational collaboration index for each country as the percentage of the number of distinct international collaborators from that country over the total number of all distinct collaborating authors.

Primary studies from Asia, which account for only 6.6% of authors, were spread evenly over two countries: China and Singapore (with one author having double affiliation in both of these countries), and one paper was co-authored by an academic affiliated to a Turkish university. Population-wise, we could have expected Asia to have more of an impact. As apart from the Nordic countries and Israel, countries such as Japan, South Korea, or Singapore are among the world's leaders in terms of the number of researchers per population of 1 million [40].

Further, we analyzed the collaboration patterns between different countries. It was motivated by the growing importance of international collaboration in modern science [48, 51]. Sugimoto et al. [48] and Wagner and Jonkers [51] found that a country's scientific influence and the quality of its research contributions are correlated with the mobility of its researchers. Figure 4 shows connections between authors' countries. Only

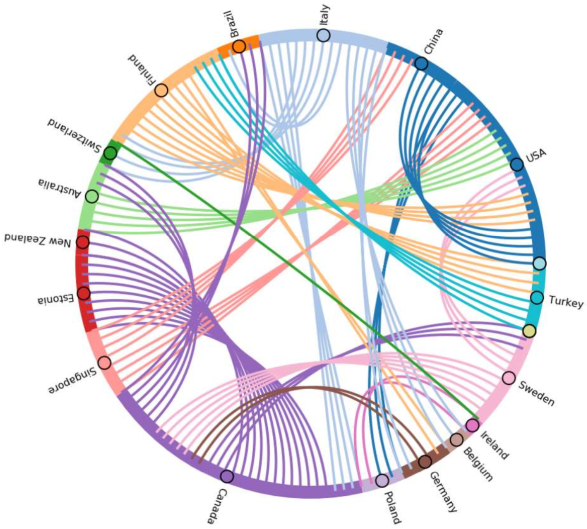
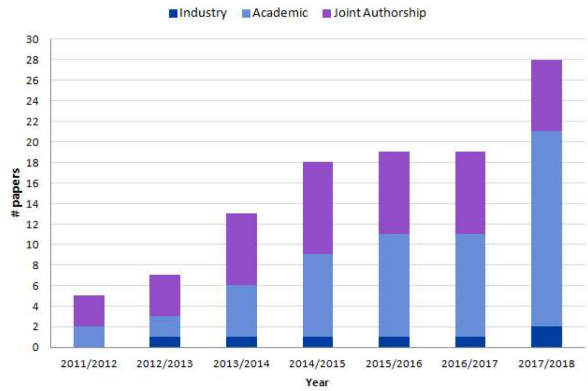


Figure 4: Papers co-authored by authors representing different countries

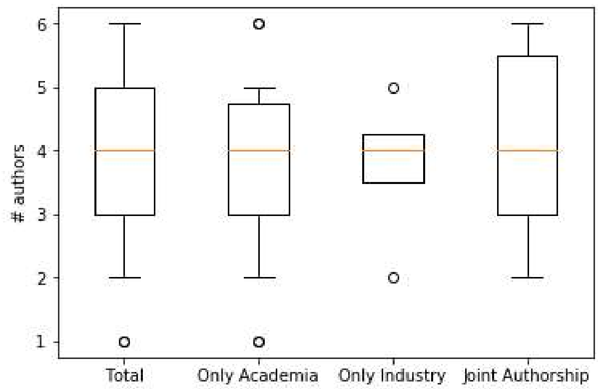
countries that collaborated with other countries are visualized in the diagram. The color of the lines shows the direction of the collaboration, i.e. links have the color of the countries from which authors initiated the collaboration (i.e. the first author and usually his/her team from the same organization). Subsequently, the number of links represents the number of scholars writing the paper from that country. In our SM study, two North American countries proved to be the top international collaborators. However, their collaboration patterns were different. Concretely, the US mainly attracted researchers from other countries and did not initiate international collaborations on its own. Whereas, Canada, more so than other countries in our study, started international collaborations with other partners. Other frequent proactive collaborators included countries from Northern Europe such as Finland and Sweden, which is similar to the findings of Sugimoto et al. [48].

4.1.4. *Authors' affiliation (RQ1.4)*

Figure 5a shows the distribution of studies with respect to the status of employment of their authors: academia (university or research institute), industry (company, research lab, or public administration) or both i.e. joint authorship. 36 (55.4%) papers were published by authors from academia, while 25 (38.5%) publications were a result of joint authorship. In total, only 4 studies (6.2%) originated purely from the industry. Furthermore, 5 authors representing more than one entity had both academic and industry affiliation [S9,S34,S54,S55]. In sum, as shown in Figure 5a using a two-year window, joint collaborations and papers originating solely from industry remained more or less at the same level throughout the years, whereas studies from academia were increasing every year, with an exception for 2016 (the decline in 2016 is synchronized with the overall decrease in the number of papers published that year). Moreover, in Figure 5b we also presented summary statistics – such as the median observation, the lower (Q1) and upper (Q3) quartile – for the number of authors per each paper. On average, a paper was written by 4 authors, irrespective of their affiliation. Articles written as joint authorship were more often authored by more than 4 researchers.



(a) number of papers per affiliation type and 2-year window



(b) number of authors per affiliation type

Figure 5: Distribution of studies w.r.t. authors' affiliation

We also analyzed how research on the topic spread across different countries and institutions. The graphical representation of this analysis is shown in Figure 6, where numbers next to countries, regions, and institutions indicate the number of authors affiliated with each one of them. It appears that Europe overall leads when it comes to conducting research in the industrial context,

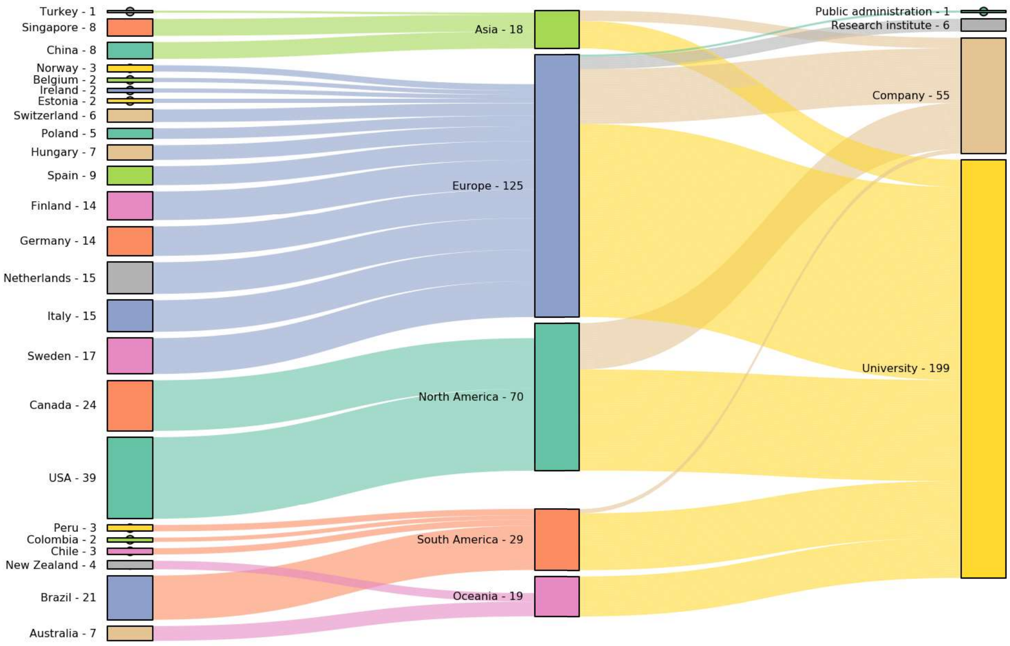


Figure 6: Distribution of studies w.r.t. authors' affiliation

having 22 papers (33.8%) written by one or more authors with industrial affiliations, out of which 11 studies (16.9%) were carried out entirely by European teams. Further, despite a largely fragmented research landscape in Europe, the continent was strongly represented by Sweden, whose researchers participated in 6 studies (9.2%) undertaken with industrial partners co-authoring the papers. Moreover, with respect to studies performed with industry, Europe was closely followed by North American countries: the USA having 9 such studies (13.8%) and Canada 4 studies (6.2%). For other countries, regardless of the region they belonged to, the number of research papers co-authored by researchers with industrial affiliations did not exceed 2 studies. In addition, all 4 papers (6.2%) which originated entirely from industry were mostly written by authors affiliated to the US offices of the companies they represented (14 out of 15), again largely American (Microsoft, Google, Fannie Mae). Only one of those papers, i.e. [S5], featured an international collaborator (in this case from Poland). Interestingly, the paper was written by the authors affiliated to the Swedish-Swiss corporation ABB, hence not an American company. Furthermore, it is also worth mentioning that not all papers discussing industrial case studies had industry practitioners or researchers with industrial affiliations among their authors. Hence, some of the selected primary studies, even though set in the industrial context, were not covered in this section.

4.1.5. Companies (RQ1.5)

Our literature search also allowed us to draw a conclusion with regard to the extent of Agile research undertaken in different companies. In total, 18 companies, 1 research lab and 1 government-related entity were explicitly mentioned by name as the subjects of the selected primary studies. Not surprisingly, some of the authors of the selected studies were affiliated with the companies presented in the papers; however, some of the papers discussed industrial case studies without having any employee of those companies among their authors (e.g. [S31,S26,S59]). Although no dominant companies were identified in our study, the most frequently covered organization was Microsoft, followed by Mozilla with its numerous projects. Further, only two other companies were featured more than once, i.e. Ericsson, Google. These four companies accounted for 16.9% of all studies (see Table 2). Moreover, 13 (20.0%) of the selected primary studies covered anonymous companies, i.e. their names were not disclosed, but in most of the cases we were able to extract information such as their size, core business, or country of origin. The relatively high number of anonymous com-

Company	Company size/type	Company domain	Country	Relevant studies
Microsoft	Corporation	Consumer and enterprise software	Global (USA, China)	S14,S20,S24,S65
Ericsson	Corporation	Telecommunications infrastructure	Global	S46,S60
Google	Corporation	Consumer and enterprise software	Global	S23,S52
ABB	Corporation	Industrial technology	Global (USA, Poland)	S5
AENSys Informatics	SME	Home IoT and IT security management		S6
Amisoft	SME	Software development services	Chile	S50
Avalia Systems	SME	Software development services	Intl. (Switzerland)	S37
Ericpol	SME	IT outsourcing and consulting services	Intl. (Poland)	S25
Fannie Mae	GSE[a]	Financial services	USA	S58
F-Secure	Corporation	Cyber security and privacy solutions	Global (Finland)	S32
Grupo Saberes	SME	Software development services	Colombia	S28
IBM	Corporation	Enterprise software	Global	S22
ING	Corporation	Financial services	Global (Netherlands)	S34
Mozilla	Corporation	Free software community		S21,S31,S59
Multilogic	SME	Software development services	Hungary	S61
Pason Systems	Corporation	Drilling data management systems	Canada	S53
Plexina	SME	Healthcare IT solutions	Canada	S36
Salesforce	Corporation	Marketing and sales software	Global (USA)	S11
Virtus	Not disclosed	Software development services	Brazil	S26
VTT	Research lab	N/A	Finland	S10
Other	Government agency	N/A		S9,S54
Not disclosed	Not disclosed	Not disclosed	Intl. (Finland, Spain)	S42
Not disclosed	Large company	N/A	Italy	S1
Not disclosed	Large company	Consumer and business security solutions	Finland	S27
Not disclosed	Large company	Subscription-based television services	Italy	S29
Not disclosed	SME	On-line business entertaining platforms	Estonia	S27
Not disclosed	Research lab	N/A	Italy	S1
Not disclosed	Not disclosed	ICT system development	Not disclosed	S44
Not disclosed	Large company	Infrastructure provider company	Not disclosed	S13
Not disclosed	SME	Gaming	Finland	S40
Not disclosed	SME	ICT services	Finland	S40
Not disclosed	Large company	ICT services	Finland	S40
Not disclosed	SME	Sports	Finland	S40
Not disclosed	SME	Software development tools	Finland	S40
Not disclosed	Large company	Security	Finland	S40
Not disclosed	Large company	Telecom	Finland	S40
Not disclosed	SME	Multimedia	Finland	S40
Not disclosed	Not disclosed	Software company	Belgium	S48
Not disclosed	SME	Software development services	USA	S51
Not disclosed	Not disclosed	Not disclosed	N/A	S19,S29,S45,S63
N/A	N/A	N/A	N/A	S2-S4,S7,S8,S12,S15-S18,S30,S33,S35,S38,S39,S41, S43,S47,S49,S55-S57,S62,S64

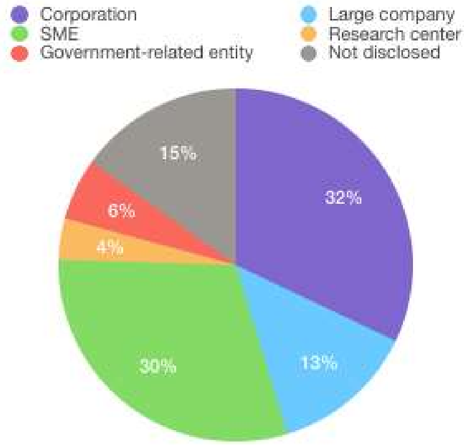
Table 2: Companies discussed in the studies

[a]Government-sponsored enterprise

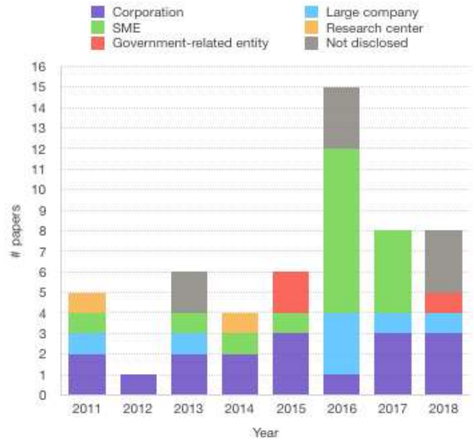
panies may be attributed to the fact that the majority of the primary studies discussing industrial case studies partnered up with more than one company, e.g. Lindgren et al. [S40] in their article covered as many as 10 anonymous software development companies from Finland working in different domains. Authors tend to be consistent with their approach to disclosing company names throughout their papers – if they mention companies, they either reveal the names of all of the covered organizations or none of them. Moreover, our SM study featured also papers without any industrial case study. Those studies, were grouped under a separate category, called *N/A* in Table 2. Furthermore, some of the selected primary studies covered companies with international presence, hence we included this information in Table 2 where possible. Namely, the *Global* label was used to indicate multinational corporations, with countries participating in the selected studies listed in brackets (if available). Similarly, other companies present in more than one country (but not being multinational corporations) were classified as international (*Intl.* abbreviation was used in this case), again with countries participating in the selected studies listed in brackets (if available). It appears that the majority of multinational corporations discussed in the selected primary studies are companies established or having their headquarters in the US. Whereas, large companies and SMEs covered mainly hailed from Europe, with large representation of Finnish companies (due to the study by Lindgren et al. [S40]).

As shown in Figure 7a, small and medium companies (SMEs) and corporations were most frequently studied. However, when grouping organizations by size, the largest group in our study would consist of corporations and large companies, together being the subject of 24 studies (36.9%). In total, only 2 research centers (3.1%) and 3 government-related entities (4.6%) took part in case studies. The small representation of public sector companies poses a question whether in less competitive environments the adoption of Agile methods and data analytics is as prevalent as in the fast-changing business landscape.

Furthermore, most studies were conducted in companies developing software and providing IT services to customers (30.8%). However, other domains, such as financial services, industry (e.g. oil and gas, manufacturing), healthcare, or sport and entertainment industry were also present. This can be perhaps attributed to the growing trend of non-technology organizations becoming technology companies with software developed in-house. However, in sum, the majority of the selected papers studied software development practices and application of BDA in organizations specializing specifically in developing software in a broad context, such as Microsoft, Google, and smaller software houses. Hence, we assume that most of the studies in industry dealt with professional software developers with proven competencies, whose software development practices generalize well.



(a) ratio



(b) per year

Figure 7: Distribution of companies w.r.t. their type

4.1.6. Research results (RQ1.6)

Following Shaw's [47] classification of types of software engineering research results, we categorized our primary studies into seven groups shown in Table 3.

We observe a trend toward proposing new *procedure and techniques*. This can be attributed to a large number of academic papers that produced new approaches providing a lab-based or theoretical validation. However, examples of use in actual practice in the industry were less prevalent. *Solutions, prototypes* and tools together

Category	Relevant studies	# studies
Procedure or technique	S1-S3,S6,S7,S13,S16-S18,S21,S26,S30,S31,S38,S39,S41-S44,S46-S49,S51,S55-S57,S61,S62,S64	30
Tool	S4,S5,S15,S20,S33,S35,S52,S60	8
Qualitative or descriptive model	S8,S24,S32	3
Empirical model	S9,S19,S27,S45,S54	5
Analytic model	S29	1
Solution, prototype or judgment	S10-S12,S14,S22,S23,S25,S28,S36,S53,S63	11
Report	S34,S37,S40,S50,S58,S59,S65	7

Table 3: Selected primary studies mapped according to the research results

Analytics type	Description	Relevant studies
Descriptive	reactive, typically based on historical data, using descriptive statistics to describe past in a summarized form (e.g. visualization, ad-hoc reporting, dashboards)	S4-S6,S9,S13,S15,S24,S26,S27,S30,S35-S37,S43,S46,S51,S52,S59,S60,S63
Predictive	reactive, typically based on historical data and using AI/ML techniques, this type of analytics uncovers hidden patterns and makes predictions about the future (forecasting) (e.g. fraud detection, sentiment analysis)	S1-S3,S7,S10-S12,S14,S16-S23,S25,S28,S29,S31,S33,S34,S36,S38,S39,S41,S43-S45,S47-S50,S54,S57,S58,S62,S64,S65
Prescriptive	reactive and proactive, typically based on historical data recommends actions that can be undertaken by decision makers along with their implications (e.g. simulations)	S8,S53,S42,S55,S56,S61,S62
Adaptive	proactive, historical and real-time data, by interacting with an environment automatically adapts to recent actions which influences the present and, in result, improves the ongoing learning process (e.g. reinforcement learning, counterfactual ML, recommender systems)	S56

Table 4: Types of analytics employed in selected primary studies

constituted about a third (19) of all studies. Moreover, only 7 (10.8%) studies reported lessons learned, followed by a small representation of frameworks and guidelines 9 (13.8%). This analysis of the contribution type facets indicates that this is still an evolving research field, which is developing new approaches and lacks evaluated models and theories.

4.2. *Data Analytics throughout the Lifecycle of the Agile Software Development Process (RQ2)*

In this section, we report the results of our analysis of different types of analytics employed in ASD, and we discuss various sources of data used in the selected primary studies as well as we shed light on numerous BDA methods and techniques used in ASD.

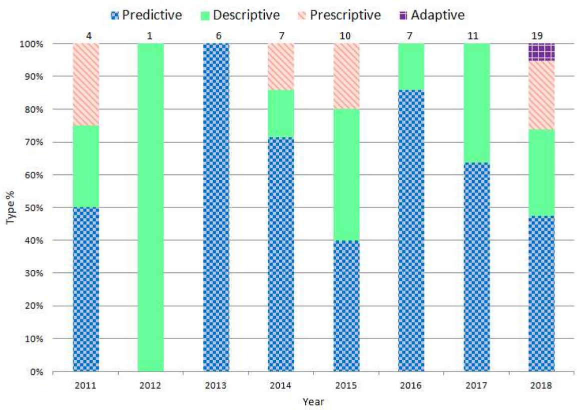


Figure 8: Analytics type

4.2.1. *Analytics types (RQ2.1)*

In order to analyze different analytics types used in ASD, we defined four types of analytics. To this end, we adopted the classification of analytics types as proposed by [20] and extended it with an additional type (i.e. adaptive) to cover the whole spectrum of analytics. Next, we split our papers into four groups, which are discussed in Table 4 and presented in Figure 8.

The majority of studies covered descriptive and predictive analytics. Both of them are primarily based on historical data and do not include more advanced ML concepts and applications such as simulation of different scenarios or reinforcement learning, among others. For instance, descriptive analytics solutions often took the form of dashboards with KPIs or reporting methods providing ASD related metrics (e.g. [S5,S9,S13,S60,S63]). For instance, the solutions proposed in [S60] and [S63] were based on typical ASD artifacts such as the number of features or test cases designed, developed, or integrated in a given period of time. Thanks to a descriptive analysis with a visual component performed in the two studies, the decision-makers in the companies gained a better overview of their ASD processes.

The more advanced type of analytics, the smaller the number of papers discussing it, e.g. prescriptive was less frequently covered in the selected primary studies. With regard to prescriptive analytics, we observed mainly studies focus on optimization [S56,S61] as well as simulation and modeling [S42,S53,S55]. The last type, which is adaptive analytics, was covered only in one primary study, i.e. [S56]. Schermann and Leitner [S56] proposed an optimization solution for experimenting in a continuous deployment environment. Specifically, the authors observed that continuous experimentation at scale requires careful orchestration of experiments, as they often depend on each other and many constraints need to be considered (e.g different user base, the volatility of experiments, code development). Since product releases can be impacted by failed experiments, it is a non-trivial problem, which is resolved using search-based methods in the study [S56]. The prescriptive/adaptive component of the solution allows rescheduling of experiments, which are frequently adjusted to changing user behavior. Some primary studies discussed more than one type of analytics (e.g. [S36,S43,S56,S62]). There were also 2 studies that did not explicitly cover any type of analytics, i.e. [S32,S40], hence were not included in Table 4. The authors of the two papers carried out qualitative studies and, based on case study data (such as surveys, interviews), drew conclusions regarding experimentation in product development and release planning practices. Although the topic of their studies is related to BDA in ASD, the papers themselves cannot be categorized into any of the 4 major groups of analytics. Interestingly, 2018 was the year when the largest number (3) of prescriptive analytics was produced. However, before that, there was a 2-year gap, when no such papers appeared. Moreover, the only paper covering adaptive analytics was also published in 2018. Hence, we could conclude that in 2018 more advanced BDA gained attention in the community. However, it remains to be seen if this trend prevails.

4.2.2. Data sources (RQ2.2)

The quality of insights provided by analytics depends to a large degree on data. In our analysis, we divided data sources into 7 major groups, which are presented in Table 5.

As shown in Table 5, in the software domain, there is no dominant source of information for analytics, and many analytics engines use various data sources to provide a more holistic and comprehensive view of the studied environment. 35.4% of the studies used more than one data source, including data from: issue reports, test results, commit logs, bug repositories, version control systems, among others. One such example is an analytics solution presented in [S5]. The authors highlighted the importance of combining disparate data sources such as the source code repository, defect backlog, personnel data. For instance, the last type of data – in this case coming from Lightweight Directory Access Protocol directory (LDAP) – proved to be helpful in resolving code ownership issues, such as identifying abandoned parts of the codebase due to the departure of staff or role changes. Similarly, Czerwonka et al. [S20] in their study used multiple data sources. Apart from source code repositories, which the authors considered the largest and most volatile sources of engineering data in their organization, they utilized test results, organization, and project execution data (e.g. release schedules and development milestones). Similarly, in [S14] the project repository was utilized specifically because it contained all source code information, such as the number of lines of code (LOC) for a class or dependency information for a method. Project execution data, although often not specified, frequently included sprint data [S7], story points [S29,S48,S54,S61]. For instance, Batarseh et al. [S7] described in detail what data they collected from sprint data: i.e. sprint number, start and failure date, total time, Mean Time between Failures (MTBF). Importantly, some primary studies used synthetic data. For instance, [S4] used simulated Scrum data; also authors of [S42] augmented real data with an artificial one (such as artificial user stories) to feed their simulation model. Data collection, processing, and further loading to an analytics engine often entailed executing complex steps along the way. For instance, Huijgens et al. [S34] mentioned combining data from the deployment registry, the configuration management database, the event monitoring data warehouse, and the deployment orchestration logging. Further, they used timestamp data to identify active configuration items and to extract deployment steps from the logged workflow data. Very rarely studies provided information on data sources which they deliberately did not use, due to high costs, for example. However, in rare cases, such information was revealed; for instance, [S50] describes that a company covered in the paper decided not to monitor neither its version control nor defect tracker systems. In a similar vein, numerous studies stressed the importance of data quality (including taking care of incomplete and inaccurate data) and the need for its verification before rolling out an analytics solution [S5,S9,S49].

When it comes to specific tools and systems providing data, JIRA turned out to be the most popular choice, with 7 (10.8%) studies using it [S2,S3,S16,S17,S18,S36,S44]. Also, various Microsoft products (such as Dynamics Ax, Windows, Windows Phone, Office, Exchange, Lync, MS SQL, Azure, Bing,

Data source type	Detailed data source	Data source system	Relevant studies
User feedback	app reviews	the App Store, Google Play	S15,S30,S33,S41,S43
	controlled experiments		S24
	interviews/surveys		S4,S8,S12,S13,S26,S32,S40,S46,S47,S51
Logs	application usage	e.g. Eclipse IDE, APM & program analysis tools	S6,S8,S35,S39,S43,S51,S56
	test results & test suite execution		S10,S20,S23,S37
	commit logs		S59
	system failures	e.g. Microsoft Dynamics Ax	S14
	failure data		S45
	revision history		S64
	workflow data logs		S34
	event monitoring logs		S34
	deployment orchestration logs		S34
Project artifacts	features		S20,S63
	user stories		S29,S42,S51,S54,S55
	defects		S20
	tasks		S38
	product backlog	e.g. JIRA	S9,S60
	defect backlog		S5,S60
	deployment registry		S34
Issue reports	issue tracking system	e.g. JIRA	S2,S3,S16,S17,S18,S21,S36,S44,S57,S59
	bug repository	e.g. Bugzilla	S31
Source code & data model	source code		S5,S14,S20,S35,S58,S59
	Ruby programs & Ruby on Rails		S28
	Java programs	e.g. Eclipse IDE plug-ins	S1,S4,S6,S11,S27
	function calls		S65
	development repository		S14,S20,S22
	test case	e.g. Microsoft Dynamics Ax	S14
	code quality	e.g. SonarQube	S44
	application data schema		S28
Project execution	sprint data		S7
	story points		S29,S48,S54,S61
	project timeline	e.g. internal Kanban tool	S25,S49,S60
	development effort	e.g. internal Kanban tool	S25,S42,S49,S60
	version control system		S20,S21
Demographic/organization data		e.g. LDAP	S5,S20,S53,S57
Other	meeting minutes		S50
	time sheets		S50
	conversations		S51

Table 5: Summary of data sources and systems used to gather data

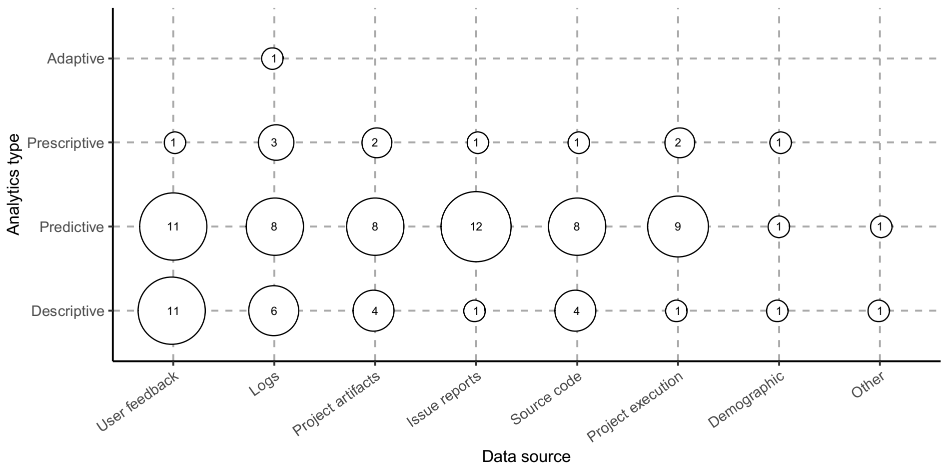


Figure 9: Distribution of data sources presented in the studies w.r.t analytics type

and Xbox) were named in several studies [S14,S20]; however, Microsoft was an industrial partner in those studies. Other software products used as data sources included, for instance, Bugzilla [S31] – a web-based bug tracking and testing tool, or Eclipse IDE (with deployed plug-ins) to capture Java developers' behaviors [S6,S27]. Further, [S11] used testing results, data from the live automation system at Salesforce. With regard to strictly Agile related data and tools, several different approaches unfolded. Some studies relied on popular solutions such as GitLab, web-based DevOps platforms [S56], while others decided to leverage tools developed in-house. For instance, Fitzgerald et al. [S25] used an internally developed, web-based Kanban board tool to record metrics such as: allocated project number, date of project arrival, project development start date, expected and actual finish dates of the project, estimated and actual development effort for the project. [S39] and [S52] utilized tools developed for the purpose of the studies, which are described in detail in section 4.3.3.

For the majority of studies, data originated from software projects, both industrial and open-source (OSS). Certainly, the amount and the variety of data as well as the ability to process it, among other factors, depend on the organization and software development teams, which are the source of data. With regard to the size of industrial projects, the discussed primary studies covered a wide range of them: from small teams with around 10 developers (e.g. [S42]) to large teams, covering as much as 300 different development teams in a large software company [S34]. Also, the number of projects varied. Some studies focused only on 1 project (e.g. [S6,S11,S14,S42,S60]), while others included hundreds of projects – e.g. [S25] studied 467 projects, [S34] covered 750 projects.

With respect to OSS projects, Apache projects dominated, with 7 (10.8%) studies using their data [S4,S16-S18,S48,S57,S64]. Other frequently utilized OSS projects included Usergrid, Appcelerator Studio, Aptana Studio, Spring XD, Titanium SDK – each one of them was employed in three studies (e.g. [S18,S48,S57]). Furthermore, benchmark datasets such as the ISBSG dataset were also used by some studies [S49]. In terms of the association of data sources used in four types of analytics, Figure 9 depicts this relation. Especially *issue reports* and *project execution* data (and to a lesser degree *project artifacts*) prove to be used more frequently for predictive analytics than for descriptive analytics. For prescriptive analytics, the usage of different data sources is more balanced, with the largest number of studies (3) using *logs*. Similarly, a single study using adaptive analytics utilized *logs* as a data source.

4.2.3. Methods, models and techniques (RQ2.3)

In this section, we discuss ML/AI methods, models, and techniques employed in the selected primary studies. A summarized version is presented in Table 6.

Application areas for BDA in ASD are as varied as a testing improvement to feedback elicitation. All of those are underpinned by a large variety of different ML models. Several studies covered more than one type of analytics (i.e. [S36,S43,S56,S62]). In some works, the application of classical ML algorithms was contrasted with more modern approaches such as Deep Learning (DL) (e.g. [S16,S18]).

Figure 10 outlines different ML/AI methods used in the selected primary studies and, if applicable, their performance in comparison to other techniques. Studies reporting more than one type of analytics are represented as a combination of shapes (e.g. [S36]: descriptive and predictive analytics, [S56]: prescriptive and adaptive analytics). Some papers discussed more than one method, and if they were compared, it is marked by arrows in Figure 10. Specifically, a method achieving the best results for a particular study is indicated by an arrow, pointing from the worse performing models in the study to the best one. For instance, in [S2] various models were deployed, such as kNN, different decision trees, or ensemble methods.

Therefore, overall, groups containing the largest number of arrowheads yielded the best results. Among the techniques, ensemble-based classifiers performed significantly better than regular classifiers, where Random Forest performed best in several studies (e.g. [S2,S16,S22]). Further, although neural networks (NNs) often yielded good results as individual classifiers (e.g. [S30]), they were outperformed by simpler methods such as SVM on several occasions (e.g. [S1,S49]). Interestingly, SVMs were used both as classifiers (e.g. [S36,S51]) and regression models (e.g. [S1,S17]). Overall, SVM was the most popular technique used in 11 (16.9%) studies, followed by Naive Bayes employed in 8 (12.3%) studies.

Notwithstanding that many studies discussed ML methods or techniques applied, some papers provided a more coarse-grained description of implemented BDA systems without delving into the details or types of methods used, hence they were not included in Table 6. One such example is [S65], where authors merely scratched the surface of used ML models by reporting only that classification and clustering were performed over execution traces. Another such example is [S33]. Along the same lines, Pareto et al. [S46] presented a

Category	Learning/Model type	Method	Relevant studies
Instance-based	Discriminative	Support Vector Machine (SVM)	S1,S2,S11,S17,S22,S30,S36,S48 S49,S51,S57
		k-Nearest Neighbor (kNN)	S2,S31,S36,S48
Decision Trees	Discriminative	Decision Tree (DT)	S48
		Classification and Regression Tree (CART)	S2
		C4.5	S2,S16,S41
Regression	Discriminative	Logistic Regression	S21,S30,S51
		Linear Regression	S1,S7,S27
		Exponential Regression	S7
		Generalized Linear Model (GLM)	S49
Clustering	Connectivity	Hierarchical Agglomerative Clustering (HAC)	S14
	Centroid	K-Means	S36
Ensemble	Boosting	AdaBoost	S2
	Bagging & Random Forests	Random Forest (RF)	S2,S3,S16,S17,S18,S22
		Bagged AdaBoost (Adabag/BAB)	S2
		Stochastic Gradient Boosting (SGB)	S2,S16,S17
		Bagging	S41
Neural Networks	Shallow Neural Networks	Neural Network (e.g. Multi-Layer Perceptron)	S1,S16,S18,S30,S49
	Deep Learning	Deep Neural Network	S16,S17,S18
Probabilistic Graphical Models	Generative	Naive Bayes	S2,S16,S22,S26,S30,S41,S44,S48
		Hidden Markov Hodel (HMM)	S31
		Expectation Maximization for Naive Bayes (EMNB)	S15
		Directed Acyclic Graph (DAG)	S55
		Stochastic Automata Network (SAN)	S19
		Social Network Analysis (SNA)	S64
Optimization	Combinatorial optimization	Multiple knapsack	S61
		Branch-and-bound	S61
		Branch-and-cut	S29
		Simulated Annealing (SA)	S56
		Genetic Algorithms (GA)	S56
	Distributed constraint optimization	*Custom algorithm (SMART)*	S38

Table 6: Classification of ML/AI methods and techniques

mixed method combining qualitative, quantitative, and analytical techniques to improve architecture documentation, without providing much detail on the used techniques. Another paper was [S5] in which authors reported on lessons learned from implementing and designing dashboards to visualize metrics and KPIs. Visualization was strongly represented by [S60,S63]. In a similar vein, Maalej and colleagues [S43] analyzed how analytics can support requirements analysts' decisions through mining explicit and implicit user feedback. The scholars only listed ML areas or tasks such as natural language processing, sentiment analysis, or topic-based summarization (e.g. LDA) without providing further explanation of methods or techniques. In the same vein [S15] discussed topic modeling i.e., LDA and Aspect and Sentiment Unification Model (ASUM). Similarly, [S12] not described in detail algorithms used besides mentioning only machine translation as an ML topic addressed by the paper. Another journal article with a low level of supporting detail concerning ML techniques underpinning its data analytics solution was [S58]. The paper described a BDA solution giving a bird's-eye view to a company of the code base through automated measures of size and structural quality. [S52] presented an approach for static analysis.

In general, studies published in IEEE Software provided higher-level of abstraction, without discussing particular ML algorithms used (e.g. [S5,S20,S33,S35,S42,S43,S50,S58,S59,S62,S64]). Several papers used their own, custom-designed methods. For instance, [S45] proposed a model with domain-specific heuristics for the prioritization of test cases. [S47] introduced a method called *Incremental Cosmic Function Points* (CFP) enabling estimating effort for each project increment. [S25] proposed a model for evidence-based decision-making in lean software development processes. Further, [S53] harnessed *System Dynamics* (SD) modeling technique to create a tool used to simulate various configurations of test processes in order to support test automation. Similarly, Elbaum et al. [S23] proposed two algorithms improving the cost-effectiveness of continuous integration processes. Batarseh and Gonzalez [S7] proposed a forecasting regression model combined with the analysis of Mean Time Between Failures (MTBF). In [S28], a model for automatically generated diagrams was demonstrated.

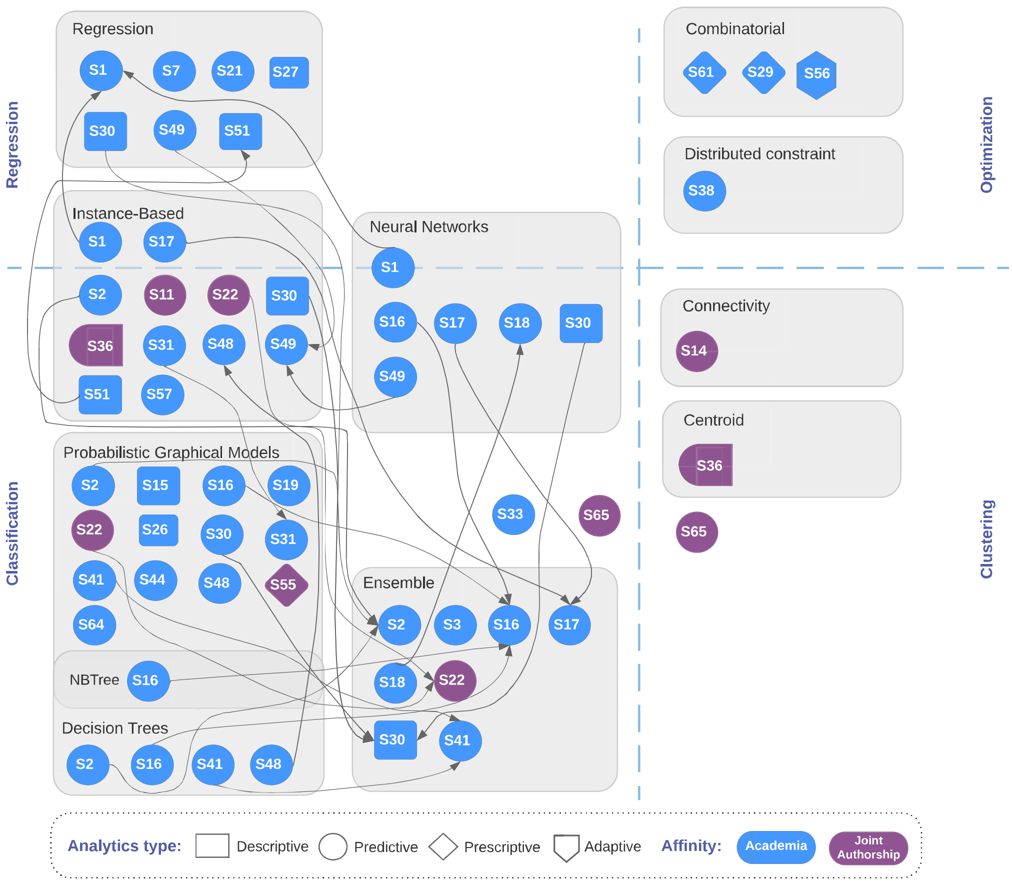


Figure 10: Classification of ML/AI algorithms w.r.t to their performance and model categories. Pointers (i.e. arrows' pointed ends) indicate algorithms with better performance if such comparison was reported in the selected primary studies. Studies having more than one type of analytics are depicted as a combination of appropriate shapes.

Whereas [S5,S9,S34,S50] developed BDA solutions which relied on various metrics and indicators to assist decision-makers with creating goals, not necessarily employing any kind of AI/ML algorithms.

Other studies focused on descriptive or Exploratory Data Analysis (EDA) [S39], which are not ML/AI methods, per se. Similarly, two studies (i.e. [S32,S40]) conducted qualitative studies and semi-structured individual interviews with industry practitioners. The papers, although reported on BDA in ASD, did not use themselves any type of analytics to enhance ASD processes. As shown in Figure 10, the majority of papers that contributed to understanding what ML techniques are used in ASD originated from academia, followed by joint authorship papers. In contrast, no industry-led research explicitly mentioned ML techniques used; hence no such papers appeared in the figure. Most studies which disclosed techniques and models used, did not provide information on the set parameters and hyperparameters.

Figure 11 provides information on types of data sources used by specific ML techniques. Instance-based family of methods was by far the most frequently used set of algorithms and it used predominantly *project artifacts* and *project execution* data. Importantly, *project artifacts* were utilized only by 3 types of techniques, with a significant disproportion in the number of studies using this type of data. Further, clustering algorithms, in principle, sourced data from *logs*. The majority of *user feedback* data was analyzed by instance-based, decision trees, and regression techniques. Optimization algorithms did not show any clear preference towards types of data used, while ensemble methods preferred *issue reports* over others.

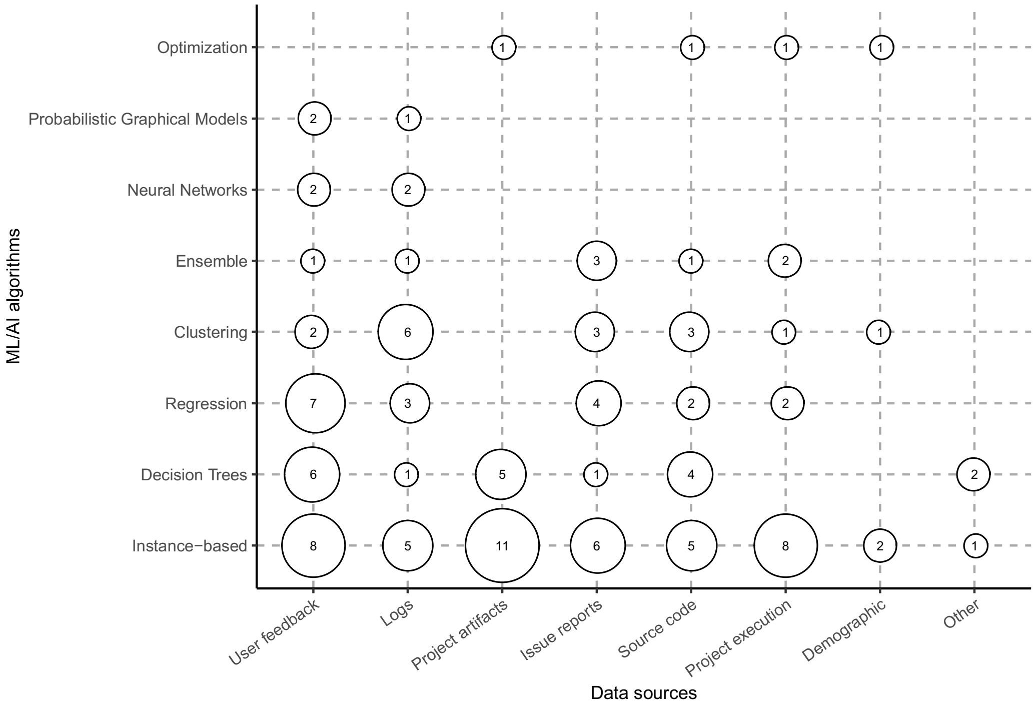


Figure 11: Distribution of data sources presented in the studies w.r.t ML/AI algorithms

Area	Sub-area	Relevant studies	# studies
7.2 Software Project Planning and Cost Estimation	7.2.3. Effort, Schedule	S1,S4-S6,S17-S19,S29,S36,S38,S39,S42 S47-S50,S54,S57,S58,S61	20
10.1 Software Quality Fundamentals	10.1.4. Software Quality Improvement	S7,S13,S15,S16,S21,S24,S30,S31,S35,S40,S52	11
1.4. Requirements Analysis	1.4.4 Requirements Negotiation	S1,S12,S22,S28,S32,S41,S43,S50,S55,S63	10
4.5 Test Process	4.5.2. Test Activities	S11,S14,S23,S33,S37,S45,S53,S56	8
7.6 Software Engineering Measurement		S9,S12,S13,S34,S35,S44,S60,S62	8
7.2 Software Project Planning and Cost Estimation	7.2.4. Resource Allocation	S5,S19,S25,S38,S58	5
6.6 Software Release Management and Delivery	6.6.2. Software Release Management	S2,S3,S21,S59,S61	5
7.4. Review and Evaluation	7.4.2 Reviewing and Evaluating Performance	S9,S44,S60,S64	4
11.2 Group Dynamics and Psychology	11.2.1 Dynamics of Working in Teams/Groups	S26,S42	2
2.2 Objectives of Testing	2.2.13 Usability and Human Computer Interaction Testing	S33,S56	2
1.3. Requirements Elicitation	1.3.2. Elicitation Techniques	S51,S63	2
3.4. Construction Technologies	3.4.16. Test-First Programming	S8,S27	2
1.4. Requirements Analysis		S4	1
2.3. Software Structure and Architecture	2.3.4 Architecture Design Decisions	S46	1
4.6 Software Testing Tools	4.6.2. Categories of Tools	S10	1
7.2 Software Project Planning and Cost Estimation	7.2.5. Risk Management	S16	1
8.3 Software Process Assessment and Improvement		S20	1
10.3 Practical Considerations	10.3.1. Software Quality Requirements	S65	1
11.3 Communication Skills	11.3.3 Team and Group Communication	S26	1

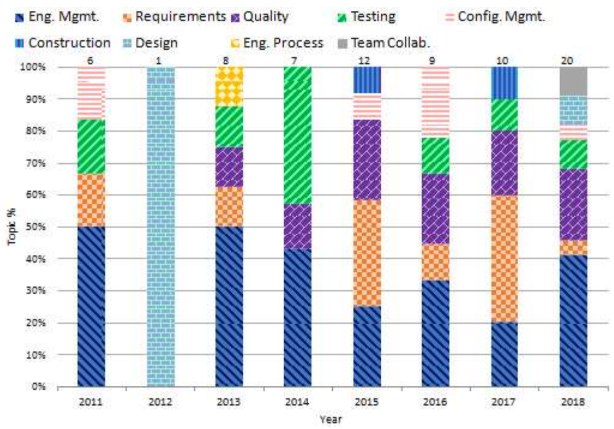
Table 7: A thematic classification of studies based on SWEBOK knowledge areas

4.3. How Data Analytics Improves Agile Software Development Process (RQ3)

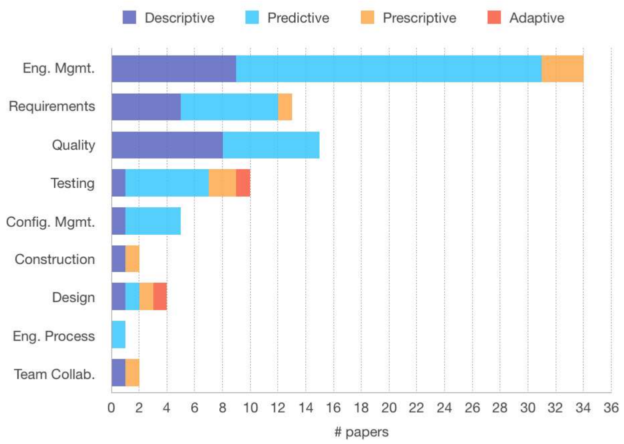
4.3.1. Research topics and sub-topics (RQ3.1)

Broad knowledge areas, such as requirements engineering, software design, development, testing, or project management, were narrowed down using the SWEBOK thematic classification (see Table 7).

BDA is applicable to almost all phases of the ASD process. 16 (24.6%) studies covered more than one area. Software engineering management topics such as project planning and effort estimation turned out to be the most popular among the authors of our primary studies - 32 (49.2%) studies covered them in total. Requirements analysis and elicitation also proved to be important, we identified 13 (20%) such studies. The year 2018 was the most diverse in terms of the SWEBOK areas covered - 7 different areas out of 9 (see Figure 12a).



(a) per year



(b) per analytics types

Figure 12: A thematic classification of studies w.r.t SWEBOK knowledge areas

Our study also reveals that software testing was one of the major applications and most diverse when it comes to different analytics types, as it covered all four of them (see Figure 12b). Issue resolution and detection of defects remain one of the primary domains of research in BDA for ASD, 14 (21.5%) studies discussed those topics. The selected prescriptive analytics studies focused mainly on test activities [S8,S53,S56], effort estimation, and delivery capability [S29,S42,S55].

Not only different types of analytics use different data sources, but also different phases of the ASD process require different sets of information on input. In Figure 13, studies focused on software engineering management activities (as the only SWEBOK group) utilized the whole spectrum of data sources with majority of works using *project artifacts*, followed by *user feedback* and *project execution* data. Software requirements used three main data sources: *user feedback*, *project artifacts* and *source code*, with a slight preference towards *user feedback*. On the other hand, testing and software quality had a clear inclination toward a single type of data source: *logs* and *user feedback*, respectively.

4.3.2. Agile practices, techniques or engineering practices (RQ3.2)

Our classification of Agile practices/methodologies, presented in Table 8, is similar to VersionOne's Annual State of Agile Report [7]. As far as it is important to discuss Agile practices that were discussed in the selected primary studies, it is no less interesting to focus on those practices that were not mentioned there. Interestingly, methodologies such as DSDM/Atern did not appear in any of the selected studies. Customized lean methodologies were mentioned in 2 papers: [S60,S46]. Whereas the DevOps approach was applied in projects from 2 studies: [S34,S58].

Despite the fact that majority of studies are in the category "Other - Not Specified", we can assume they use Agile-inspired methodologies, as papers discussing them refer to e.g. Planning Poker (as the method for requirements prioritization) [S12], project sprints taking two weeks each [S9], Test-Driven Development (TDD) [S27], or user stories and story points [S55]. Also, a number of studies in this category discuss OSS Apache Software Foundation projects, which also employ Agile-inspired software development processes – they are self-governing and egalitarian by using a collaborative, consensus-based process [1]. For instance, although RUP is not ASD per se, it encompasses many Agile-inspired techniques. Therefore, similar to other authors of literature reviews on Agile, e.g. [19], we also included those practices in our study.

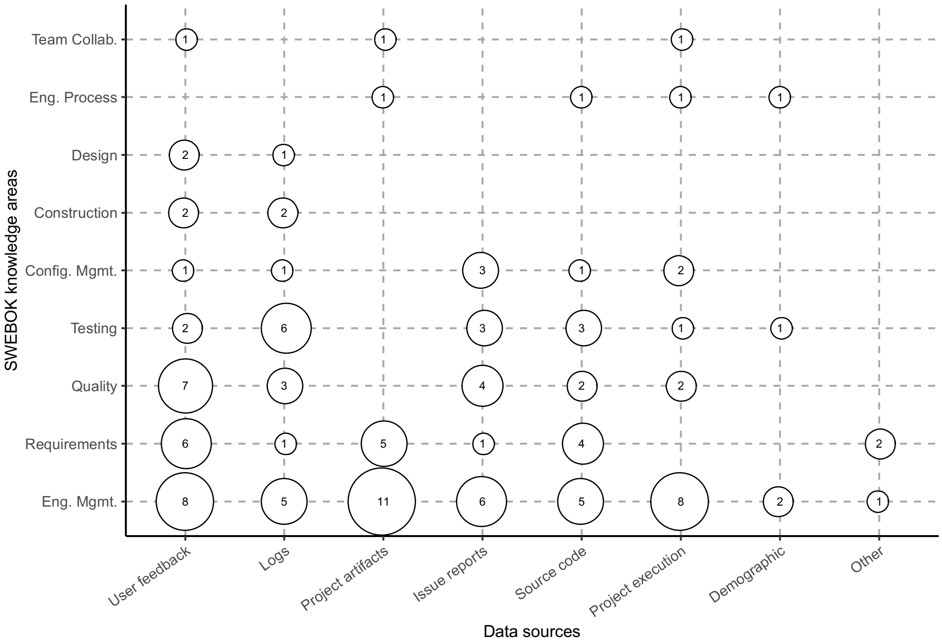


Figure 13: Distribution of data sources presented in the studies w.r.t SWEBOK knowledge areas

Category	Relevant studies	# studies
	Agile practices	
Scrum	S4,S5,S26,S28,S29,S32,S34,S36,S38,S39,S42,S53	12
Kanban	S25,S42	2
Custom Extreme Programming (XP)	S1,S37,S48,S63	4
Custom Hybrid (multiple methodologies)	S40,S46,S58	3
Other - Rational Unified Process (RUP)	S46,S47,S50	3
Other - Not Specified	S2,S3,S6-S12,S14-S20,S22-S24,S27,S30,S41,S43,S45,S47,S49-S52,S55,S59,S60,S61,S54,S65	32
	Agile engineering practices	
Refactoring	S1,S26	2
Pair programming	S1,S37,S48,S63	4
Model-Driven Development (MDD)	S46	1
Test-Driven development (TDD)	S1,S8,S27	3
Behavior-driven development (BDD)	S33,S37,S58	3
Automated acceptance testing	S59	1
Continuous integration	S11,S23,S13,S45,S40,S62	6
Continuous delivery	S13,S34,S35,S62	4
Continuous deployment	S13,S56,S62	3
Continuous experimentation	S56	1
	Agile techniques	
Story mapping	S1,S63,S28,S18,S22,S26,S42,S51,S54,S57	10
Iteration/sprint planning	S1,S9,S7,S26,S28,S29,S36,S38,S39,S42,S48,S53,S63	13
Release planning	S22,S26,S29,S32,S42,S56,S59,S61,S63	9
Team-based estimation	S1,S12,S36,S38,S39,S48,S54	7
Retrospectives	S32,S42	2
Agile/Lean UX	S40	1
Standups	S51	1

Table 8: Selected primary studies mapped according to Agile practices, techniques, and engineering practices

Agile techniques and engineering practices are listed in Table 8. Not all selected primary studies shed light on details of embraced Agile practices or techniques. For instance, [S65] advocated the iterative Agile process with quick feedback loops. As Zhang et al. [S65] stressed, close collaboration between researchers and practitioners fosters continuous development and improvement. Nevertheless, particular Agile practices em-

Tool name	Description	Adoption	Relevant study
ScrumCity Conceptual Visualisation	Visualisation linking code with functional requirements and development activity	Partial	S4
Qlik Sense Metrics Portal (QSMP)	Dashboard with KPIs visualizing metrics	Yes	S5
Besouro	Automatic TDD behavior evaluation system	Partial	S8
eConferenceMT	Automatic translation of text messages	No	S12
AR-Miner	Mobile app review mining platform	No	S15
CODEMINE	Platform for collecting and analyzing engineering process data	Yes	S20
AppFlow	Synthesizing highly robust, highly reusable UI tests	Partial	S33
Crumb Management Platform	Feature increment assessment with additional knowledge sources	No	S35
Probe Dock	Test analytics platform	Partial	S37
Human-centred Agile Software Engineering (HASE)	Online Agile project management (APM) tool	No	S39
ROCKET	Tool automating test selection and scheduling in continuous integration	Partial	S45
Unnamed	Platform for analyzing project scheduling and delivery capabilities	Yes	S50
Tricoder	Program analysis platform	Yes	S52
Unnamed	Simulator for evaluating degrees of test automation	Partial	S53
Application Intelligence Platform (AIP)	Structural-quality analytics	Yes	S58
Unnamed	Measurement system for monitoring bottlenecks	Partial	S60
Feature Survival Charts+ (FSC+)	Requirements scope tracking tool	Partial	S63
StackMine	Postmortem performance debugging system	Yes	S65

Table 9: Tools discussed in the primary studies

ployed remained to be disclosed.

4.3.3. Tools (RQ3.3)

After reporting on software analytics applications, this section is devoted to a discussion regarding tools and models that were designed and developed by the authors of the selected primary studies. In this section, we try to understand how pervasive is the transfer of research outcomes into practice and the state of industrial adoption of BDA solutions. We created three groups with regard to the degree of tools' adoption:
i) No practical application - a tool with no immediate application. Experiments were performed based on publicly available data (e.g. Apache repositories) or in an academic setting.
ii) Partial application - numerous studies conducted a case study at a company to develop a model or a tool together with stakeholders from a company. Such proofs-of-concept or pilots were only partially introduced in the company.
iii) Practical application - a tool with practical application. Most often, an internal team within a company developed a tool that supported the organization's software development efforts in some way.

In Table 9, tools with *Practical application* are denoted by *Yes* in the *Adoption column*, tools with *no practical application* as *No*, while *partial application* as *Partial*. Tools having their roots in the industry proved to be widely exploited research results by practitioners (e.g. [S5,S20,S25,S52]). Furthermore, a substantial number of the tools adopted in the industry were featured in the papers that reported on their usage through experience reports. The majority of them (e.g. [S5,S20,S50,S58]) were covered in IEEE Software journal which, in contrast with other publishing forums, particularly encourages contributions that cater to practitioners. Some tools included in our paper were developed for specific case studies and their adaptation to other scenarios and organizations would require conducting further validations as well as substantial effort (e.g. [S39,S45,S53]). Furthermore, the dissemination of research results originating from academia was not prevalent. Although authors of such papers often claimed that their solutions could provide practical value, however, they indicated that further experiments on real-world data sets and empirical studies to validate industrial applicability of the techniques would be needed. For instance, HASE was only tested in an academic setting (i.e. a group of undergraduate students). However, the authors suggested that their goal is to design a situation-aware decision support system that would help global ASD teams to allocate tasks more efficiently. Similarly, the authors of the feature crumb management platform [S35] provided only a reference implementation, which was validated in a university capstone project.

In terms of the kind of tools described in the studies, many works incorporated additional functionalities to existing tools or integrated with solutions already in use by the studied organizations. For instance, Calefato et al. [S12] augmented the eConference software, introduced in [18], with the machine translation plug-in named eConferenceMT. Marijan et al. [S45] proposed an approach for test case prioritization for continuous regression testing, which was implemented within a tool developed by the studied organization. Further, Augustine et al. [S5] used a third-party business intelligence (BI) platform, Qlik Sense, to build an in-house analytics solution called the Qlik Sense Metrics Portal (QSMP).

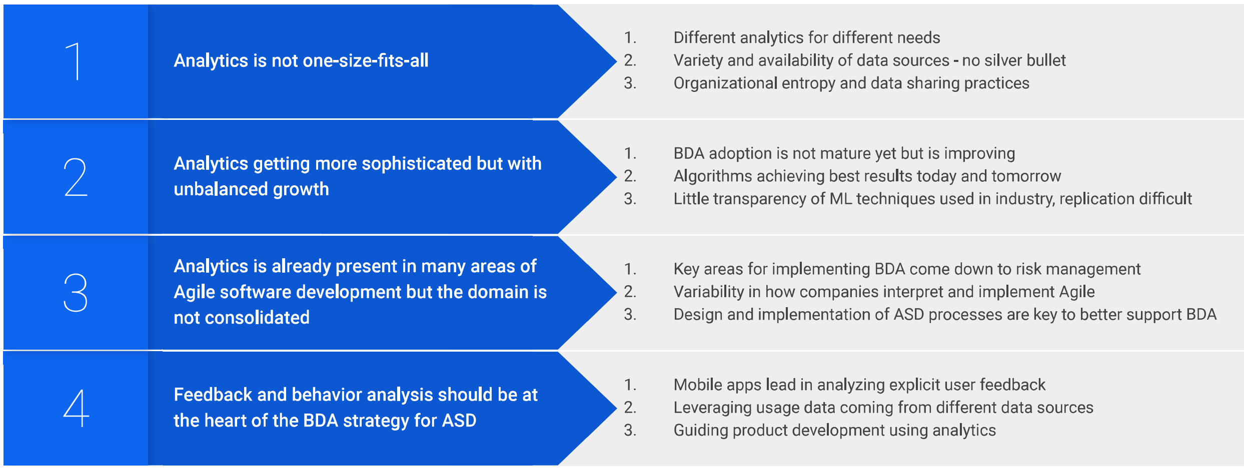


Figure 14: Key takeaways and observations of our mapping study

The tool was used to visualize data, provide KPIs and other important team-related metrics. Thanks to this solution, the company's software development teams were provided with insights not available through an application lifecycle management (ALM) tool, which they normally used. In a similar vein, teams at Fannie Mae IT [S58] complemented insights from the pre-build static-analysis tool SonarQube with an internally developed solution, AIP. Snyder and Curtis [S58] stressed the importance of leveraging existing platforms where possible and adding new solutions only when necessary. Similarly, the CODEMINE data platform [S20], an internal Microsoft tool used by hundreds of users across all major Microsoft product groups, is integrated with a test prioritization, failure and change risk prediction tool called CRANE. CODEMINE supplied CRANE with data related to source code, features, defects, and people from different Microsoft's product teams.

However, not all tools were used only as in-house solutions. Some were made available for use by the public. For instance, Besouro's [S8] source code, which is an Eclipse plug-in, is available on GitHub[1]. Likewise Probe Dock [S37], a web-based test analytics platform, is an open-source tool[2]. AppFlow [S33] is available for Android applications. Although Chen et al. [S15] did not provide access to their tool (AR-Miner) the authors made their dataset public[3]. In general, Eclipse plug-in tools were especially well-represented among the tools (e.g. [S4,S8]). One such example is ScrumCity [S4], which is an Eclipse plug-in and available only for Java programs.

Some works featured models that authors wished to improve and build as tools in the future – as they were not ready at the time of writing those papers, they are not listed in Table 9, however in this mapping study we shortly comment on them. For instance, [S9] discussed a work-in-progress reporting model for Agile projects. The authors of the paper reported that at that time, the method was being validated with new clients and they aimed to implement it in an automated tool focusing on backlog data from JIRA to automatically calculate metrics. Similarly, Lunesu et al. [S42] demonstrated a simulator which the authors planned to extend as a simulation tool sourcing data from issue tracking systems such as JIRA or Redmine.

5. Discussion

In this section, we review our findings as well as discuss their implications and limitations. We also comment and summarize best practices reported in the literature regarding applying BDA to ASD projects as well as the ASD process itself. We complement our discussion with related work and provide recommendations for the software engineering research community and practitioners alike. Figure 14 wraps up our key findings.

5.1. Analytics is not one-size-fits-all

Even though all companies covered in the selected primary studies adopted some of the Agile principles, they tend to differ in terms of size, structure, IT landscape, data ecosystem, among others. As those organizations are not homogeneous, their analytics needs and

[1] https://github.com/brunopedroso/besouro
[2] https::/probedock.io
[3] https://sites.google.com/site/appsuserreviews/

employed BDA solutions also differ, as we will show in this section.

5.1.1. Different analytics for different needs

Company size and budget. Even though at the beginning, Agile methods were designed for bottom-up adoption in small organizations, our literature review shows that large companies and corporations are the most frequently studied Agile organizations in terms of BDA. Also, from the analyzed studies, it becomes evident that big companies are more focused on leveraging the potential of BDA, because of the abundance of data, various data sources, and greater resources (i.e. money, labor). Larger budgets and headcount also allow bigger organizations to allocate more resources to collaborate with academia and publish their research. Consequently, not all software analytics efforts are created equal and some are more suitable for large organizations, whereas others are more appropriate for SMEs [S50,S63]. The lack of overlap, as Robbes et al. [S50] explained, lies in different analytics needs in the case of small companies. However, as authors [S50] emphasize, some "decision-making scenarios are similar (such as release planning, targeting training, and understanding customers)".

Reusing existing platforms, tools and building on top of them. Our literature review also allowed us to draw a conclusion that organizations exhibited a different degree of BDA adoption, also due to varying technological maturity. As we demonstrated in section 4.3.3, many works incorporated additional functionalities to existing tools or integrated with solutions already in use by the studied organizations. For instance, numerous authors mention that they plan to integrate their analytics solutions with tools such as JIRA [S9,42] - a popular choice among organizations tracking software issues. Leveraging existing platforms wherever possible, is a strategy advised by several works covered in our study [S20,S58].

5.1.2. Variety and availability of data sources - no silver bullet

Challenge to utilize heterogeneous data. Over a third of the selected studies used more than one data source. As illustrated in [S34], managing heterogeneous data sources and data orchestration is a complex task, which involves data collection, processing, and loading for further use in analytics solutions, which is a time-consuming effort. Hence providing a broad spectrum of data available in a timely manner is a challenge for organizations [S34,S40]. The complexity inevitably increases with ML-powered applications. Keeping track of the metadata characteristics of datasets and storing versioned datasets is an important challenge to consider [12, 11]. As reported in the literature, data versioning is not as prevalent practice as code versioning [11]. This an impediment for adopting especially more advanced forms of BDA.

Deficiencies in the availability of data. Often analytics teams and researchers need to resort to readily available data. Getting the exact type of data they would need for their analysis is frequently not possible [S40] or requires manual effort [S60], because mechanisms to automatically collect the data are not in place. There are numerous reasons for this: confidentiality restrictions, business constraints limiting the collection of such data (shortage of money, time, human resources), or simply lack of knowledge on the business side that such data could improve business operations of the company.

The amount of useful data is capped. Consistent with previous research [49], too much data may inhibit the decision-making process, and as a consequence, the abundance of data results in putting limits on their usage. For instance, several primary studies included in our mapping study demonstrated that the availability of data does not necessarily imply its usage [S50,S58]. For example, [S50] described a scenario where a company decided not to monitor neither its version control nor defect tracker systems, even though in other studies these types of data sources were used (e.g. [S5,S21]). Furthermore, inadequate analysis of available data is pervasive [S40]. Therefore, it is important to recognize what type of data would be needed for a given scenario and ensure its completeness and good quality.

5.1.3. Organizational entropy and data sharing practices

Organizational silos and legacy systems. It is also important to ensure there is a single version of the truth, which is technology-agnostic and comparable across applications [S58]. For instance, because of organizational silos that implied having different data sources with no comparable measures, Snyder et al. [S58] decided to use only one measure for analyzing productivity, even though more than one were available. Apart from aligning data sources at the company level, it is also important to ensure a common connectivity mechanism between applications (e.g. through well-defined APIs) [S32] as well as preserve backward compatibility [S20]. ASD is associated with reduced emphasis on documentation. However, as indicated by [S4,S28,S46] the low degree of architecture documentation adversely affects the maintenance of existing applications, in particular legacy software. Pareto et al. [S46] stressed that "lack of a clear architecture" significantly increases projects' lead time. In the same vein, [S32] contended that documenting software architecture in large-scale Agile development is an important challenge without a clear solution and it is especially important to provide detailed architectural guidance to the ASD teams at the beginning of the projects. The authors [S46] advocated prioritization of documentation in line with a single business goal to streamline development.

Data disclosure. As discussed in the previous paragraph, the availability of data within organizations often tends to be limited, not to mention data being shared with third-party or the public. Companies, in general, are focused on building a competitive advantage and, hence, are more sensitive about disclosing information considered intellectual property.

5.2. Analytics getting more sophisticated but with unbalanced growth

BDA solutions proposed by academics and practitioners differ in many respects. Limited data and AI/ML model sharing, scarce transfer of research outcomes from academia into practice widen the BDA inequality gap. ML techniques proposed by scholars today may no longer be competitive in the market in the future.

5.2.1. BDA adoption is not mature yet but is improving

As it couldn't be otherwise, more sophisticated analytics is on the rise in recent years. Although descriptive analytics is still the most prevalent among all discussed types of analytics as it is the most fundamental and hence clearly understood type. Nowadays, even the most basic variant of analytics enhances ASD in practice. The industry tends to implement and use tools that provide actionable insights, often based on analysis of metrics, without an explicit predictive component. The low number of adaptive analytics papers may be explained by the fact that it is a relatively new concept. However, one may assume it will most likely change in the future. With the increased adoption of simple solutions, we predict more use of advanced types of analytics (prescriptive, adaptive) in the coming years. Furthermore, different applications of adaptive analytics (even when not explicitly named as such) are being proposed in the scientific literature (e.g. [21]). Enriching software with adaptive capabilities is a promising research avenue.

5.2.2. Algorithms achieving best results today and tomorrow

Dominating algorithms. With regard to specific methods employed in the selected primary studies, techniques such as SVMs, NBs, NNs, or different ensemble methods were frequently chosen. Ensemble models, achieving strong results in many selected primary studies, are stacked models, hence provide often the best results. For instance, RF, popular among the authors of the studies, is an easy to train, ensemble decision tree. On the other hand, NNs and DL methods, in general, require large datasets and careful hyperparameter tuning. Hence, NNs need more time to train and are more prone to overfitting. In terms of methods, DL is gaining momentum (e.g. [S18]), however traditional statistical methods and regression models are still pervasive. Relatively low usage of process mining for process-related analysis in the software domain.

Data maintenance and model upkeep. Even the best performing model may not provide satisfying results in the future. Models degrade with time. Several measures need to be taken to make AI/ML models usable for a long time and sustain their performance. Data needs to be continuously updated and validated [S7]. It is important to constantly ensure that the model is adjusted to the changing data distribution. Accordingly, the management of data pipelines is critical so that the maintenance of the model is streamlined.

5.2.3. Little transparency of ML techniques used in industry, replication difficult

Half of the studies were written either as a joint collaboration between academia and industry or solely by industry practitioners, which is a positive sign.

Industry vs academia: different environments, different priorities. Industry-led research is less explicit when it comes to ML methods used in ASD than publications originating from academia. Apart from data disclosure policies, another possible explanation might be that business ventures provide high-level information because it is simply more critical for them to focus on business-relevant metrics and system design rather than on specific algorithms. In the same vein, as opposed to theoretical works, companies are more interested in satisfying business needs, which often require complex orchestration of existing systems. Further, the analysis of performance is oftentimes not a straightforward task. Unlike in academia, where a standard dataset can be used for evaluation, industry-led research is set in a specific context. As Zhang et al. [S65] contended, although researchers can measure intermediate results using evaluation criteria such as precision or recall, for real-world tasks these are often not sufficient, requiring empirical evaluations performed together with practitioners. Therefore, following the same evaluation procedure is frequently neither feasible nor practical.

Difficult operationalization and dissemination of academic research to industry. In order for ML models to work within the business environment, analytics teams need to consider many practical constraints. Our study reveals that papers focused on BDA in ASD favored new procedures and techniques - the large number of procedures and techniques can be attributed to the considerably high number of journal articles covered in our study. A possible explanation is that although BDA is a highly applied discipline. Nevertheless, the number of real-world applications of BDA in ASD with well-documented scientific evidence is still scarce. Our analysis of the state of industrial adoption and dissemination of research results originating from academia shows that the transfer of research outcomes into practice is not pervasive.

Replication is difficult. Research papers produced by academics often utilized data from open source projects. Nevertheless, the number of standard datasets for evaluation is relatively low. For instance, as reported in [S49], in the field of software estimation, datasets are often small, outdated, and suffer from homogeneity, which may cause under- or over-fitting while training an ML model. A possible explanation is that the generation of a high-quality dataset is time and resource-intensive. Furthermore, it is often a non-trivial task, as software repositories may vary in terms of the information granularity or quality. However, efforts to create benchmark datasets were undertaken by some scholars (e.g. [S18,S48]). Furthermore, most studies which disclosed techniques and models used (mostly performed by academics), did not provide information on the parameters. Hence replication of results, based on information provided, might be cumbersome. Also, releasing the code is not a common practice in SE. Nevertheless, based on how scholars in other domains (i.e. AI/ML) benefit from releasing the code to reproduce experimental results or to streamline the software development process, we hypothesize that it could also help in SE to establish baselines and advance the state of the art.

5.3. Analytics is already present in many areas of Agile software development, but the domain is not consolidated

ASD implementations may differ from company to company, but the main objective for BDA remains unchanged. BDA is predominantly employed to support informed decision-making and minimize project-related risks. To achieve better results, ASD processes can be tuned to provide BDA with more valuable data.

5.3.1. Key areas for implementing BDA come down to risk management

Data-driven ASD is mostly focused on project management analytics (effort estimation, resource allocation), requirements engineering, and software quality assurance (defects/bug fixing, testing). Hence, managing risk related to software development turned out to be the common denominator of the majority of studies. The authors of selected primary studies recognized the importance of minimizing delays (e.g. predicting the delay of issues) and predicting time for completing activities (e.g. predicting delivery capability or time to fix bugs). This can be explained by the fact that organizations are willing to apply BDA especially to ASD areas where they could suffer possibly the greatest capital and resource losses. Although studies discussing those areas confirm the potential of using ML models to improve activities in the respective fields, they rarely reach consensus regarding best approaches. Further studies are required to provide recommendations.

5.3.2. Variability in how companies interpret and implement Agile

There was considerable variability in how companies interpreted and implemented ASD. Also, the large number of corporations and big companies imposed that large-scale Agile implementations were also present in our study, perhaps being over-represented. Nevertheless, as such large ASD implementations generate a vast amount of data, it is understandable that BDA is particularly suitable for such applications. Moreover, especially among purely academic publications with no industrial use cases, Agile development context was rarely in focus. Given scarce information on analytics suited to specific Agile practices/techniques, we may conclude that the domain is not consolidated. However, a few common themes emerged:

Effort estimation is highly context-dependent, but a set of key features for cross-project estimation exists for BDA. The efficiency of utilization of team resources depends to a large extent on the experience of the users and is context-dependent [S38]. Story point estimation in ASD is specific to teams and projects [S18]. "In an issue report, the fields containing a summary, description, names of related components, and issue type provide relevant features for story point estimation. Most frequently, these features are project dependent." [S48]. Furthermore, analyses of feature relevance show that although features are highly dependent on the project and prediction stage, certain properties are important for most projects and phases. Such as a requirement creator, time remained to the end of an iteration, time since last requirement summary change, and the number of times requirement has been replanned for a new iteration [S22]. Cross-project estimation is possible, but less accurate than estimation within the same organization [S18]. Moreover, Dehghan et al. [S22] contend that although satisfying predictions can be made at the early stages, the performance of predictions improves over time by taking advantage of requirements' progress data. Therefore, as advocated by Abrahamsson et al. [S1], BDA-aided effort estimation that may prove especially helpful for inexperienced developers with limited estimation and software implementation experience. ML-models to validate their estimates or as an initial first estimate [S1]. With respect to productivity, Balogh et al. [S6] stressed the importance of measuring productivity at a fine-grained level.

Data presentation is no less important. Numerous studies (e.g. [S4,S5,S30,S60,S63]) stressed the importance of visual design, navigation, and visualization techniques in BDA as it improves comprehension of data. With regard to ASD, Alshakhouri et al. [S4] argued that visual analytics in software engineering mainly provides insights about intangible software product artifacts (such as program runtime behavior or software evolution), while software process visualization is far less popular and deserves more attention. In the same vein, Augustine et al. [S5] demonstrated the ability of visual analytics to improve the consistency of process implementation among software development teams. Furthermore, numerous studies (e.g. [S5,S63]) proved that visual representation of data often offers the kind of insights that are especially useful for decision-makers, such as dashboards visualizing most vital metrics for ASD processes and KPIs. Similar to [49, 29], we conclude that visualization that supports the decision-making process is a very important addition to the BDA toolchain.

5.3.3. Design and implementation of ASD processes are key to better support BDA

All too often, BDA is built to support existing ASD processes with a presumption that those processes cannot change. That is to say, approaches for leveraging BDA are mostly top-down: with ASD process in place, once data is available, a BDA solution is built. In contrast, we perceive that key to increasing the adoption of BDA in ASD is to implement ASD processes in organizations in a way that better accommodates BDA. Namely, the bottom-up approach could focus on identifying what value should the BDA solution deliver and what data it needs. Based on that, the focus should be shifted to continually improve the quality of data and Agile practices to better serve the BDA solution. The changes in the ASD process can be small but may greatly improve the quality of insights provided by the analytics and, in result, decision-making. There is a wealth of research on adopting ASD processes for ML and AI, which is beyond the scope of this mapping study. Hence here, we will only highlight some of the approaches proposed in the selected primary studies to improve Agile practices so that they better support BDA.

- ☐ *Backlog*: projects often have ill-structured product backlog, with missing properties for product backlog items [S9]. The quality of backlog data can be improved with dedicated tools (e.g. JIRA, VersionOne) offering standardized backlog functionality [S9].
- ☐ *Features*: it is important to capture information such as the requirement creator, time remained to the end of an iteration, time since last requirement summary change and the number of times requirement replanned for a new iteration [S22].
- ☐ *User stories*: user stories should be written with a higher level of detail, which is more suitable for automatic prediction of effort [S1]. Abrahamsson et al. suggest that "very short (containing a few words), informal user stories do not provide enough information to train any effort prediction model that would yield results with a required level of accuracy."

In general, it is advised to capture fine-grained data. For instance, detailed descriptions of product backlog items, their creators, time remained to the end of an iteration, time past since the last requirement summary change, the number of times requirements are replanned for a new iteration. Such a well-structured approach to gathering evidence can be aided by a dedicated Agile product lifecycle management tools.

5.4. *Feedback and behavior analysis should be at the heart of the BDA strategy for ASD*

Measuring customer feedback unobtrusively (e.g. telemetry, sentiment analysis of online reviews) offers unprecedented opportunity to understand customers' needs and consequently set proper product objectives. Yet, the opportunity has not been fully exploited so far.

5.4.1. *Mobile apps lead in analyzing explicit user feedback*

According to our analysis, the majority of customer feedback is gathered and analyzed for mobile applications. Data comes predominantly from user reviews in application stores. However, as Chen et al. [S15] found, only about 35% of app reviews provide insights that are directly applicable so that developers can improve the applications. We would like to see more industrial studies investigating data gathered from customer feedback and experimentation other than mobile applications.

5.4.2. *Leveraging usage data coming from different data sources*

Systematic experiments with customers are rare [S40], which authors of the selected primary studies advocate [S24]. Very rarely studies monitored the usage of their proposed solutions and evaluated feedback of the users given (both implicitly or explicitly) during their regular work. Analyzing implicit feedback is critical as it lacks many biases of explicit feedback. In many cases, the product instrumentation only covered performance data and basic user demographics. However, the awareness of companies of the importance of product usage data is growing, as some studies suggest [S40]. Companies need to start collecting product usage data, such as telemetry from IDEs, or various ALM systems.

5.4.3. *Guiding product development in ASD through BDA*

We expect to see more research on BDA that guides product development in an Agile environment, which also includes AI-powered applications. For instance, online controlled experiments (OCEs) approach to rolling out changes to ML-centric software. Some of the covered studies already incorporated mechanisms to gather feedback. For example [S52], described a tool that includes a feedback loop that helps to improve the platform.

5.5. *Threats to Validity*

In order to ensure the high quality of our study in terms of completeness and rigor, we identified the following threats to validity and implemented appropriate mitigation strategies. Our threats to validity fall into the following main categories [42]:

Incompleteness of search strategy (internal validity). The snowballing method, despite being considered to give comparable results to database search method [28], suffers from exhaustiveness. Hence, the results of this study can be skewed, as not all papers might be included (i.e. could be mistakenly excluded or missed by the first author doing the review). However, as Kitchenham et al. [32] contend, this is also a defining characteristic of mapping studies in general. What is more, limiting the number of venues based on their ranking and relevancy of published papers also impacted the list of our selected primary studies.

Sample size representativeness (conclusion validity). The sample size of the literature review was relatively small and we were clearly not able to capture all studies covering the topic for the selected period. Nevertheless, we believe that the sample size should be representative of this area of research. To reduce this threat, in our literature review, we extracted data for analytical reviews. As it turns out, similar to [8], *IEEE Software* happened to be the most frequently represented publishing venue in our study. However, we do not consider this an anomaly. Unlike in [8], our study is not skewed by the 2013 special issue of *IEEE Software* on software analytics. Moreover, also in the work of Dingsøyr et al. [23] the aforementioned journal was the major journal outlet for ASD related works. Although there were companies that were more frequently represented in our study than others – i.e. Microsoft [S14,S20,S24,S65], Ericsson [S46,S60], and Google [S23,S52]. However, they together accounted only for about 15% of all studies. Hence, we did not face a problem similar to the one identified in [33], where the majority of primary studies were conducted with companies known for collaborating with empirical software engineering researchers on a regular basis. Nevertheless, 9 papers did not reveal the names of industrial partners participating in those studies (some of those papers discussed more than one industrial case study). Therefore, in total, names of 21 companies were not disclosed; hence we cannot entirely rule out the possibility of having a bias towards particular companies, as it might be implicit.

Researcher bias (construct validity). Since the study selection and data extraction was performed by a single researcher, in principle, there was a high probability that researcher bias could potentially adversely affect the overall result of the mapping study. However, in order to reduce the risks of research bias, we first developed and later followed a structured review method when conducting this study (designed in the form of a research protocol - demonstrated in section 3.1). Hence, we argue that no systematic errors were introduced in the course of the study selection and data extraction that could impact our findings.

Naming confusion (external validity). Agile methodology, and ASD in particular, not always appear in titles, author keywords, index terms, or even in abstracts of the selected primary studies (e.g. [S16,S18]). Furthermore, ASD is often referred to as iterative software development in papers (e.g. [S17]). In addition, several concepts such as regression testing, automatic app review classification (e.g. [S14,S30,S45]) or continuous integration, although popular in ASD, are not directly linked to Agile practices in papers describing them. For that reason, it was not always clear if a particular work discussed only plan-driven software development approaches or rather focused on modern Agile development practices, where incremental and iterative development plays a vital role. For instance, although the Rational Unified Process (RUP) process framework is not ASD per se, it encompasses many Agile-inspired techniques. Therefore, we often had to infer some information from text, and, as in the case of RUP (e.g. [S46,S47,S50]), such studies were also included in our survey. In the same vein, naming confusion also occurred for the BDA concept. Terms such as software analytics and data analytics are used interchangeably in the literature. Also, AI, ML, and DL can be found in a growing body of studies on SE in ASD. Even though their meaning is slightly different, they also share a number of common properties. Furthermore, software analytics or data analytics belong to a common category. In order to provide non-obvious insights and improve the decision-making process, they need to employ ML, DL, or AI methods in some shape or form. Those methods require a significant amount of data (often complex or unstructured) to train models. For that reason, we decided to use the term BDA as an umbrella term for all the concepts mentioned above.

6. Conclusion

By performing a mapping study, we examined a body of primary research studies, and we described how data analytics improves ASD. We also found that past work in the concerned field is mainly focused on requirements and delivery capabilities. As much of the work discussed in this SM study takes the form of solution proposals, rather than empirical studies in industry, we suggest that future research would benefit from: (1) more generalizable, representative empirical studies discussing methods and their applications not only at the level of experience reports or solution proposals, (2) more industrial studies investigating data gathered from customer feedback and experimentation other than mobile applications. It is especially important, because it is said that up to 80% of products lose money due to wrongly set product objectives that result in building products that customers do not need [6]. We expect to see more research on BDA that guides product development in an Agile environment.
Further research on this topic may focus on reviewing grey literature which, as we hypothesize, covers related approaches to those discussed in the selected primary studies, but seemingly in a less rigorous manner. Finally, it appears that ASD and BDA alike are very much similar to other phenomena that over the decades, were once popular and considered breakthroughs, but eventually became absorbed into what is called *business as usual*. In that respect, ASD and BDA concepts will most probably not fade away, but rather become embedded in many activities of modern software companies. Hence, at this point, it will be pointless to call them out as they will be an integral part of business operations.

Acknowledgments

This work is supported in part by the Catalan Agencia de Gestión de Ayudas Universitarias y de Investigación (AGAUR) through the FI PhD grant. The research is also partially supported by the Spanish Ministerio de Economía, Industria y Competitividad through the GENESIS project (grant TIN2016-79269- R).

Appendix A. Additional Resources

The spreadsheets used to track the following metadata and RQ-related information from the reviewed primary studies:

- Data extraction form
- Selected and excluded publishing outlets

can be found under the following link: `https://mydisk.cs.upc.edu/s/G5XfnkFntj6oKoe`

References

[1] The apache software foundation. `https://www.apache.org/foundation/how-it-works.html`. Accessed: 2018-07-30.

[2] Hype cycle for emerging technologies. `https://www.gartner.com/doc/2571624/hype-cycle-emerging-technologies-`, 2003. Accessed: 2018-07-30.

[3] The agile system development lifecycle (sdlc). `http://www.ambysoft.com/essays/agileLifecycle.html`, 2005. Accessed: 2018-07-30.

[4] Less framework. `http://less.works/`, 2015. Accessed: 2018-07-30.

[5] Scaled agile framework. `http://scaledagileframework.com/`, 2015. Accessed: 2018-07-30.

[6] *Visions*, 40(3), 2016.

[7] The 12th annual state of agile report. `https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report`, 2018. Accessed: 2019-02-25.

[8] Tamer Mohamed Abdellatif, Luiz Fernando Capretz, and Danny Ho. Software analytics to software practice: A systematic literature review. In *Proceedings of the First International Workshop on BIG Data Software Engineering*, BIGDSE '15, pages 30–36, Piscataway, NJ, USA, 2015. IEEE Press.

[9] Scott W. Ambler. Balancing agility and formalism in software engineering. chapter Agile Software Development at Scale, pages 1–12. Springer-Verlag, Berlin, Heidelberg, 2008.

[10] David Ameller, Xavier Burgués, Oriol Collell, Dolors Costal, Xavier Franch, and Mike P. Papazoglou. Development of service-oriented architectures using model-driven development. *Inf. Softw. Technol.*, 62(C):42–66, June 2015.

[11] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300. IEEE, 2019.

[12] Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. Software engineering challenges of deep learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 50–59. IEEE, 2018.

[13] Selami Bagriyanik and Adem Karahoca. Big data in software engineering: A systematic literature review. *Global Journal of Information Technology*, 6(1):107–116, 03 2016.

[14] Boehm and Richard Turner. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[15] Pierre Bourque and Richard E. Fairley, editors. *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, version 3.0 edition, 2014.

[16] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(40):571–583, 2007.

[17] David Budgen, Mark Turner, Pearl Brereton, and Barbara Kitchenham. Using mapping studies in software engineering. In *PPIG*, page 20. Psychology of Programming Interest Group, 2008.

[18] Fabio Calefato and Filippo Lanubile. Using frameworks to develop a distributed conferencing system: an experience report. *Software: Practice and Experience*, 39(15):1293–1311, 2009.

[19] Amadeu Silveira Campanelli and Fernando Silva Parreiras. Agile methods tailoring–a systematic literature review. *Journal of Systems and Software*, 110:85–100, 2015.

[20] Longbing Cao. Data science: A comprehensive overview. *ACM Comput. Surv.*, 50(3):43:1–43:42, June 2017.

[21] Hoa Khanh Dam, Truyen Tran, John Grundy, Aditya Ghose, and Yasutaka Kamei. Towards effective ai-powered agile project management. *arXiv preprint arXiv:1812.10578*, 2018.

[22] Torgeir Dingsøyr, Nils Brede Moe, Tor Erlend Fægri, and Eva Amdahl Seim. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23(1):490–520, Feb 2018.

[23] Torgeir Dingsøyr, Sridhar Nerur, VenuGopal Balijepally, and Nils Brede Moe. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6):1213 – 1221, 2012. Special Issue: Agile Development.

[24] Tore Dybå and Torgeir Dingsøyr. Empirical studies of agile software development: A systematic review. *Inf. Softw. Technol.*, 50(9-10):833–859, August 2008.

[25] Øyvind Hauge, Claudia P. Ayala, and Reidar Conradi. Adoption of open source software in software-intensive organizations - a systematic literature review. *Information & Software Technology*, 52(11):1133–1154, 2010.

[26] Rashina Hoda, Norsaremah Salleh, John Grundy, and Hui Mien Tee. Systematic literature reviews in agile software development: A tertiary study. *Information and Software Technology*, 85:60 – 70, 2017.

[27] Han Hu, Yonggang Wen, Tat-Seng Chua, and Xuelong Li. Toward scalable systems for big data analytics: A technology tutorial. *IEEE Access*, 2:652–687, 2014.

[28] Samireh Jalali and Claes Wohlin. Systematic literature studies: Database searches vs. backward snowballing. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '12, pages 29–38, New York, NY, USA, 2012. ACM.

[29] Marijn Janssen, Haiko van der Voort, and Agung Wahyudi. Factors influencing big data decision-making quality. *Journal of Business Research*, 70:338–345, 2017.

[30] B. Kitchenham and S Charters. Guidelines for performing systematic literature reviews in software engineering. *Technical Report EBSE-2007-01, School of Computer Science and Mathematics, Keele University*, 2007.

[31] Barbara Kitchenham. What's up with software metrics? - a preliminary mapping study. *J. Syst. Softw.*, 83(1):37–51, January 2010.

[32] Barbara A. Kitchenham, David Budgen, and O. Pearl Brereton. Using mapping studies as the basis for further research - a participant-observer case study. *Inf. Softw. Technol.*, 53(6):638–651, June 2011.

[33] Eetu Kupiainen, Mika Mäntylä, and Juha Itkonen. Using metrics in agile and lean software development - a systematic literature review of industrial studies. *Information & Software Technology*, 62:143–163, 2015.

[34] Maarit Laanti, Outi Salo, and Pekka Abrahamsson. Agile methods rapidly replacing traditional methods at nokia: A survey of opinions on agile transformation. *Inf. Softw. Technol.*, 53(3):276–290, March 2011.

[35] Ayman Meidan, J. A. García-García, Isabel M. Ramos, and María José Escalona Cuaresma. Measuring software process: A systematic mapping study. *ACM Comput. Surv.*, 51:58:1–58:32, 2018.

[36] Ayman Meidan, Julián A. García-García, Isabel Ramos, and María José Escalona. Measuring software process: A systematic mapping study. *ACM Comput. Surv.*, 51(3):58:1–58:32, June 2018.

[37] Tim Menzies and Thomas Zimmermann. Goldfish bowl panel:

Software development analytics. In *Proceedings of the 34th International Conference on Software Engineering*, pages 1032–1033. IEEE Press, 2012.

[38] Tim Menzies and Thomas Zimmermann. Software analytics: so what? *IEEE Software*, 30(4):31–37, 2013.

[39] Tim Menzies and Thomas Zimmermann. Software analytics: What's next? *IEEE Software*, 35(5):64–70, 2018.

[40] UNESCO Institute of Statistics. How much does your country invest in r&d? `http://uis.unesco.org/apps/visualisations/research-and-development-spending/`. Accessed: 2020-02-02.

[41] Maria Paasivaara, Benjamin Behm, Casper Lassenius, and Minna Hallikainen. Large-scale agile transformation at ericsson: a case study. *Empirical Software Engineering*, Jan 2018.

[42] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1 – 18, 2015.

[43] Kai Petersen and Claes Wohlin. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15:654–693, 2010.

[44] Dong Qiu, Bixin Li, Shunhui Ji, and Hareton K. N. Leung. Regression testing of web service: A systematic mapping study. *ACM Comput. Surv.*, 47:21:1–21:46, 2014.

[45] Ali Ahmadian Ramaki, Abbas Rasoolzadegan Barforoush, and Abbas Ghaemi Bafghi. A systematic mapping study on intrusion alert analysis in intrusion detection systems. *ACM Comput. Surv.*, 51:55:1–55:41, 2018.

[46] Davi Silva Rodrigues, Márcio Eduardo Delamaro, Cléber Gimenez Corrêa, and Fátima de Lourdes dos Santos Nunes. Using genetic algorithms in test data generation: A critical systematic mapping. *ACM Comput. Surv.*, 51:41:1–41:23, 2018.

[47] Mary Shaw. Writing good software engineering research papers: Minitutorial. In *Proceedings of the 25th International Conference on Software Engineering*, ICSE '03, pages 726–736. IEEE Computer Society, 2003.

[48] Cassidy R. Sugimoto, Nicolas Robinson-Garcia, Dakota Murray, Alfredo Yegros-Yegros, Rodrigo Costas, and Vincent Larivière. Scientists have most impact when they're free to move. *Nature*, 550 7674:29–31, 2017.

[49] Richard Berntsson Svensson, Robert Feldt, and Richard Torkar. The unfulfilled potential of data-driven decision making in agile software development. In *International Conference on Agile Software Development*, pages 69–85. Springer, 2019.

[50] Ambler SW, editor. *Disciplined Agile Delivery: a practitioner's guide to agile software delivery in the enterprise*. IBM Press, 2012.

[51] Caroline S. Wagner and Koen Jonkers. Open countries have strong science. *Nature*, 550:32–33, 2017.

[52] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Martin J. Shepperd, Tracy Hall, and Ingunn Myrtveit, editors, *EASE*, pages 38:1–38:10. ACM, 2014.

[53] Claes Wohlin. Second-generation systematic literature studies using snowballing. In Sarah Beecham, Barbara Kitchenham, and Stephen G. MacDonell, editors, *EASE*, pages 15:1–15:6. ACM, 2016.

Primary studies

[S1] P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko, and W. Pedrycz. Predicting development effort from user stories. In *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on*, pages 400–403. IEEE, 2011.

[S2] S. D. A. Alam, M. R. Karim, D. Pfahl, and G. Ruhe. Comparative analysis of predictive techniques for release readiness classification. In *Proceedings of the 5th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering*, pages 15–21. ACM, 2016.

[S3] S. D. A. Alam, D. Pfahl, and G. Ruhe. Release readiness classification: An explorative case study. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 27. ACM, 2016.

[S4] M. Alshakhouri, J. Buchan, and S. G. MacDonell. Synchronised visualisation of software process and product artefacts: Concept, design and prototype implementation. *Information and Software Technology*, 98:131–145, 2018.

[S5] V. Augustine, J. Hudepohl, P. Marcinczak, and W. Snipes. Deploying software team analytics in a multinational organization. *IEEE Software*, 35(1):72–76, 2018.

[S6] G. Balogh, G. Antal, Á. Beszédes, L. Vidács, T. Gyimóthy, and Á. Z. Végh. Identifying wasted effort in the field via developer interaction data. In *Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference on*, pages 391–400. IEEE, 2015.

[S7] F. A. Batarseh and A. J. Gonzalez. Predicting failures in agile software development through data analytics. *Software Quality Journal*, 26(1):49–66, 2018.

[S8] K. Becker, B. de Souza Costa Pedroso, M. S. Pimenta, and R. P. Jacobi. Besouro: A framework for exploring compliance rules in automatic tdd behavior assessment. *Information and Software Technology*, 57:494–508, 2015.

[S9] M. P. Boerman, Z. Lubsen, D. A. Tamburri, and J. Visser. Measuring and monitoring agile development status. In *Proceedings of the Sixth International Workshop on Emerging Trends in Software Metrics*, pages 54–62. IEEE Press, 2015.

[S10] P. Braione, G. Denaro, A. Mattavelli, M. Vivanti, and A. Muhammad. Software testing with code-based test generators: data and lessons learned from a case study with an industrial software component. *Software Quality Journal*, 22(2):311–333, 2014.

[S11] B. Busjaeger and T. Xie. Learning for test prioritization: an industrial case study. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 975–980. ACM, 2016.

[S12] F. Calefato, F. Lanubile, T. Conte, and R. Prikladnicki. Assessing the impact of real-time machine translation on multilingual meetings in global software projects. *Empirical Software Engineering*, 21(3):1002–1034, 2016.

[S13] G. Çalikli, M. Staron, and W. Meding. Measure early and decide fast: transforming quality management and measurement to continuous deployment. In *Proceedings of the 2018 International Conference on Software and System Process*, pages 51–60. ACM, 2018.

[S14] R. Carlson, H. Do, and A. Denton. A clustering approach to improving test case prioritization: An industrial case study. In *ICSM*, pages 382–391. IEEE Computer Society, 2011.

[S15] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang. Arminer: mining informative reviews for developers from mobile app marketplace. In P. Jalote, L. C. Briand, and A. van der Hoek, editors, *ICSE*, pages 767–778. ACM, 2014.

[S16] M. Choetkiertikul, H. K. Dam, T. Tran, and A. Ghose. Predicting the delay of issues with due dates in software projects. *Empirical Software Engineering*, 22(3):1223–1263, 2017.

[S17] M. Choetkiertikul, H. K. Dam, T. Tran, A. Ghose, and J. Grundy. Predicting delivery capability in iterative software development. *IEEE Transactions on Software Engineering*, 44(6):551–573, 2018.

[S18] M. Choetkiertikul, H. K. Dam, T. Tran, T. Pham, A. Ghose, and T. Menzies. A deep learning model for estimating story points. *IEEE Transactions on Software Engineering*, PP(99):1, 2018.

[S19] R. M. Czekster, P. Fernandes, L. Lopes, A. Sales, A. R. Santos, and T. Webber. Stochastic performance analysis of global software development teams. *ACM Trans. Softw. Eng. Methodol.*, 25(3):26, 2016.

[S20] J. Czerwonka, N. Nagappan, W. Schulte, and B. Murphy. Codemine: Building a software development data analytics platform at microsoft. *IEEE Software*, 30(4):64–71, 2013.

[S21] D. A. da Costa, S. McIntosh, C. Treude, U. Kulesza, and A. E. Hassan. The impact of rapid release cycles on the integration delay of fixed issues. *Empirical Software Engineering*, 23(2):835–904, 2018.

[S22] A. Dehghan, A. Neal, K. Blincoe, J. Linaker, and D. Damian. Predicting likelihood of requirement implementation within the planned iteration: an empirical study at ibm. In *Mining Software Repositories (MSR), 2017 IEEE/ACM 14th International Conference on*, pages 124–134. IEEE, 2017.

[S23] S. Elbaum, G. Rothermel, and J. Penix. Techniques for improving regression testing in continuous integration development environments. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 235–245. ACM, 2014.

[S24] A. Fabijan, P. Dmitriev, H. H. Olsson, and J. Bosch. The evolution of continuous experimentation in software product development: from data to a data-driven organization at scale. In *ICSE*, pages 770–780. IEEE / ACM, 2017.

[S25] B. Fitzgerald, M. Musiał, and K.-J. Stol. Evidence-based decision making in lean software project management. In *Companion Proceedings of the 36th International Conference on Software Engineering*, pages 93–102. ACM, 2014.

[S26] A. S. Freire, M. Perkusich, R. M. Saraiva, H. O. de Almeida, and A. Perkusich. A bayesian networks-based approach to assess and improve the teamwork quality of agile teams. *Information Software Technology*, 100:119–132, 2018.

[S27] D. Fucci, H. Erdogmus, B. Turhan, M. Oivo, and N. Juristo. A dissection of the test-driven development process: Does it really matter to test-first or to test-last? *IEEE Transactions on Software Engineering*, 43(7):597–614, 2017.

[S28] J. García and K. Garcés. Improving understanding of dynamically typed software developed by agile practitioners. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 908–913. ACM, 2017.

[S29] M. Golfarelli, S. Rizzi, and E. Turricchia. Multi-sprint planning and smooth replanning: An optimization model. *Journal of systems and software*, 86(9):2357–2370, 2013.

[S30] E. Guzman, M. El-Haliby, and B. Bruegge. Ensemble methods for app review classification: An approach for software evolution. In *ASE*, pages 771–776. IEEE Computer Society, 2015.

[S31] M. Habayeb, S. S. Murtaza, A. V. Miranskyy, and A. B. Bener. On the use of hidden markov model to predict the time to fix bugs. *IEEE Transactions on Software Engineering*, 44:1224–1244, 2018.

[S32] V. Heikkilä, M. Paasivaara, K. Rautiainen, C. Lassenius, T. Toivola, and J. Järvinen. Operational release planning in large-scale scrum with multiple stakeholders - a longitudinal case study at f-secure corporation. *Information and Software Technology*, 57:116–140, 2015.

[S33] G. Hu, L. Zhu, and J. Yang. Appflow: using machine learning to synthesize robust, reusable ui tests. In *ESEC/SIGSOFT FSE*, 2018.

[S34] H. Huijgens, R. Lamping, D. Stevens, H. Rothengatter, G. Gousios, and D. Romano. Strong agile metrics: mining log data to determine predictive power of software metrics for continuous delivery teams. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 866–871. ACM, 2017.

[S35] J. O. Johanssen, A. Kleebaum, B. Brügge, and B. Paech. Feature crumbs: Adapting usage monitoring to continuous software engineering. In *PROFES*, 2018.

[S36] M. R. Karim, A. Alam, S. Didar, S. J. Kabeer, G. Ruhe, B. Baluta, and S. Mahmud. Applying data analytics towards optimized issue management: An industrial case study. In *Proceedings of the 4th International Workshop on Conducting Empirical Studies in Industry*, pages 7–13. ACM, 2016.

[S37] O. Liechti, J. Pasquier, and R. Reis. Supporting agile teams with a test analytics platform: a case study. In *Automation of Software Testing (AST), 2017 IEEE/ACM 12th International Workshop on*, pages 9–15. IEEE, 2017.

[S38] J. Lin. Context-aware task allocation for distributed agile team. In *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering*, pages 758–761. IEEE Press, 2013.

[S39] J. Lin, H. Yu, Z. Shen, and C. Miao. Studying task allocation decisions of novice agile teams with data from agile project management tools. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, pages 689–694. ACM, 2014.

[S40] E. Lindgren and J. Münch. Raising the odds of success: the current state of experimentation in product development. *Information and Software Technology*, 77:80–91, 2016.

[S41] M. Lu and P. Liang. Automatic classification of non-functional requirements from augmented app user reviews. In *EASE*, pages 344–353. ACM, 2017.

[S42] M. I. Lunesu, J. Münch, M. Marchesi, and M. Kuhrmann. Using simulation for understanding and reproducing distributed software development processes in the cloud. *Information Software Technology*, 103:226–238, 2018.

[S43] W. Maalej, M. Nayebi, T. Johann, and G. Ruhe. Toward data-driven requirements engineering. *IEEE Software*, 33(1):48–54, 2016.

[S44] M. Manzano, E. Mendes, C. Gómez, C. P. Ayala, and X. Franch. Using bayesian networks to estimate strategic indicators in the context of rapid software development. In *PROMISE*, 2018.

[S45] D. Marijan, A. Gotlieb, and S. Sen. Test case prioritization for continuous regression testing: An industrial case study. In *ICSM*, pages 540–543. IEEE Computer Society, 2013.

[S46] L. Pareto, A. B. Sandberg, P. S. Eriksson, and S. Ehnebom. Collaborative prioritization of architectural concerns. 85(9):1971–1994, 2012.

[S47] F. Paz, C. Zapata, and J. A. Pow-Sang. An approach for effort estimation in incremental software development using cosmic function points. In M. Morisio, T. Dybå, and M. Torchiano, editors, *ESEM*, pages 44:1–44:4. ACM, 2014.

[S48] S. Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli. Estimating story points from issue reports. In *Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering*, page 2. ACM, 2016.

[S49] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski. An effective approach for software project effort and duration estimation with machine learning algorithms. *Journal of Systems* *and Software*, 137:184–196, 2018.

[S50] R. Robbes, R. Vidal, and M. C. Bastarrica. Are software analytics efforts worthwhile for small companies? the case of amisoft. *IEEE software*, 30(5):46–53, 2013.

[S51] P. Rodeghero, S. Jiang, A. Armaly, and C. McMillan. Detecting user story information in developer-client conversations to generate extractive summaries. In *Software Engineering (ICSE), 2017 IEEE/ACM 39th International Conference on*, pages 49–59. IEEE, 2017.

[S52] C. Sadowski, J. Van Gogh, C. Jaspan, E. Söderberg, and C. Winter. Tricorder: Building a program analysis ecosystem. In *Proceedings of the 37th International Conference on Software Engineering-Volume 1*, pages 598–608. IEEE Press, 2015.

[S53] Z. Sahaf, V. Garousi, D. Pfahl, R. Irving, and Y. Amannejad. When to automate software testing? decision support based on system dynamics: an industrial case study. In *Proceedings of the 2014 International Conference on Software and System Process*, pages 149–158. ACM, 2014.

[S54] C. J. T. Salinas, J. Sedeño, M. J. E. Cuaresma, and M. Mejías. Estimating, planning and managing agile web development projects under a value-based perspective. *Information and Software Technology*, 61:124–144, 2015.

[S55] A. Scheerer, S. Bick, T. Hildenbrand, and A. Heinzl. The effects of team backlog dependencies on agile multiteam systems: A graph theoretical approach. In T. X. Bui and R. H. S. Jr., editors, *HICSS*, pages 5124–5132. IEEE Computer Society, 2015.

[S56] G. Schermann and P. Leitner. Search-based scheduling of experiments in continuous deployment. *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 485–495, 2018.

[S57] E. Scott and D. Pfahl. Using developers' features to estimate story points. In *ICSSP*, 2018.

[S58] B. Snyder and B. Curtis. Using analytics to guide improvement during an agile–devops transformation. *IEEE Software*, 35(1):78–83, 2018.

[S59] R. R. G. Souza, C. von Flach G. Chavez, and R. A. Bittencourt. Rapid releases and patch backouts: A software analytics approach. *IEEE Software*, 32(2):89–96, 2015.

[S60] M. Staron and W. Meding. Monitoring bottlenecks in agile and lean software development projects–a method and its industrial use. In *International Conference on Product Focused Software Process Improvement*, pages 3–16. Springer, 2011.

[S61] A. Szoke. Conceptual scheduling model and optimized release scheduling for agile environments. *Information and Software Technology*, 53(6):574–591, 2011.

[S62] E. L. Vargas, J. Hejderup, M. Kechagia, M. Bruntink, and G. Gousios. Enabling real-time feedback in software engineering. *2018 IEEE/ACM 40th International Conference on Software Engineering: New Ideas and Emerging Technologies Results (ICSE-NIER)*, pages 21–24, 2018.

[S63] K. Wnuk, T. Gorschek, D. Callele, E.-A. Karlsson, E. Åhlin, and B. Regnell. Supporting scope tracking and visualization for very large-scale requirements engineering-utilizing fsc+, decision patterns, and atomic decision visualizations. *IEEE Transactions on Software Engineering*, 42(1):47–74, 2016.

[S64] L. Xiao, Z. Yu, B. Chen, and X. Wang. How robust is your development team? *IEEE Software*, 35(1):64–71, 2018.

[S65] D. Zhang, S. Han, Y. Dang, J.-G. Lou, H. Zhang, and T. Xie. Software analytics in practice. *IEEE software*, 30(5):30–37, 2013.