The Journal of Systems & Software 203 (2023) 111749

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Behaviour driven development: A systematic mapping study☆

Leonard Peter Binamungu *, Salome Maro

Department of Computer Science and Engineering, College of Information and Communication Technologies, University of Dar es Salaam, Dar es Salaam, Tanzania

ARTICLE INFO

Article history:
Received 7 July 2022
Received in revised form 24 April 2023
Accepted 8 May 2023
Available online 19 May 2023

Dataset link: https://doi.org/10.5281/zenodo.7934235

Keywords:
Behaviour Driven Development
Systematic mapping study
Systematic mapping studies in software engineering

ABSTRACT

Context: Behaviour Driven Development (BDD) uses scenarios written in semi-structured natural language to express software requirements in a way that can be understood by all stakeholders. The resulting natural language specifications can also be executed to reveal correct and problematic parts of a software. Although BDD was introduced about two decades ago, there is a lack of secondary studies in peer-reviewed scientific literature, making it difficult to understand the state of BDD research and existing gaps.
Objective: To understand the current state of BDD research by conducting a systematic mapping study that covers studies published from 2006 (when BDD was introduced) to 2021.
Method: By following the guidelines for conducting systematic mapping studies in software engineering, we sought to answer research questions on types of venues in which BDD papers have been published, research types, contribution types, studied topics and their evolution, as well as evaluation methods used in published BDD research.
Results: The study identified 166 papers which were mapped. Key results include the following: the dominance of conference papers; scarcity of research with insights from the industry; shortage of philosophical papers on BDD; acute shortage of metrics for measuring various aspects of BDD specifications and the processes for producing BDD specifications; the dominance of studies on using BDD for facilitating various software development endeavours, improving the BDD process and associated artefacts, and applying BDD in different contexts; scarcity of studies on using BDD alongside other software techniques and technologies; increase in diversity of studied BDD topics; and notable use of case studies and experiments to study different BDD aspects.
Conclusion: The paper improves our understanding of the state of the art of BDD, and highlights important areas of focus for future BDD research.

1. Introduction

Behaviour Driven Development (BDD) (North, 2006; Wynne and Hellesoy, 2012) is an agile technique in which software requirements are specified in a semi-structured natural language using Given-When-Then to express examples (also called *scenarios*) of expected software behaviour or how a user will interact with a software system. Apart from acting as software requirements, the scenarios can also act as test cases, which can be used to verify if the software is behaving as expected. Scenarios are connected to the production code of the System Under Test (SUT) using glue code. The use of glue code means that the scenarios can be executed to determine the correct and problematic parts of the SUT.

Since BDD was introduced about two decades ago, the Software Engineering research community has investigated various aspects of BDD. Additionally, BDD is becoming an established industry practice for software development and is currently used in different domains (Binamungu et al., 2018b). However, little is known about what has been studied and what has not been studied regarding BDD due to the lack of existing systematic mapping studies as well as systematic literature reviews on the topic. This creates a need to identify what has been most studied and to what extent in order to identify gaps that could inform future research on BDD. A few secondary studies have been conducted about BDD (Abushama et al., 2020; Egbreghts, 2017; Lillnor and He, 2020). However, these studies were about the impacts of BDD on software projects (Abushama et al., 2020) or were student projects that never got published in peer-reviewed scientific literature (Egbreghts, 2017; Lillnor and He, 2020). This poses a challenge regarding the availability and reliability of peer-reviewed secondary studies that could inform future research

☆ Editor: Neil Ernst.

* Corresponding author.
E-mail addresses: lepebina@udsm.ac.tz (L.P. Binamungu), maro.salome@udsm.ac.tz (S. Maro).

https://doi.org/10.1016/j.jss.2023.111749
0164-1212/

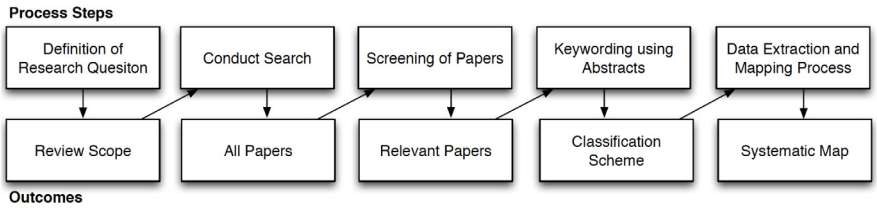

Fig. 1. Systematic mapping process (Petersen et al., 2008).

on BDD. Additionally, no peer-reviewed published study has attempted to create the landscape of BDD research. To fill this gap, we followed the guidelines by Petersen et al. (2015, 2008) to conduct a Systematic Mapping Study on BDD, to understand the types of venues in which BDD papers have been published, types of research and contributions in published BDD studies, focus and evolution of research on BDD, as well as evaluation methods that have been used in BDD studies.

We found that BDD research has been dominated by conference papers over journal and workshop papers. Also, the majority of BDD studies were found to have focused on three aspects: using BDD for facilitating various software development endeavours, improving the BDD process and associated artefacts, and applying BDD in different contexts. Moreover, there has been an increase in the diversity of studied BDD topics, and there is a notable use of case studies and experiments to evaluate the contributions of different BDD studies. However, there is a scarcity of BDD research that has been evaluated in industry settings. This leads to a lack of understanding of the applicability of the proposed solutions because of a lack of evidence from the software industry. Too, there is a scarcity of studies on using BDD alongside other software development techniques and technologies. Besides, this study has identified a shortage of philosophical papers on BDD, and an acute shortage of metrics for measuring various aspects of BDD, such as metrics for BDD specifications and metrics for processes of producing BDD specifications.

The rest of this paper is structured as follows. Section 2 describes the methodology we followed to search, filter and analyse the papers; Section 3 presents the results of the study; Section 4 discusses the results and their implications; Section 5 presents notable areas of future research based on the results of this study; Section 6 presents the threats to the validity of our results, and how the validity threats were mitigated; Section 7 presents the related work; and Section 8 concludes the paper.

2. Research methodology

To understand the studied BDD topics and provide a structure of the research field, a systematic mapping study is an appropriate research methodology. Systematic mapping studies provide a coarse-grained view of a specific research field by providing visual representations of the research topics as well as the results in the field (Petersen et al., 2008). This study was conducted by following the steps to conduct a systematic mapping study described by Petersen et al. (2008), as shown in Fig. 1. The details of how each step was conducted are provided in the following subsections.

2.1. Definition of research questions

The purpose of the present study is to analyse primary research papers on BDD in order to provide an understanding and structuring of different BDD aspects studied in the literature. To derive the research questions, two researchers conducted a brainstorming session, which aimed at getting a quick overview of existing research on BDD by looking at different papers and referring to existing systematic mapping studies in software engineering. The interest of this systematic mapping study is to understand the following aspects with respect to BDD research: timeline, frequency of publications, publication venues, research topics, methodologies used, and achieved results. To this end, the following research questions were formulated:

RQ1: What are the types of venues in which BDD studies have been published?

RQ2: Who is working on BDD research?

RQ3: What types of research have been conducted on BDD?

RQ4: What types of contributions have been made by existing research on BDD?

RQ5: What research evaluation methods have been used in BDD research?

RQ6: What themes have been covered in existing BDD research?

RQ7: How have the studied BDD topics evolved over time?

RQ8: How are the contributions in BDD research distributed with respect to research types and topics?

2.2. Conducting the search

In this step, we searched for scientific papers for use in the study. Five scientific databases were selected due to their relevance in indexing papers published in the field of software engineering and computer science in general (cf. Table 1). Database search is the most commonly used strategy for searching papers in systematic mapping studies in software engineering (Petersen et al., 2015). We also conducted snowballing to identify BDD papers that may not have been obtained through database search. Both forward and backward snowballing were conducted. To derive the search string, we scanned titles and abstracts of already known BDD research papers, as well as used the experience of the researchers to come up with keywords that are common in BDD-related research papers. The initial set of keywords was used to derive other keywords in form of synonyms or spellings in British English and American English. The following search string is an example of the search string that was used in the IEEE Xplore database. Since each scientific database uses a slightly different syntax – for instance, the need for brackets and double quotes – the semantics of the search string remained the same but syntactic variations of the search string were made to make it match the syntax of the different databases.

```
''Behaviour Driven Development'' OR ''Behavior
Driven Development'' OR ''Behavioural software
development'' OR ''Given when then''
```

Since our interest was to survey all papers on BDD, the search string is generic and therefore the number of papers found was also high (791).

Table 1
Number of studies found in each database.

Database	Search results
Scopus	188
ACM Digital Library	119
IEEEXplore	55
ScienceDirect	61
Springer Link	368
Total	791

2.3. Screening of papers

The screening of papers was done in three phases. In the first phase, we removed all duplicate papers and remained with 628 papers. We then excluded papers that were not written in English. Our search returned some papers which were in Russian and Portuguese and these were removed. This led to a total of 600 papers included.

In the second phase, we used a tool called Abstrackr (Byron et al., 2012) to structure and review all the remaining papers. Abstrackr is an online web-based tool that provides the functionality for uploading and organising search results (citation information and abstracts), allowing the reviewers to screen them in a collaborative manner. While Abstrackr also has a feature for automatically screening the abstracts for inclusion or exclusion, we did not use this feature to avoid the probability of introducing errors.

A pilot screening phase using Abstrackr was conducted, where 20 papers were reviewed by both authors, out of which the authors agreed on the judgements made on 14 papers. The six conflicts were discussed until a consensus was reached, and this improved our common understanding of how to further judge the papers in subsequent iterations. After the pilot stage and the discussion session, we divided the remaining papers into two, where one researcher reviewed 290 papers and the other 290 papers individually using the following inclusion and exclusion criteria.

Inclusion criteria:

1. The title and abstract show that the paper is about BDD
2. The paper reports a primary study
3. The paper is peer-reviewed

Exclusion criteria:

1. Papers mentioning BDD in the title or abstract but could not be considered as describing research on BDD
2. Papers presenting editorials and summaries of conferences
3. Papers published in non-peer-reviewed venues
4. Papers not accessible in full text
5. Papers that are duplicates of other studies
6. General book chapters that explain what BDD is
7. Papers of poor quality that lack sound methodology, clear statement of aims and/or have no contribution

For some papers, the decision for including or excluding was made based on the abstract, but for others, the abstract was not enough to make a decision, so the reviewers skimmed the full papers to get more information. While working individually, reviewers were allowed to include or exclude a paper if they were completely sure that it meets a certain inclusion or exclusion criteria; in case an individual reviewer was not sure whether to include a paper or not, they marked a paper as "maybe" for later discussion and consensus. Thus, after the individual screening, the two reviewers had another meeting to go through all the papers marked as "maybe" to decide on whether to include or exclude the papers. After this meeting, the final set of included papers was 147. This number also consists of the papers included during the pilot screening. To make sure that we did not miss any valuable papers, we conducted both forward and backward snowballing on all 147 included papers. The snowballing process led to an addition of 19 papers which were not captured by our initial search. Fig. 2 summarises the whole papers screening process.

Table 2
Research types (Petersen et al., 2015; Wieringa et al., 2006).

S/n	Research Type	Description
1	Validation Research	Techniques investigated are novel and have not yet been implemented in practice. Techniques used are for example experiments, i.e., work done in the lab.
2	Evaluation Research	Techniques are implemented in practice and an evaluation of the technique is conducted. That means, it is shown how the technique is implemented in practice (solution implementation) and the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation). This also includes identifying problems in the industry.
3	Solution Proposal	A solution for a problem is proposed, the solution can be either novel or a significant extension of an existing technique. The potential benefits and the applicability of the solution are shown by a small example or a good line of argumentation.
4	Philosophical Papers	These papers sketch a new way of looking at existing things by structuring the field in the form of a taxonomy or conceptual framework.
5	Opinion Papers	These papers express the personal opinion of somebody on whether a certain technique is good or bad, or how things should be done. They do not rely on related work and research methodologies.
6	Experience	Papers explain what has been done in practice, and how it has been done. It has to be the personal experience of the author.

2.4. Keywording using abstracts

To create the classification scheme of the included papers, four facets (research type, contribution type, research evaluation method and research theme) were selected. We selected these facets because categorising the papers using the facets enables us to answer the research questions posed in Section 2.1. These facets are explained next.

Research type: For each paper, we identified the type of research that the conducted study fits in. We adopted the research type classification by Wieringa et al. (2006), which is also the classification scheme recommended by Petersen et al. (2015). The classification scheme consists of six types of research which are validation research, evaluation research, solution proposal, philosophical papers, opinion papers and experience papers, as described in Table 2.

Contribution type: Similarly, for each of the included papers, we identified the contribution type of the reported study. In particular, we adopted the following five types of contributions from Mujtaba et al. (2008): process, method, model, tool, and metric. The five contribution types are described in Table 3. We devised a sixth contribution type called *Empirical Insights* to categorize studies whose contributions do not fall into any of the above five contribution types, but still report important empirical insights regarding BDD.

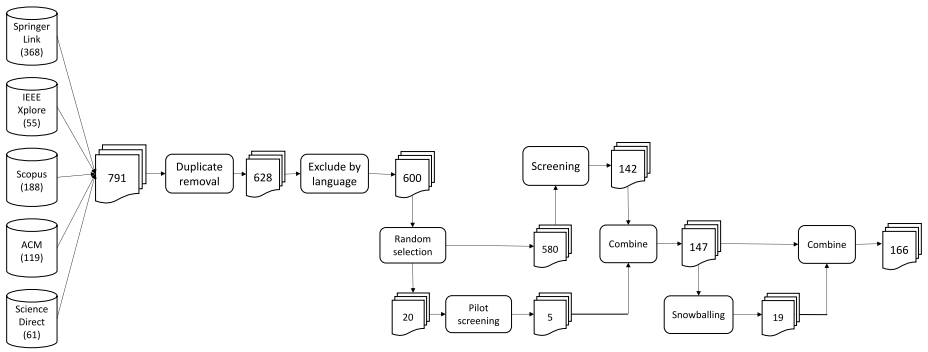

Fig. 2. The papers screening process.

Table 3
Contribution types (Mujtaba et al., 2008).

S/n	Contribution Type	Description
1	Process	Describes actions or activities, their associated workflows and artifacts
2	Method	Describes rules for doing things. It includes algorithms
3	Model	Describes the real world without giving details. It should be very formal with semantics and notations, e.g., UML models
4	Tool	A software tool has been developed to support different BDD aspects
5	Metric	Describes metrics and measurements for different aspects of a BDD process and the resulting specifications
6	Empirical insights	Describes lessons or experience of using BDD in practice. The lessons are not based on the evaluation of a process, model, method, tool, or metric. For example, a study that mainly reports the benefits and challenges of using BDD in a particular context (Nascimento et al., 2020) would be regarded to have contributed empirical insights.

Table 4
Types of evaluation methods (Chen and Babar, 2011).

S/n	Evaluation Method	Description
1	Rigorous Analysis	Involves thorough analysis, deriving, or proving certain aspects of a system, e.g., in formal models
2	Case Study	Uses multiple sources of evidence to investigate a phenomenon in its real world context, when there are unclear boundaries between a phenomenon and its context
3	Discussion	Evaluation is based on qualitative, textual, opinions, to compare and contrast some aspects of a proposed solution or discuss the upsides and downsides of a proposed solution
4	Example	An example is used to both explain what has been proposed by authors and demonstrate how good a proposal is
5	Experience Report	Using evidence collected formally or informally to demonstrate the use of a particular proposal, without using case studies or controlled experiments
6	Experiment	Testing an hypothesis through controlled experiments in industry or laboratory settings
7	Simulation	Artificial data have been used to execute a system, by modelling the real world

Research evaluation method: For each paper, we analysed how the contribution of the paper was evaluated. Specifically, we used the research evaluation methods classification scheme proposed by Chen and Babar (2011) in their systematic literature review (refer to Table 4).

Research theme: The research themes were derived in an inductive manner. The two researchers read the abstracts and screened the papers (where necessary) to identify thematic areas covered by the papers. Note that one paper can cover more than one theme.

To place the papers into specific categories, the two researchers had three workshops where 50 randomly selected included papers were discussed and the research type, contribution type, and research theme were decided. This was to make sure that both the researchers agree to the categories, especially in the research theme facet, since the categories were decided on in an inductive manner rather than a deductive manner. Categorising 50 papers together also meant that the researchers have a common understanding of the categories used. The remaining set of papers (97) was divided almost equally among the two researchers, and each researcher continued with the categorisation independently. After the categorisation of papers by individual researchers, a meeting of researchers was held to discuss the categorisation results and to resolve any challenges encountered by individual researchers when assigning themes to papers.

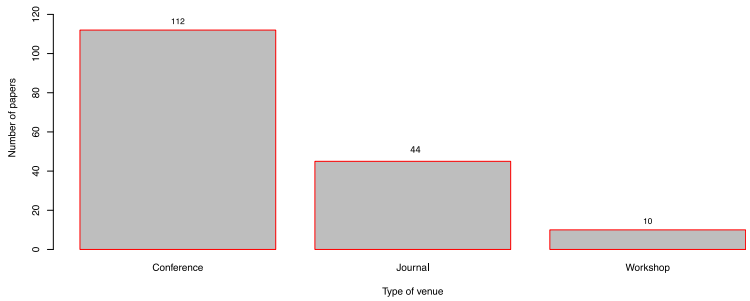

Fig. 3. Distribution of publications by venue.

2.5. Data extraction and mapping process

The categorisation of each paper was done in an Excel file where, for each paper, we marked the following: research type, research theme, contribution type, and evaluation method from the key-wording stage. Additionally, for each paper, we marked the year of publication, all authors, as well as the publication type (journal, book chapter, conference or workshop). The frequencies of the publications in each of these categories were calculated with the help of Excel formulas.

3. Results

We now present the results and discuss them in Section 4.

3.1. RQ1: Types of venues that have published BDD papers

Fig. 3 shows the number of publications in different venues. It is observed that most papers were published in conferences, followed by journals and workshops. Further analysis of the conference papers shows that the papers were published in a variety of conferences. As seen in Table 5, the study found that the conference papers were published in 89 different conferences. This shows that apart from the International Conference on Agile Software Development (formally called XP) which has 7 papers and the Brazilian Symposium on Software Engineering which has 5 papers, the rest of the conferences have less than 5 papers on BDD over the years (cf. Table 5). A similar trend is observed in the journal publications, the 44 journal papers are scattered in 38 different journals. Out of the 38 journals, Information and Software Technology (IST) has three papers and the Journal of Systems and Software (JSS) has three papers, while the rest of the journals have one paper each (cf. Table 6). Additionally, all workshop papers are also published in distinct workshops.

3.2. RQ2: Researchers who are working on BDD

We analysed all 166 papers to identify the authors, with the aim of identifying top researchers in the BDD area. There are a total of 640 unique authors and co-authors. However, further analysis shows that most authors only have one or two publications in the field. The top ten authors in terms of publication frequency are shown in Fig. 4, where the author with most publications has published 10 of the included papers and there are 11 authors in the 8th position who have published 3 papers. Note that this analysis includes both first authors and co-authors.

Table 5
Distribution of papers by conferences.

Conference	No. of papers
Agile Software Development (XP)	7
Brazilian Symposium on Software Engineering	5
Model Driven Engineering Languages and Systems (MODELS)	4
Iberian Conference on Information Systems and Technologies	3
Computational Science and Its Applications	2
Conceptual Modelling	2
Information Technology - New Generations	2
Requirements Engineering	2
Requirements Engineering: Foundation for Software Quality	2
Brazilian Symposium on Systematic and Automated Software Testing	2
Symposium on Engineering Interactive Computing Systems	2
International Conference on the Quality of Information and Communications Technology	2
Other conferences	77

Table 6
Distribution of papers by journals.

Journal	No. of papers
Information and Software Technology	3
Journal of Systems and Software	3
Other journals	38

3.3. RQ3: Types of conducted BDD research

Fig. 5 shows the distribution of different research types conducted about BDD. On the one hand, there are more solution proposals and validation research than evaluation research. On the other hand, there are a few papers reporting the experience of using BDD in practice. Also, philosophical and opinion papers are almost non-existent. Table A.8 summarises the studies under each research type.

3.4. RQ4: Types of contributions in conducted BDD research

Fig. 6 shows the distribution of different contributions made by studies on BDD. In general, BDD studies have proposed more processes, tools, and empirical insights than methods, models, and metrics. Table A.9 summarises the studies under each contribution type.

3.5. RQ5: Research evaluation methods used

Fig. 7 shows the different research evaluation methods that have been used in BDD research. Because some papers have used more than one evaluation method, the total in Fig. 7 is more than the total number of papers analysed in the present study. Also, it

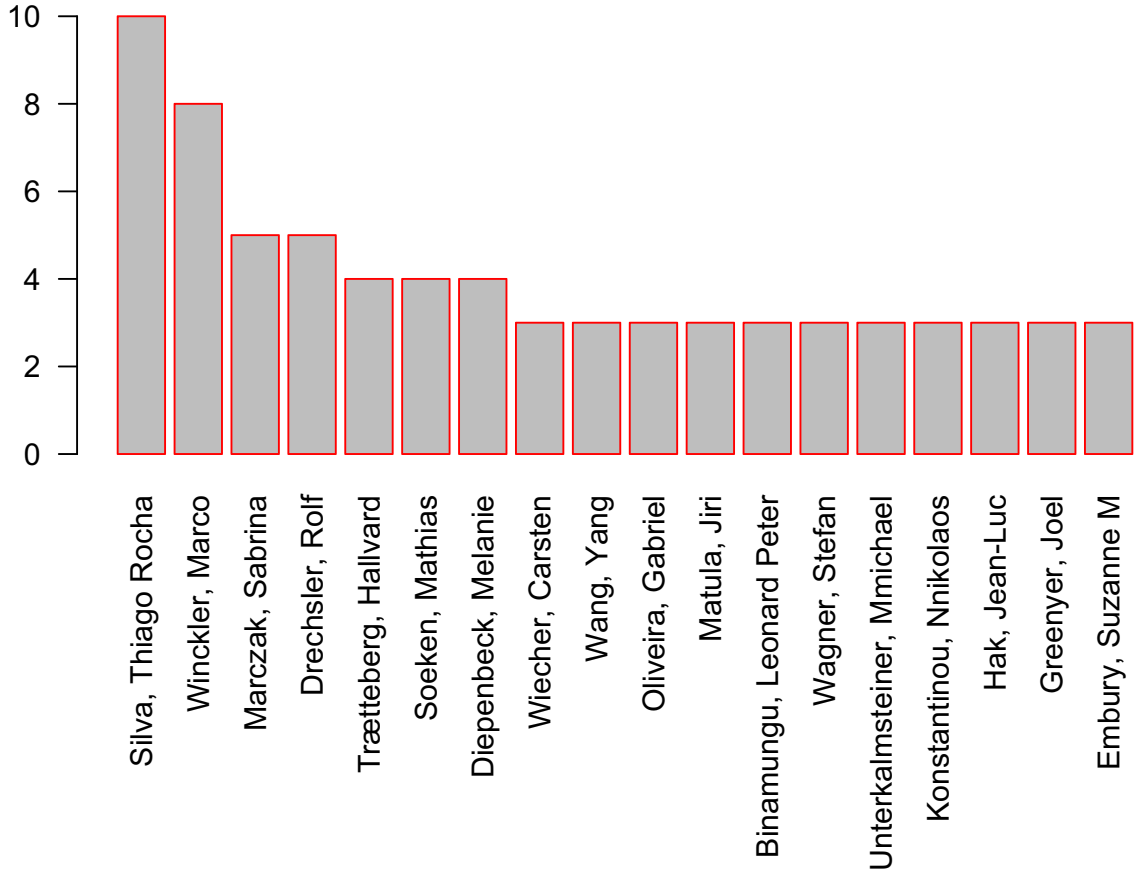

Fig. 4. Top authors in the BDD research area.

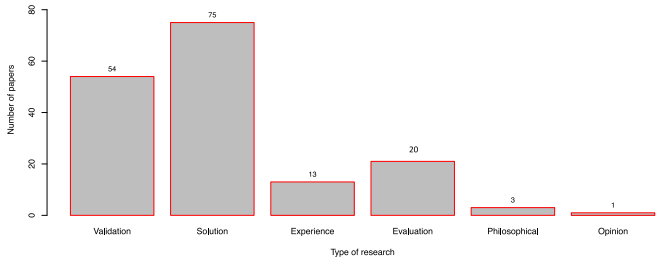

Fig. 5. Distribution of types of conducted BDD research.

can be seen in Fig. 7 that most studies have used example applications or case studies to demonstrate the applicability of solutions or insights generated in BDD research. Table A.10 summarises the studies under each evaluation method.

3.6. RQ6: Themes in existing BDD research

Our systematic mapping study identified six major themes, which are presented in Fig. 8 and described in this subsection. Each major theme had several sub-themes which are more specific, and the papers are therefore categorised using sub-themes. The themes and sub-themes are summarised in Table 7, while the studies under each theme are presented in Table A.11. It can be seen that BDD research has mainly focused on how BDD is used to facilitate various software engineering aspects, improving the BDD process and the resulting artefacts, and on how BDD has been used in different application areas. Studies on using ontologies, BDD alongside other processes and BDD with other technologies are still on the lower side.

Theme 1: Facilitating software development

In most studies, BDD has been used to facilitate different aspects of a software or software engineering process. In particular, BDD has been used to facilitate the following: safety of software systems (Wang and Wagner, 2018a,b; Baillon and Bouchez-Mongardé, 2010; Wiecher et al., 2020; Wang et al., 2018), testing of graphical user interfaces (GUI) (Silva et al., 2017b; Bünder and Kuchen, 2019a; Sivanandan et al., 2014; Schur and Williams, 2017; Bahaweres et al., 2020; Rocha Silva et al., 2019a,b; Bünder and Kuchen, 2019b), testing of spreadsheets (Almeida et al., 2016), product line engineering (Elshandidy, 2019; Bagheri and

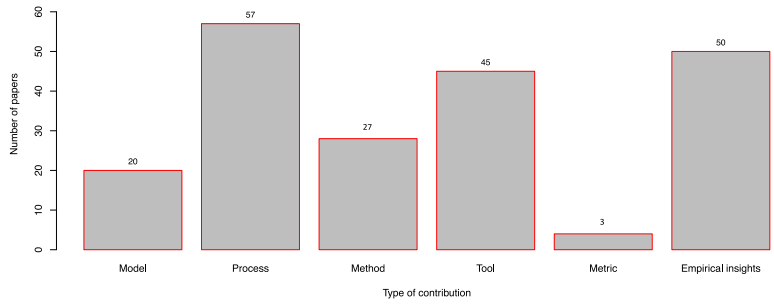

Fig. 6. Distribution of types of contributions in conducted BDD research.

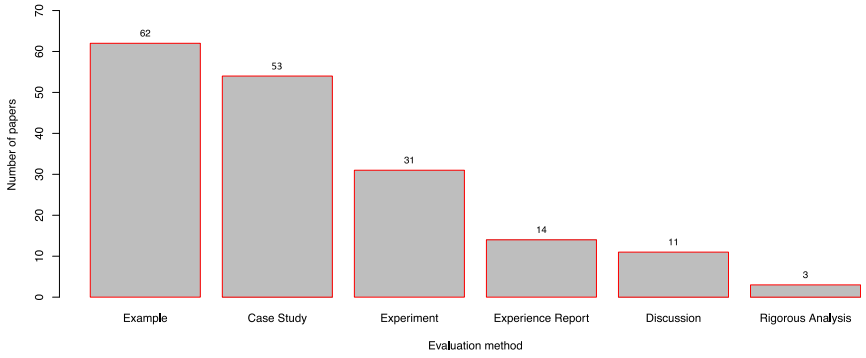

Fig. 7. Evaluation methods used in BDD research.

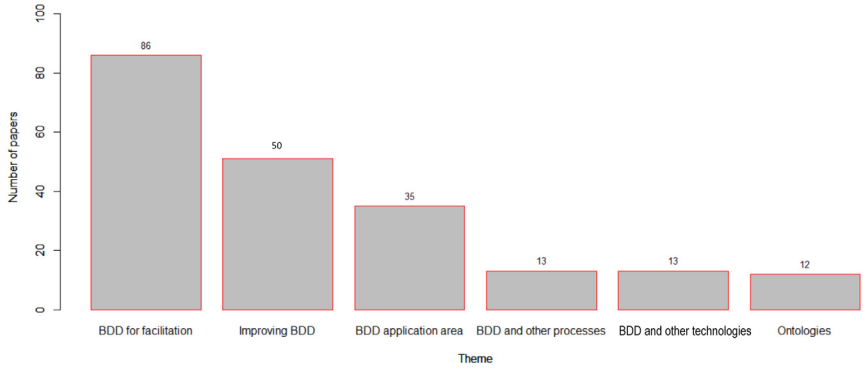

Fig. 8. Distribution of themes in BDD research.

Ensan, 2013; Elshandidy et al., 2021), compliance testing (Lopez-Pellicer et al., 2014; Moult and Krijnen, 2020; Williams et al., 2020; Morrison et al., 2013a,b; Pang Fung et al., 2021), traceability of software artefacts (Lucassen et al., 2017; Silva and Fitzgerald, 2021b; Yang et al., 2019), model testing (Mens et al., 2019; Snook et al., 2018, 2021; Silva and Fitzgerald, 2021a; Rocha Silva et al., 2019a; Silva et al., 2019; Fischer and Dghyam, 2019), business process modelling (de Carvalho et al., 2012; Carvalho et al., 2013; Lübke and van Lessen, 2016; Matula and Zacek, 2018b), test selection (Xu et al., 2021), usability testing (Güncan and Onay Durdu, 2021), security and privacy of software systems (Lai et al., 2014; Purkayastha et al., 2020; Okubo et al., 2014), model-based testing (Sivanandan et al., 2014; Li et al., 2016), automatic generation of different software artefacts, including various artefacts of a BDD project (Wanderley and da Silveria, 2012; Bünder and Kuchen, 2019a; Dimanidis et al., 2018; Soeken et al., 2012; Schur and Williams, 2017; Nguyen et al., 2020; Kamalakar et al., 2013; Malik et al., 2021; Carrera et al., 2014; Storer and Bob, 2019; Nezhad et al., 2018; Lazăr et al., 2010; Snook et al., 2018; Carter and Gardner, 2016a,a; Deng et al., 2021; Alferez et al., 2019; Gupta et al., 2019; Pandit et al., 2016; Silva and Fitzgerald, 2021a; Fazzolino and Rodrigues, 2019; Gao and Li, 2016; Bonfanti et al., 2018; Schoeneman and Liu, 2013; King et al., 2014; Williams et al., 2020; Nguyen et al., 2020; Li et al., 2016; Mahalakshmi and Vani, 2017; O'Brien, 2018; Diepenbeck et al., 2013; Bünder and Kuchen, 2019b; Pinto et al., 2019; Siqueira et al., 2017), integration testing (Purkayastha et al., 2020; Bussenot et al., 2018), load testing (Schulz et al., 2019), syntax highlighting (Matula and Zacek, 2018a), and reproducing bugs (Karagöz and Sözer, 2017).

Table 7
Themes in existing BDD research.

Sn	Theme	Sub-theme(s)
1	Facilitating software development	Safety GUI testing Artifact generation Product lines Compliance testing Traceability Model testing Business process modelling Security and privacy Model-based testing Integration testing Load testing Syntax highlighting Reproducing bugs Spreadsheets testing Test selection Usability testing Software documentation
2	Improving BDD	Specification quality BDD adoption Co-evolution of artifacts Duplicate detection and refactoring BDD characteristics Specification language Test cases as requirements BDD tools
3	BDD application areas	Hardware context Health context Education context Mobile apps Enterprise systems BDD and computer networking Product configuration systems Multi-agent systems
4	Ontologies	Ontologies
	BDD and other processes	Comparison with other techniques BDD and Scrum BDD and DevOps
6	BDD and other technologies	Machine Learning Blockchain Microservices Web services

Theme 2: Improving the BDD process and the resulting artefacts

Another substantial strand of research on BDD has been focusing on improving various aspects of the BDD process and the resulting artefacts. Specifically, studies on improving BDD have covered the following aspects: assessing and improving the quality of BDD specifications (Binamungu et al., 2020; Oliveira et al., 2019; Binamungu et al., 2018b; Oliveira and Marczak, 2017, 2018; Binamungu et al., 2018a), ensuring co-evolution of BDD artifacts (Sathawornwichit and Hosono, 2012; Zampetti et al., 2020; Santos et al., 2019; Yang et al., 2019; Rocha et al., 2019), adoption of BDD by software teams (Irshad et al., 2021; Contan et al., 2017; Bezsmertnyi et al., 2020; Gil et al., 2016; Scandaroli et al., 2019; Pereira et al., 2018; Shafiee et al., 2018; Nascimento et al., 2020; Zampetti et al., 2020; Binamungu et al., 2018b; Mello et al., 2018; Cavalcante and Sales, 2019; Barus, 2019), characteristics of BDD (Solis and Wang, 2011), duplicate detection and refactoring of BDD specifications (Binamungu et al., 2018a), developing a software specification language (Kudo et al., 2019; Lucena, 2017; Lazăr et al., 2010; Deng et al., 2021; Snook et al., 2021; Rocha Silva et al., 2020; Li et al., 2017; Häser et al., 2016; King et al., 2014; Hesenius et al., 2014; Pang Fung et al., 2021; Bünder and Kuchen, 2019b), using test cases as requirements (Bjarnason et al., 2016, 2015; Elshandidy et al., 2021), and BDD tools (Okolnychyi and Fögen, 2016; Pyshkin et al., 2012).

Theme 3: Application of BDD in different settings

Different studies have also reported the application of BDD in different areas/contexts. Existing literature shows that BDD has been particularly used in the following settings: development of multi-agent systems (Carrera et al., 2014), development and testing of systems that have hardware components (hardware context) (Kannengiesser et al., 2020; Alhaj et al., 2019; Zaeske et al., 2021; Diepenbeck et al., 2012; Nezhad et al., 2018; Diepenbeck et al., 2018, 2014; Deng et al., 2021; Mwakyanjala et al., 2020; Winkler et al., 2019; Bussenot et al., 2018; Zafar et al., 2021; Alhaj et al., 2017; Cauchi et al., 2016), use of BDD to develop software systems for use in the health sector (health context) (Hatko et al., 2014; Purkayastha et al., 2020; Mello et al., 2018; Morrison et al., 2013a,b; Giorgi and Paulisch, 2019), use of BDD to teach software engineering or teaching BDD in software engineering classes (education context) (Sarinho, 2019; Nascimento et al., 2020; Rocha et al., 2021), facilitating computer networking (Esposito et al., 2018), developing product configuration systems (Shafiee et al., 2018), development and testing of mobile applications (Nguyen et al., 2020; Ali et al., 2019), and development of enterprise systems (Fazzolino et al., 2018; Gohil et al., 2011).

Theme 4: Ontologies

Another theme that has been identified is on BDD in combination with ontologies. Specifically, this has been studied in

two aspects; first, developing an ontology that formalises BDD vocabularies that are used in user stories, scenarios and GUI interactions (Silva et al., 2017a). The ontology can be used to build model-driven tools that enable the modelling of BDD scenarios. The existence of such a model also makes it possible to evaluate the consistency of scenarios and other GUI-related artefacts such as GUI prototypes and task models (Silva et al., 2017b; Rocha Silva et al., 2019b, 2020a, 2019a; Silva et al., 2016). Second, BDD has been used in combination with ontologies of different domains with the aim of reducing the ambiguity of defining BDD scenarios using natural language. Our study found that BDD has been used with an ontology in the higher education domain (Souza et al., 2018; Lopes de Souza et al., 2021) and an ontology in the business information systems domain (Matula and Hunka, 2018; Matula and Zacek, 2018a). Similarly, the presence of domain ontologies used in BDD scenarios facilitates consistency checking between the functionality of the system and business-related tasks (Matula and Zacek, 2018b).

Theme 5: BDD and other processes

BDD has also been used alongside other software engineering processes. Specifically, BDD has been compared with other software development techniques (Santos et al., 2015; Oran et al., 2017; Dookhun and Nagowah, 2019; Santos and Vilain, 2018; Bezsmertnyi et al., 2020; Manuaba, 2019) and combined with other software development methods such as Scrum (de Souza et al., 2017; Souza et al., 2018; Lopes de Souza et al., 2021) and DevOps (Zaeske and Durak, 2020; Giorgi and Paulisch, 2019), to generate lessons for informing better software development.

Theme 6: BDD and other technologies

BDD has been studied by integrating it with other technologies. Precisely, lessons have been generated when BDD was used alongside machine learning technology from two perspectives. First, how machine learning can enhance the use of BDD artefacts, for instance by recommending micro-services from a service catalogue based on selected scenarios (Ma et al., 2018). Second, how BDD can facilitate the development of machine learning applications, for instance by using BDD scenarios to specify accountability requirements for machine learning algorithms (Pang Fung et al., 2021). A few studies have also been identified with respect to microservices technology where BDD scenarios have been used to select relevant microservices (Ma et al., 2018, 2019), automate acceptance testing of microservices by executing appropriate scenarios (Rahman and Gao, 2015) and using BDD scenarios to solicit crowdsourced microservices (Aghayi et al., 2021). Two studies have also been found where BDD is used to facilitate the development and testing of web services (Dimanidis et al., 2018; Oruç and Ovatman, 2016). One unique study discussed the use of BDD to support the development of blockchain applications (Liao et al., 2017).

3.7. RQ7: Evolution of topics over time

We start by showing the frequency of BDD papers published over the years, and, thereafter, we show how the themes in those papers have evolved over time.

3.7.1. Publication frequency over the years

Fig. 9 shows the frequency of publication over the years. These results show that, after the introduction of BDD through a magazine article in 2006 (North, 2006), there has been an increase in the number of publications on BDD; a peak was reached in 2019, when 32 papers were published. In particular, a notable increase in the number of BDD papers was observed from 2016 to 2021.

3.7.2. Evolution of studied BDD topics over the years

Fig. 10 shows how the themes in BDD research (Table A.11) have changed over the years. In general, the number and diversity of studied topics in published BDD papers have been increasing over time. Similar to the number of papers, more diversity in the studied topics was mainly observed from 2016 to 2021, in which almost all themes were studied each year. More specifically, the results in Fig. 10 show that there has been a consistent and increasing interest in studying the use of BDD to facilitate various aspects of software engineering. This can be seen in all years but 2015. Generally, the papers on how BDD facilitates various aspects of software engineering have increased over time from 2009 to 2019, but seem to have reduced in 2020 and 2021. A similar trend is observed with studies on improving the actual BDD process. Studies on using BDD in different application areas have also increased over time and, on a deeper analysis, the diversity of domains has also increased. In 2015, we see new themes such as studying BDD alongside other software development processes and with different technologies. In 2016, the first paper on BDD and ontologies was published. However, the number of papers in these themes has not picked up over the years like papers in the other themes.

3.8. RQ8: Distribution of contributions w.r.t research types and topics

Fig. 11 shows the map that combines research types, contribution types, and themes in BDD research.

Generally, research on BDD has been dominated by studies in which the research type is solution proposal and the contribution type is process. Second in amount are solution proposals that have been accompanied by tools. Examples of these tools are tools for modelling scenarios (Wiecher et al., 2021b), ensuring consistency between scenarios and GUI prototypes (Silva et al., 2019) and code generation from scenarios (Storer and Bob, 2019). Apart from solution proposals, validation research is another research type that dominates existing research on BDD. Specifically, dominant contributions in the conducted validation research are of type process, empirical insights, method, and tool. Thus, there is a scarcity of models and metrics in the conducted validation research. Notably, there is a scarcity of BDD research that has been evaluated in industry settings, as well as philosophical and experience papers for all types of contributions.

Moreover, based on Fig. 11, the following can be noted regarding themes and contribution types. Most BDD studies have focused on producing tools, processes, methods, empirical insights, and models for facilitating software engineering processes. As regards improving the BDD process and the resulting artefacts, it can be seen that the contributions in most of the research on improving BDD are empirical insights and tools. Studies on the application of BDD in different settings have mostly contributed processes, empirical insights and tools, albeit to a relatively small extent. There is, however, a considerable lack of models, metrics and methods to inform the use of BDD in different settings. Also, most ontologies in existing BDD research have been accompanied by models, processes, methods, and tools. Nevertheless, the proposed ontologies have been accompanied by minimal empirical insights. This suggests the need for more BDD-related ontologies that are supported by strong empirical evidence. As well, when BDD was studied alongside other software engineering processes, the studies contributed more empirical insights and processes than models, methods and metrics. Finally, fewer processes, methods and tools were contributed when BDD was studied by integrating it with other technologies.

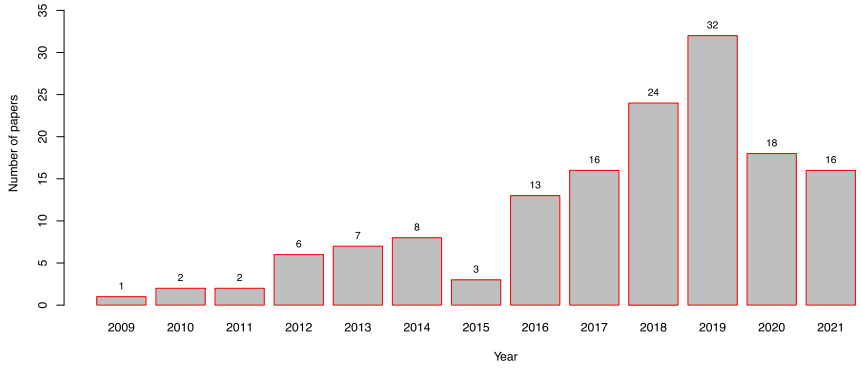

Fig. 9. Frequency of publication over the years (2009–2021).

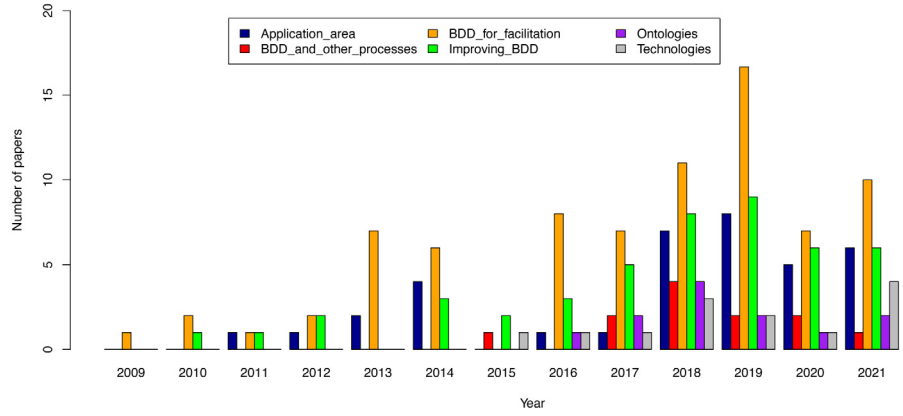

Fig. 10. Trend of themes over the years.

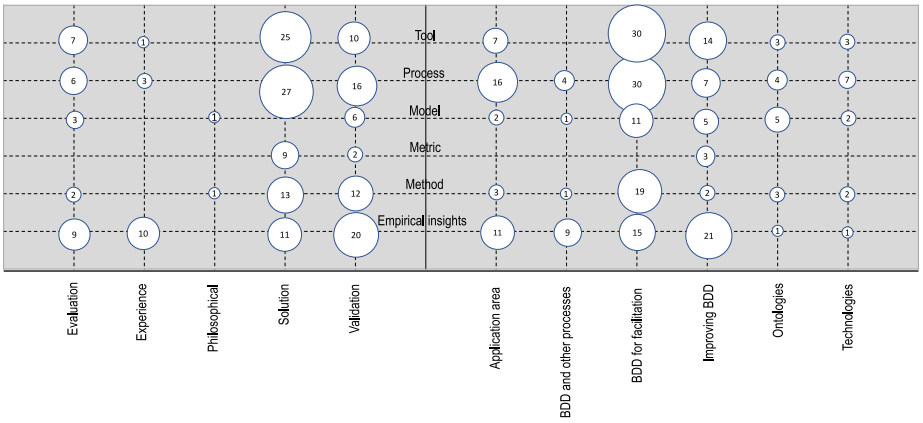

Fig. 11. Map of research types, themes, and contribution types.

4. Discussion

4.1. Publication venues for BDD papers

The publications about BDD are scattered in different journals, conferences and workshops. This suggests that BDD papers are not concentrated in specific journals, conferences or workshops, and implies that there are very few specific venues that publish BDD research. The International Conference on Agile Software Development has published 7 papers over 10 years, making it one of the popular venues for BDD research. Moreover, our analysis shows that BDD research publication venues, even in the form of workshops, do not exist yet. Therefore, we call upon the software engineering research community to establish BDD-specific publication venues to consolidate the BDD research. Doing so will easily inform both the state-of-the-art and the state-of-the-practice.

Additionally, referring to Fig. 3, it can be seen that the majority of BDD research papers have been published in conference proceedings. Although there are very strong software engineering conferences such as the International Conference on Software Engineering (ICSE), generally, conference papers have space limitations, which restrict the amount of content in papers published in conference proceedings. The scarcity of journal papers means that most research on BDD has been published in brief as part of conference and workshop proceedings, which could lack substantial and elaborate insights of interest to both researchers and practitioners. As noted by Montesi and Lago (2008), journal publications allow extended discussion of research work with little worry about the limitations of space and time. We, therefore, encourage the software engineering research community to produce more journal publications with rich insights on BDD research to better inform research and practice.

4.2. Researchers on BDD

The results show that there are approximately 600 unique authors and co-authors in the included set of papers. However, most of these researchers have only published a single paper or two in the area. The diverse nature of papers found by the present study can imply that researchers have other fields that they are working on and the BDD papers they have published have been opportunistic papers since BDD as a process is applicable in many domains. For instance, papers under the BDD application areas theme are published by researchers who are working on different domains and not necessarily software engineering.

4.3. Types of BDD research

The dominance of solution proposals and validation research over evaluation research (Fig. 5 and Table A.8) means that most of the BDD research either ended up proposing solutions for specific software engineering problems or went as far as a lab evaluation of the proposed solutions. The scarcity of studies that have been evaluated in the industry could affect the applicability of the proposed solutions because of the lack of evidence to demonstrate their effectiveness, when there is a general expectation that software engineering research should solve problems faced by practitioners (Pfleeger and Glass, 1994). Moreover, the fact that there are a few papers reporting the experience of using BDD in practice implies that more BDD experience reports are required, so that, the software engineering research community can understand BDD aspects that are important in practice, how the important BDD aspects are used, and the results they give (Klotins et al., 2019). This will, in turn, enable BDD researchers to prioritize aspects that matter to BDD practitioners. Thus, BDD practitioners need to report more on their experiences of working with BDD. In addition, more BDD philosophical papers are required to properly guide BDD research and practice. For example, papers on the theoretical foundations of BDD could enable the software engineering community to better understand how BDD is related to well-established software engineering theories. Finally, another lacking type of research is opinion papers. Since opinion papers present personal reflections on whether a certain method, technique or tool is good or bad, the lack of such studies means that the research community does not know how practitioners as well as academics feel about certain aspects of BDD. However, it is possible that such opinions are channelled through non-peer-reviewed venues such as blogs, because it is common for researchers and practitioners to air their opinions through blogs rather than scientific peer-reviewed venues. Therefore, conducting multi-vocal literature reviews may be valuable to uncover personal opinions on the BDD topic.

4.4. Contribution types in BDD research

The results in Fig. 6 and Table A.9 indicate that, while there is an arguably significant amount of other types of contributions, there is an acute shortage of studies on metrics for measuring various aspects of BDD specifications and the processes for producing BDD specifications. Out of the three studies that proposed metrics, two focus on measuring the quality of individual scenarios in a BDD specification (Oliveira et al., 2019; Oliveira and Marczak, 2017), and one focuses on measuring the overall quality of a BDD specification (Binamungu et al., 2020). However, there are no studies that have proposed metrics for measuring other aspects of the BDD technique and the resulting specifications—e.g., the quality of a BDD workflow used by a software team. The lack of studies on metrics on different aspects of BDD indicates that practitioners lack ways to measure the quality of the BDD process and its associated aspects, and ways to measure the impact and value that BDD brings.

4.5. Evaluation methods used in BDD research

Although some BDD researchers have employed rigorous evaluation methods such as case studies or experiments to study various aspects of BDD, most of the BDD studies have used example applications to demonstrate the applicability of solutions or insights generated in BDD research (Fig. 7 and Table A.10). This implies that a substantial proportion of BDD research is immature, because, as observed by Chen and Babar (2011), example applications, experience reports, and discussions are evaluation methods that cannot be considered to be scientifically rigorous. BDD researchers should, therefore, invest more efforts in rigorous empirical evaluation methods to increase the chances of transfer of research results into practice (Dyba et al., 2005; Chen and Babar, 2011). In addition, as mentioned in Section 3.3, there is a paucity of studies reporting experiences of using BDD in industry, which limits the understanding of the software engineering research community on how BDD is used in practice in order to glean lessons that could inform future BDD research. Moreover, only three studies (Lucassen et al., 2017; Diepenbeck et al., 2018, 2014) have performed rigorous analysis through formal methods. Specifically, formal methods in BDD have been used for hardware testing and verification (Diepenbeck et al., 2018, 2014) and for tracing software requirements (Lucassen et al., 2017). More studies focusing on formal aspects of BDD are required to reap the benefits of formal methods in software engineering, which include producing properly abstracted and unambiguous software specifications (Plat et al., 1992).

4.6. Evolution of BDD research over time

BDD research has evolved over time and space, producing more publications that cover a wide range of topics (Figs. 9 and 10). This shows that various aspects of BDD have been attracting the attention of the software engineering research community. More specifically, initial research on BDD was about using BDD to facilitate software documentation (Brolund, 2009). However, over the years, apart from paying attention to how BDD facilitates other software development endeavours, BDD researchers have also focused on other topics (Fig. 10 and Table A.11). With respect to the application context, in 2011, BDD was first studied in the context of enterprise systems (Gohil et al., 2011), but similar studies have been conducted to cover other specific contexts such as health (Hatko et al., 2014; Purkayastha et al., 2020; Mello et al., 2018; Morrison et al., 2013a,b; Giorgi and Paulisch, 2019; Bruschi et al., 2019), education (Sarinho, 2019; Nascimento et al., 2020; Rocha et al., 2021; Goulart, 2014), and others (Fig. 10 and Table A.11). This shows that BDD has a potential to be adopted in any domain where software is used. Moreover, studies on using BDD alongside other software development processes, techniques, and technologies started a bit later (2015 onwards), but the popularity of such studies is still relatively low compared to studies on other BDD topics (Fig. 10). Among other things, this limits our ability to understand how the combination of BDD and other techniques and technologies such as scrum, microservices, machine learning, and blockchain can produce better solutions for different software engineering problems. Nevertheless, the noted evolution and diversity of the studied BDD topics over time suggest that more new and diverse BDD topics could be studied in the future, improving our understanding of the use of BDD in software development.

Importantly, apart from paying attention to other BDD topics discussed in the preceding paragraph, BDD researchers have had a sustained interest in improving the BDD process and the artefacts produced by the BDD process (Fig. 10). This implies that the BDD process and the resulting artefacts could get better in the future, improving the way practitioners use BDD to develop and maintain software systems. However, as we indicate in Section 5, further research on the studied topics is required to produce more mature insights to guide the work of BDD practitioners.

4.7. Distribution of contributions w.r.t research types and topics

The finding that BDD research has been dominated by studies in which the research type is solution proposal and the contribution type is process (Fig. 11) implies that most BDD studies have proposed processes as well as associated artefacts that act as solutions for software engineering problems. This is not a surprising finding due to the following explanation. As can be seen from many papers reported in the present study (see, for example, Fig. 8), research on BDD has been focusing primarily on the artefacts (the BDD stories) resulting from the BDD process. One of the hypotheses could be that the BDD process is quite similar to the TDD process (which is already well known and studied), so the novelty brought by BDD lies more in the notation for specifying the scenarios (the artefacts), as it opens many opportunities for automation and support of other software development activities.

In addition to the paucity of specific types of contributions such as metrics, there is a notable scarcity of BDD research that has been evaluated in industry settings, as can be deduced from the small amount of evaluation research of all types of contributions. This leads to a lack of understanding of the applicability of the proposed solutions. Finally, the scarcity of philosophical and experience papers for all types of contributions could pose different challenges, including the following two. One, there is a possibility of having BDD research that is not properly guided (by appropriate theoretical foundations, which could be obtained from philosophical papers). Two, due to the scarcity of experience reports, in which lessons about the use of BDD in practice are shared, software teams could improperly practice BDD, failing to reap the envisaged benefits.

5. Future research

In addition to the research opportunities that can be deduced from the discussion of results (Section 4), we now discuss some key opportunities for BDD research, based on the results of the present study.

First, given a relatively small number of journal papers compared to conference papers (Fig. 3), there is a need for BDD researchers to focus more on producing mature and substantial studies, publishable by journals, to better inform the BDD state-of-art and state-of-practice. Although some software engineering conferences publish mature work, almost all of them have space limitations, hindering the ability of most authors to report in full and extensively. This might, in turn, limit the understanding and application of knowledge in conference and workshop papers. Although also limited, the space provided by software engineering journals is usually higher than the space provided by software engineering conferences. So, more journal publications about BDD would give a clear understanding of research achievements in the area.

Second, based on the scarcity of studies that have been evaluated in the industry (Fig. 5 and Fig. 11), there is a need for more empirical evidence and insights which are informed by actual practices in the software industry. This will increase the chances of applicability of the proposed solutions because of reliable evidence to demonstrate their (solutions) effectiveness. Moreover, there is a need for more experience reports on BDD, to share best practices that can inform the user and research community about how BDD is actually used in the industry. In addition, future research should consider philosophical aspects of BDD, to better guide this study field. Among others things, philosophical studies could include theories, frameworks and taxonomies on various aspects of the BDD technique.

Third, based on the distribution of different contributions in existing BDD research (Fig. 6 and Fig. 11), future studies should give due attention to metrics for measuring various aspects of BDD specifications and the processes for producing BDD specifications. Future research should also put more emphasis on developing models and methods to address different facets of BDD.

Fourth, the results in Fig. 11 suggest the need for more processes, models, methods, and metrics that aim to improve the BDD process and associated artefacts. This is crucial for the development of the BDD research field since most studies have focused on the use of BDD in different contexts and to facilitate different activities, but few have focused on how core aspects of BDD such as scenario specifications can be improved. Fifth, the results in Fig. 11 indicate that, apart from the need of developing more BDD-related ontologies, there is a need for BDD-related ontologies that are accompanied by empirical insights. This will influence their use in industry settings.

Sixth, there is a need for more research that studies BDD alongside other software processes and technologies (refer to Fig. 11 and Fig. 10). Specifically, since BDD is an Agile technique, studies on how it can be used with specific Agile methods such as Scrum and Kanban will be interesting for practitioners who already use Agile. Additionally, studies on how BDD can be enhanced with emerging technologies such as machine learning are needed to generate lessons that could inform efficient software development.

6. Threats to validity

6.1. Identification of primary studies

There could be biases in the identification of the primary studies, for two reasons: first, there is a chance that the search string could have excluded relevant studies; second, reviewers could be biased in excluding papers. To mitigate the effects of these threats, the search string (Section 2.2) was formulated to be as generic as possible to avoid missing any BDD-related publications. Also, as described in Section 2.2, snowballing was conducted to account for papers that might have been missed through a database search. However, even with snowballing, there is still a chance that some papers might have been missed in our search. This is tolerable because, as noted by Petersen et al. (2015), different from a systematic literature review, which aims to capture all appropriate evidence on a particular research question, the aim of a systematic mapping study is to give a good general overview of a research area.

Moreover, since the majority of the papers were initially screened by individual researchers, there is a chance that relevant papers could have been excluded. To mitigate the effects of this threat, there was a pilot stage, in which 20 papers were reviewed by both researchers to agree on when to exclude or include papers. In addition, individual researchers could only mark a paper as excluded when they were completely sure that a paper meets a specific exclusion criterion; all papers with doubts were marked as "maybe" and were later discussed by both researchers in a workshop.

6.2. Data extraction and mapping

To mitigate biases during data extraction and mapping, two strategies were used. First, to categorise a paper into a specific research type, contribution type, or research evaluation method, we used established classification schemes of Wieringa et al. (2006), Mujtaba et al. (2008), and Chen and Babar (2011) respectively. Second, to categorise a paper to a specific theme, an inductive approach was used, where themes would emerge as reviewers proceeded with the categorisation process. With this approach, there is a chance that different reviewers would assign different themes to the same paper. To mitigate the effects of this threat, we conducted a pilot workshop, in which 50 papers were randomly selected and both reviewers did the categorisation together. Each emerging theme was thoroughly discussed for both reviewers to agree on papers that would fit that theme. The rest of the papers were categorised by individual reviewers; however, a workshop was held afterward to discuss papers on which reviewers had doubts regarding the themes to which they (reviewers) had categorised the papers, as well as to discuss and harmonise new themes that had emerged during individual work.

7. Related work

We found three secondary studies that are related to BDD. One of these studies has focused on BDD alongside TDD, while the other two studies have focused on BDD only. Abushama et al. (2020) conducted a systematic literature review to understand the effect of TDD and BDD on project success by focusing on time, cost, and customer satisfaction. BDD was found to be more likely to achieve higher customer satisfaction than TDD, and TDD consumed more development time than BDD. However, both BDD and TDD were found to consume more cost and time than traditional approaches to software testing.

The first of the two secondary studies that focused on BDD only was reported in 2017 by Egbreghts (2017), who conducted a systematic literature review in order to describe what BDD is and how it is used in practice. In 2020, Lillnor and He (2020) reported a systematic mapping study on BDD, which was conducted as part of a bachelor thesis. This is the second secondary study on BDD. However, both these studies were conducted as student projects and are only partially reported (only abstracts are accessible) in the grey literature; none of them is reported in peer-reviewed scientific literature. This poses concerns about the scientific rigour and coverage of these studies. We, thus, set out to conduct a systematic mapping study that can provide a reliable help to the scientific community in understanding the current state of research on BDD and the gaps in the existing literature, to inform future research on BDD.

8. Conclusion

BDD enables software teams to specify software requirements based on a structured natural language format, which facilitates understanding of software requirements among all software project stakeholders. The specifications in a natural language also act as test cases that can be executed to verify the behaviour of a software. Since the introduction of BDD about two decades ago, the software engineering research community has been investigating various aspects of BDD. However, there is a lack of secondary studies on BDD that have been published in peer-reviewed scientific outlets. This makes it difficult to understand the state of BDD research, as well as its strengths and gaps that could inform future BDD research.

This paper reports a Systematic Mapping Study in which 166 papers were analysed to understand the state of BDD research from 2006 to 2021. The study has revealed that most BDD research has focused on three things: using BDD to facilitate various aspects of software development; improving the BDD process and the resulting artefacts; and applying BDD in different application areas. Some of the notable gaps include scarcity of BDD research that is linked to practices in the software industry, scarcity of research in which BDD is studied alongside other software development processes and technologies, scarcity of philosophical papers on BDD, and acute shortage of metrics for measuring various aspects of BDD specifications and the processes for producing BDD specifications.

CRediT authorship contribution statement

Leonard Peter Binamungu: Conceptualization, Methodology, Investigation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Salome Maro:** Conceptualization, Methodology, Investigation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data used in our research is available at https://doi.org/10.5281/zenodo.7934235.

Appendix. Distribution of individual papers

This section contains details on the classification of the different papers according to the research type, contribution type, evaluation method and theme.

Table A.8
Studies for each research type.

Research type	Primary Studies
Validation research	Silva et al. (2017a), Esposito et al. (2018), Kannengiesser et al. (2020), Mens et al. (2019), Xu et al. (2021), Güncan and Onay Durdu (2021), Irshad et al. (2021), Oran et al. (2017), Dookhun and Nagowah (2019), Santos and Vilain (2018), Nguyen et al. (2020), Malik et al. (2021), Sarinho (2019), Carrera et al. (2014), Storer and Bob (2019), Nascimento et al. (2020), Lucassen et al. (2017), Carter and Gardner (2016a), Deng et al. (2021), Binamungu et al. (2020), Wang and Wagner (2018a,b), Mwakyanjala et al. (2020), Purkayastha et al. (2020), Aghayi et al. (2021), Zampetti et al. (2020), Binamungu et al. (2018a), Brolund (2009), Silva and Fitzgerald (2021a), Rocha et al. (2021), Rocha Silva et al. (2019a, 2020), Silva et al. (2019), Ma et al. (2019), Santos et al. (2019), Häser et al. (2016), Oliveira and Marczak (2017, 2018), Yang et al. (2019), Ma et al. (2018), Wiecher et al. (2020), Lopes de Souza et al. (2021), Wang et al. (2018), Almeida et al. (2016), Pinto et al. (2019), Rocha et al. (2019), Elshandidy et al. (2021), Cisneros et al. (2018), Wanderley et al. (2015), Pyshkin et al. (2012), Ali (2019), Okolnychyi and Fögen (2016), Hoisl et al. (2014), Goulart (2014), Wolde and Boltana (2021)
Solution proposal	Kudo et al. (2019), Silva et al. (2017b), Wanderley and da Silveria (2012), Dimanidis et al. (2018), Rahman and Gao (2015), Santos et al. (2015), Sivanandan et al. (2014), Soeken et al. (2012), Schur and Williams (2017), Contan et al. (2017), Kamalakar et al. (2013), Zaeske et al. (2021), Diepenbeck et al. (2012), Nezhad et al. (2018), Shafiee et al. (2018), Diepenbeck et al. (2018, 2014), Lopez-Pellicer et al. (2014), Hatko et al. (2014), Lazăr et al. (2010), Snook et al. (2018), Elshandidy (2019), Wiecher et al. (2021b), Carter and Gardner (2016b), de Carvalho et al. (2012), Manuaba (2019), Lai et al. (2014), Sathawornwichit and Hosono (2012), Gupta et al. (2019), Pandit et al. (2016), Sarinho (2019), Winkler et al. (2019), Schneider et al. (2018), Rocha Silva et al. (2019b, 2020a), Matula and Hunka (2018), Baillon and Bouchez-Mongardé (2010), Fazzolino and Rodrigues (2019), Fischer and Dghyam (2019), Gao and Li (2016), Bonfanti et al. (2018), Li et al. (2017), Oliveira et al. (2019), Carvalho et al. (2013), Schoeneman and Liu (2013), Wiecher et al. (2021c), King et al. (2014), Bagheri and Ensan (2013), Matula and Zacek (2018a), Bussenot et al. (2018), Silva and Fitzgerald (2021b), Morrison et al. (2013a,b), Matula and Zacek (2018b), Okubo et al. (2014), Oruç and Ovatman (2016), Silva et al. (2016), Barus (2019), Mahalakshmi and Vani (2017), Liao et al. (2017), O'Brien (2018), Hesenius et al. (2014), Pang Fung et al. (2021), Diepenbeck et al. (2013), Siqueira et al. (2017), Ali et al. (2019), Cauchi et al. (2016), Patkar et al. (2021), Lima et al. (2021), Wolde and Boltana (2020), Raharjana et al. (2020), Yen et al. (2021), Tuglular and Coşkun (2021), Wiecher et al. (2021a)
Evaluation research	Bünder and Kuchen (2019a), Bjarnason et al. (2016), Lucena (2017), Bjarnason et al. (2015), Alhaj et al. (2019), Pereira et al. (2018), Nascimento et al. (2020), Schulz et al. (2019), Alferez et al. (2019), Souza et al. (2018), So (2017), Williams et al. (2020), Binamungu et al. (2018b), Nguyen et al. (2020), Karagöz and Sözer (2017), Li et al. (2016), Zafar et al. (2021), Bünder and Kuchen (2019b), Alhaj et al. (2017), Bruschi et al. (2019)
Experience papers	Fazzolino et al. (2018), Bezsmertnyi et al. (2020), Gil et al. (2016), Bahaweres et al. (2020), Scandaroli et al. (2019), de Souza et al. (2017), Moult and Krijnen (2020), Zaeske and Durak (2020), Lübke and van Lessen (2016), Mello et al. (2018), Cavalcante and Sales (2019), Giorgi and Paulisch (2019), Butler et al. (2019)
Philosophical papers	Solis and Wang (2011), Gohil et al. (2011), Lenka et al. (2018)
Opinion papers	Faragó et al. (2020)

Table A.9
Studies for each contribution type.

Contribution type	Primary studies
Model	Silva et al. (2017a), Kudo et al. (2019), Silva et al. (2017b, 2016), Lenka et al. (2018), Rahman and Gao (2015), Lucena (2017), Solis and Wang (2011), Güncan and Onay Durdu (2021), Shafiee et al. (2018), Elshandidy (2019), Carter and Gardner (2016a), Lai et al. (2014), Winkler et al. (2019), Matula and Hunka (2018), So (2017), Williams et al. (2020), Yang et al. (2019), Matula and Zacek (2018b), Wolde and Boltana (2021)
Process	Esposito et al. (2018), Kannengiesser et al. (2020), Bünder and Kuchen (2019a), Dimanidis et al. (2018), Lucena (2017), Santos et al. (2015), Güncan and Onay Durdu (2021), Irshad et al. (2021), Alhaj et al. (2019), Soeken et al. (2012), Diepenbeck et al. (2012, 2018, 2014), Lopez-Pellicer et al. (2014), Hatko et al. (2014), Snook et al. (2018), Elshandidy (2019), Wiecher et al. (2021b), Carter and Gardner (2016b), de Carvalho et al. (2012), de Souza et al. (2017), Wang and Wagner (2018a,b), Sathawornwichit and Hosono (2012), Purkayastha et al. (2020), Aghayi et al. (2021), Sarinho (2019), Schneider et al. (2018), Rocha Silva et al. (2019a), Silva et al. (2019), Fazzolino and Rodrigues (2019), Carvalho et al. (2013), Wiecher et al. (2021c), Zaeske and Durak (2020), Bagheri and Ensan (2013), Lübke and van Lessen (2016), Bussenot et al. (2018), Silva and Fitzgerald (2021b), Karagöz and Sözer (2017), Wiecher et al. (2020), Lopes de Souza et al. (2021), Almeida et al. (2016), Silva et al. (2016), Liao et al. (2017), O'Brien (2018), Zafar et al. (2021), Pang Fung et al. (2021), Pinto et al. (2019), Alhaj et al. (2017), Elshandidy et al. (2021), Cauchi et al. (2016), Faragó et al. (2020), Ali (2019), Lima et al. (2021), Wolde and Boltana (2020), Bruschi et al. (2019), Wiecher et al. (2021a)
Method	Wanderley and da Silveria (2012), Mens et al. (2019), Xu et al. (2021), Schur and Williams (2017), Malik et al. (2021), Carrera et al. (2014), Schulz et al. (2019), Lucassen et al. (2017), Carter and Gardner (2016a), Deng et al. (2021), Alferez et al. (2019), Manuaba (2019), Mwakyanjala et al. (2020), Gupta et al. (2019), Binamungu et al. (2018a), Pandit et al. (2016), Rocha Silva et al. (2019b, 2020a), Ma et al. (2019), Matula and Zacek (2018a), Ma et al. (2018), Okubo et al. (2014), Mahalakshmi and Vani (2017), Diepenbeck et al. (2013), Siqueira et al. (2017), Gohil et al. (2011), Tuglular and Coşkun (2021)
Tool	Wanderley and da Silveria (2012), Bünder and Kuchen (2019a), Dimanidis et al. (2018), Lucena (2017), Soeken et al. (2012), Nguyen et al. (2020), Contan et al. (2017), Kamalakar et al. (2013), Sarinho (2019), Carrera et al. (2014), Storer and Bob (2019), Nezhad et al. (2018), Lopez-Pellicer et al. (2014), Lazăr et al. (2010), Wiecher et al. (2021b), Carter and Gardner (2016a), Moult and Krijnen (2020), Sathawornwichit and Hosono (2012), Brolund (2009), Sarinho (2019), Rocha Silva et al. (2019b, 2020a), Baillon and Bouchez-Mongardé (2010), Gao and Li (2016), Bonfanti et al. (2018), Li et al. (2017), Santos et al. (2019), Schoeneman and Liu (2013), Wiecher et al. (2021c), King et al. (2014), Nguyen et al. (2020), Karagöz and Sözer (2017), Li et al. (2016), Wang et al. (2018), Oruç and Ovatman (2016), Silva et al. (2016), Liao et al. (2017), Hesenius et al. (2014), Zafar et al. (2021), Bünder and Kuchen (2019b), Pinto et al. (2019), Rocha et al. (2019), Patkar et al. (2021), Wanderley et al. (2015), Raharjana et al. (2020)
Metric	Binamungu et al. (2020), Oliveira et al. (2019), Oliveira and Marczak (2017)
Empirical insights	Bjarnason et al. (2016), Irshad et al. (2021), Sivanandan et al. (2014), Bjarnason et al. (2015), Oran et al. (2017), Fazzolino et al. (2018), Dookhun and Nagowah (2019), Santos and Vilain (2018), Bezsmertnyi et al. (2020), Zaeske et al. (2021), Gil et al. (2016), Bahaweres et al. (2020), Scandaroli et al. (2019), Pereira et al. (2018), Nascimento et al. (2020), Lopez-Pellicer et al. (2014), Binamungu et al. (2020), Manuaba (2019), de Souza et al. (2017), Wang and Wagner (2018b), Zampetti et al. (2020), Silva and Fitzgerald (2021a), Rocha et al. (2021), Rocha Silva et al. (2020), Fischer and Dghyam (2019), Souza et al. (2018), So (2017), Häser et al. (2016), Williams et al. (2020), Binamungu et al. (2018b), Mello et al. (2018), Oliveira and Marczak (2018), Morrison et al. (2013a,b), Karagöz and Sözer (2017), Wang et al. (2018), Cavalcante and Sales (2019), Barus (2019), Diepenbeck et al. (2013), Giorgi and Paulisch (2019), Elshandidy et al. (2021), Ali et al. (2019), Cisneros et al. (2018), Pyshkin et al. (2012), Okolnychyi and Fögen (2016), Hoisl et al. (2014), Goulart (2014), Yen et al. (2021), Butler et al. (2019)

Table A.10
Studies that have employed specific evaluation methods.

Evaluation method	Primary studies
Example	Kannengiesser et al. (2020), Kudo et al. (2019), Silva et al. (2017b), Wanderley and da Silveria (2012), Dimanidis et al. (2018), Rahman and Gao (2015), Sivanandan et al. (2014), Soeken et al. (2012), Schur and Williams (2017), Contan et al. (2017), Zaeske et al. (2021), Diepenbeck et al. (2012), Nezhad et al. (2018), Diepenbeck et al. (2018, 2014), Hatko et al. (2014), Lazăr et al. (2010), Snook et al. (2018), Wiecher et al. (2021b), Carter and Gardner (2016b), de Carvalho et al. (2012), Wang and Wagner (2018a), Sathawornwichit and Hosono (2012), Gupta et al. (2019), Pandit et al. (2016), Brolund (2009), Sarinho (2019), Schneider et al. (2018), Morrison et al. (2013a,b), Matula and Hunka (2018), Baillon and Bouchez-Mongardé (2010), Fischer and Dghyam (2019), Bonfanti et al. (2018), Carvalho et al. (2013), Schoeneman and Liu (2013), Wiecher et al. (2021c), King et al. (2014), Matula and Zacek (2018a), Bussenot et al. (2018), Silva and Fitzgerald (2021b), Matula and Zacek (2018b), Wiecher et al. (2020), Almeida et al. (2016), Oruç and Ovatman (2016), Mahalakshmi and Vani (2017), O'Brien (2018), Hesenius et al. (2014), Zafar et al. (2021), Pang Fung et al. (2021), Siqueira et al. (2017), Cauchi et al. (2016), Patkar et al. (2021), Lima et al. (2021), Wolde and Boltana (2020, 2021), Raharjana et al. (2020), Yen et al. (2021), Tuglular and Coşkun (2021), Wiecher et al. (2021a)
Case study	Silva et al. (2017a), Bjarnason et al. (2016), Lucena (2017), Güncan and Onay Durdu (2021), Irshad et al. (2021), Bjarnason et al. (2015), Alhaj et al. (2019), Nguyen et al. (2020), Malik et al. (2021), Carrera et al. (2014), Pereira et al. (2018), Nascimento et al. (2020), Schulz et al. (2019), Lopez-Pellicer et al. (2014), Carter and Gardner (2016a), Alferez et al. (2019), Binamungu et al. (2020), Manuaba (2019), Mwakyanjala et al. (2020), Purkayastha et al. (2020), Aghayi et al. (2021), Zampetti et al. (2020), Winkler et al. (2019), Rocha et al. (2021), Rocha Silva et al. (2019a,b, 2020a,?), Silva et al. (2019), Fazzolino and Rodrigues (2019), Oliveira et al. (2019), Souza et al. (2018), So (2017), Williams et al. (2020), Binamungu et al. (2018b), Nguyen et al. (2020), Oliveira and Marczak (2017, 2018), Karagöz and Sözer (2017), Okubo et al. (2014), Li et al. (2016), Liao et al. (2017), Diepenbeck et al. (2013), Alhaj et al. (2017), Elshandidy et al. (2021), Ali et al. (2019), Ali (2019), Goulart (2014), Bruschi et al. (2019)
Experiment	Esposito et al. (2018), Mens et al. (2019), Bünder and Kuchen (2019a), Xu et al. (2021), Oran et al. (2017), Dookhun and Nagowah (2019), Santos and Vilain (2018), Kamalakar et al. (2013), Storer and Bob (2019), Deng et al. (2021), Wang and Wagner (2018b), Zampetti et al. (2020), Binamungu et al. (2018a), Silva and Fitzgerald (2021a), Gao and Li (2016), Li et al. (2017), Ma et al. (2019), Santos et al. (2019), Häser et al. (2016), Yang et al. (2019), Ma et al. (2018), Lopes de Souza et al. (2021), Wang et al. (2018), Barus (2019), Bünder and Kuchen (2019b), Pinto et al. (2019), Rocha et al. (2019), Hoisl et al. (2014)
Experience report	Fazzolino et al. (2018), Bezsmertnyi et al. (2020), Gil et al. (2016), Bahaweres et al. (2020), Scandaroli et al. (2019), de Souza et al. (2017), Moult and Krijnen (2020), Zaeske and Durak (2020), Lübke and van Lessen (2016), Mello et al. (2018), Cavalcante and Sales (2019), Gohil et al. (2011), Giorgi and Paulisch (2019), Butler et al. (2019)
Discussion	Santos et al. (2015), Solis and Wang (2011), Contan et al. (2017), Shafiee et al. (2018), Elshandidy (2019), Lai et al. (2014), Bagheri and Ensan (2013), Silva et al. (2016), Pyshkin et al. (2012), Faragó et al. (2020), Okolnychyi and Fögen (2016)
Rigorous analysis	Lucassen et al. (2017), Diepenbeck et al. (2018, 2014)

Table A.11
Themes in existing BDD research.

Sn	Theme	Sub-theme(s)	Citation(s)
1	Facilitating software development	Safety	Wang and Wagner (2018a,b), Baillon and Bouchez-Mongardé (2010), Wiecher et al. (2020), Wang et al. (2018)
		GUI testing	Silva et al. (2017b), Bünder and Kuchen (2019a), Sivanandan et al. (2014), Schur and Williams (2017), Bahaweres et al. (2020), Rocha Silva et al. (2019a,b), Bünder and Kuchen (2019b)
		Artefact generation	Wanderley and da Silveria (2012), Bünder and Kuchen (2019a), Dimanidis et al. (2018), Soeken et al. (2012), Schur and Williams (2017), Nguyen et al. (2020), Kamalakar et al. (2013), Malik et al. (2021), Carrera et al. (2014), Storer and Bob (2019), Nezhad et al. (2018), Lazăr et al. (2010), Snook et al. (2018), Carter and Gardner (2016a,b), Deng et al. (2021), Alferez et al. (2019), Gupta et al. (2019), Pandit et al. (2016), Silva and Fitzgerald (2021a), Fazzolino and Rodrigues (2019), Gao and Li (2016), Bonfanti et al. (2018), Schoeneman and Liu (2013), King et al. (2014), Williams et al. (2020), Nguyen et al. (2020), Li et al. (2016), Mahalakshmi and Vani (2017), O'Brien (2018), Diepenbeck et al. (2013), Bünder and Kuchen (2019b), Pinto et al. (2019), Siqueira et al. (2017), Raharjana et al. (2020), Butler et al. (2019)
		Product lines	Elshandidy (2019), Bagheri and Ensan (2013), Elshandidy et al. (2021), Tuglular and Coşkun (2021)
		Compliance testing	Lopez-Pellicer et al. (2014), Moult and Krijnen (2020), Williams et al. (2020), Morrison et al. (2013a,b), Pang Fung et al. (2021)
		Traceability	Lucassen et al. (2017), Silva and Fitzgerald (2021b), Yang et al. (2019)
		Model testing	Mens et al. (2019), Snook et al. (2018, 2021), Silva and Fitzgerald (2021a), Rocha Silva et al. (2019a), Silva et al. (2019), Fischer and Dghyam (2019)
		Business process modelling	de Carvalho et al. (2012), Carvalho et al. (2013), Lübke and van Lessen (2016), Matula and Zacek (2018b)
		Security and privacy	Lai et al. (2014), Purkayastha et al. (2020), Okubo et al. (2014)
		Model-based testing	Sivanandan et al. (2014), Li et al. (2016)
		Integration testing	Purkayastha et al. (2020), Bussenot et al. (2018)
		Load testing	Schulz et al. (2019)
		Syntax highlighting	Matula and Zacek (2018a)
		Reproducing bugs	Karagöz and Sözer (2017)
		Spreadsheets testing	Almeida et al. (2016)
		Test selection	Xu et al. (2021)
		Usability testing	Güncan and Onay Durdu (2021)
		Software documentation	Brolund (2009)
2	Improving BDD	Specification quality	Binamungu et al. (2020), Oliveira et al. (2019), Binamungu et al. (2018b), Oliveira and Marczak (2017, 2018), Binamungu et al. (2018a)
		BDD adoption	Irshad et al. (2021), Contan et al. (2017), Bezsmertnyi et al. (2020), Gil et al. (2016), Scandaroli et al. (2019), Pereira et al. (2018), Shafiee et al. (2018), Nascimento et al. (2020), Zampetti et al. (2020), Binamungu et al. (2018b), Mello et al. (2018), Cavalcante and Sales (2019), Barus (2019), Zampetti et al. (2020)
		Co-evolution of artefacts	Sathawornwichit and Hosono (2012), Zampetti et al. (2020), Santos et al. (2019), Yang et al. (2019), Rocha et al. (2019)
		Duplicate detection and refactoring	Binamungu et al. (2018a)
		BDD characteristics	Solis and Wang (2011), Faragó et al. (2020)
		Specification language	Kudo et al. (2019), Lucena (2017), Lazăr et al. (2010), Deng et al. (2021), Snook et al. (2021), Rocha Silva et al. (2020), Li et al. (2017), Häser et al. (2016), King et al. (2014), Hesenius et al. (2014), Pang Fung et al. (2021), Bünder and Kuchen (2019b), Patkar et al. (2021), Wanderley et al. (2015), Hoisl et al. (2014), Gutiérrez Rodríguez et al. (2017)
		Test cases as requirements	Bjarnason et al. (2016, 2015), Elshandidy et al. (2021)
		BDD tools	Okolnychyi and Fögen (2016), Pyshkin et al. (2012)

(*continued on next page*)

Table A.11 (*continued*).

Sn	Theme	Sub-theme(s)	Citation(s)
3	BDD application areas	Hardware context	Kannengiesser et al. (2020), Alhaj et al. (2019), Zaeske et al. (2021), Diepenbeck et al. (2012), Nezhad et al. (2018), Diepenbeck et al. (2018, 2014), Deng et al. (2021), Mwakyanjala et al. (2020), Winkler et al. (2019), Bussenot et al. (2018), Zafar et al. (2021), Alhaj et al. (2017), Cauchi et al. (2016), Yen et al. (2021), Wiecher et al. (2021a)
		Health context	Hatko et al. (2014), Purkayastha et al. (2020), Mello et al. (2018), Morrison et al. (2013a,b), Giorgi and Paulisch (2019), Bruschi et al. (2019)
		Education context	Sarinho (2019), Nascimento et al. (2020), Rocha et al. (2021), Goulart (2014)
		Mobile apps	Nguyen et al. (2020), Ali et al. (2019), Ali (2019)
		Enterprise systems	Fazzolino et al. (2018), Gohil et al. (2011)
		BDD and computer networking	Esposito et al. (2018)
		Product configuration systems	Shafiee et al. (2018)
		Multi-agent systems	Carrera et al. (2014)
4	Ontologies	Ontologies	Silva et al. (2017a), Rocha Silva et al. (2019a,b, 2020a), Matula and Hunka (2018), Souza et al. (2018), Matula and Zacek (2018a), Silva and Fitzgerald (2021b), Matula and Zacek (2018b), Lopes de Souza et al. (2021), Silva et al. (2016)
5	BDD and other processes	Comparison with other techniques	Santos et al. (2015), Oran et al. (2017), Dookhun and Nagowah (2019), Santos and Vilain (2018), Bezsmertnyi et al. (2020), Manuaba (2019), Cisneros et al. (2018)
		BDD and Scrum	de Souza et al. (2017), Souza et al. (2018), Lopes de Souza et al. (2021)
		BDD and DevOps	Zaeske and Durak (2020), Giorgi and Paulisch (2019)
6	BDD and other technologies	Machine Learning	Deng et al. (2021), Ma et al. (2018), Pang Fung et al. (2021)
		Blockchain	Liao et al. (2017)
		Microservices	Rahman and Gao (2015), Aghayi et al. (2021), Ma et al. (2019, 2018), Lima et al. (2021)
		Web services	Dimanidis et al. (2018), Oruç and Ovatman (2016), Wolde and Boltana (2020, 2021)

References

Abushama, H.M., Alassam, H.A., Elhaj, F.A., 2020. The effect of test-driven development and behavior-driven development on project success factors: A systematic literature review based study. In: 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering. ICCCEEE, IEEE, pp. 1–9.

Aghayi, E., LaToza, T.D., Surendra, P., Abolghasemi, S., 2021. Crowdsourced behavior-driven development. J. Syst. Softw. 171, 110840.

Alferez, M., Pastore, F., Sabetzadeh, M., Briand, L., Riccardi, J.-R., 2019. Bridging the gap between requirements modeling and behavior-driven development. In: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems. MODELS, IEEE, pp. 239–249.

Alhaj, M., Arbez, G., Peyton, L., 2017. Using behaviour-driven development with hardware-software co-design for autonomous load management. In: 2017 8th International Conference on Information and Communication Systems. ICICS, IEEE, pp. 46–51.

Alhaj, M., Arbez, G., Peyton, L., 2019. Approach of integrating behaviour-driven development with hardware/software codesign. Int. J. Innovative Comput. Inf. Control 15 (3), 1177–1191.

Ali, Z., 2019. Behavior-driven development as an error-reduction practice for mobile application testing. Int. J. Comput. Sci. Issues (IJCSI) 16 (2), 1–10.

Ali, Z., Awwad, A., Slany, W., 2019. Using executable specification and regression testing for broadcast mechanism of visual programming language on smartphones. Int. J. Interact. Mob. Technol. 13 (2), 50–65.

Almeida, L., Cirilo, E., Barbosa, E.A., 2016. SS-BDD: Automated acceptance testing for spreadsheets. In: Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing. pp. 1–10.

Bagheri, E., Ensan, F., 2013. Light-weight software product lines for small and medium-sized enterprises (SMEs). In: Proceedings of the 2013 Conference of the Center for Advanced Studies on Collaborative Research. pp. 311–324.

Bahaweres, R.B., Oktaviani, E., Wardhani, L.K., Hermadi, I., Suroso, A., Solihin, I.P., Arkeman, Y., 2020. Behavior-driven development (BDD) Cucumber Katalon for automation GUI testing case CURA and Swag labs. In: 2020 International Conference on Informatics, Multimedia, Cyber and Information System. ICIMCIS, IEEE, pp. 87–92.

Baillon, C., Bouchez-Mongardé, S., 2010. Executable requirements in a safety-critical context with Ada. Ada User J. 31 (2), 131–135.

Barus, A.C., 2019. The implementation of ATDD and BDD from testing perspectives. In: J. Phys. Conf. Ser.. 1175 (1), 012112.

Bezsmertnyi, O., Golian, N., Golian, V., Afanasieva, I., 2020. Behavior driven development approach in the modern quality control process. In: 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology. PIC S&T, IEEE, pp. 217–220.

Binamungu, L.P., Embury, S.M., Konstantinou, N., 2018a. Detecting duplicate examples in behaviour driven development specifications. In: 2018 IEEE Workshop on Validation, Analysis and Evolution of Software Tests. VST, IEEE, pp. 6–10.

Binamungu, L.P., Embury, S.M., Konstantinou, N., 2018b. Maintaining behaviour driven development specifications: Challenges and opportunities. In: 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering. SANER, IEEE, pp. 175–184.

Binamungu, L.P., Embury, S.M., Konstantinou, N., 2020. Characterising the quality of behaviour driven development specifications. In: International Conference on Agile Software Development. Springer, Cham, pp. 87–102.

Bjarnason, E., Unterkalmsteiner, M., Borg, M., Engström, E., 2016. A multi-case study of Agile requirements engineering and the use of test cases as requirements. Inf. Softw. Technol. 77, 61–79.

Bjarnason, E., Unterkalmsteiner, M., Engström, E., Borg, M., 2015. An industrial case study on test cases as requirements. In: International Conference on Agile Software Development. Springer, pp. 27–39.

Bonfanti, S., Gargantini, A., Mashkoor, A., 2018. Generation of behavior-driven development C++ tests from abstract state machine scenarios. In: International Conference on Model and Data Engineering. Springer, pp. 146–152.

Brolund, D., 2009. Documentation by example. In: International Conference on Agile Processes and Extreme Programming in Software Engineering. Springer, pp. 251–252.

Bruschi, S., Xiao, L., Kavatkar, M., et al., 2019. Behavior driven development (BDD): A case study in healthtech. In: Pacific NW Software Quality Conference.

Bünder, H., Kuchen, H., 2019a. A model-driven approach for behavior-driven GUI testing. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. pp. 1742–1751.

Bünder, H., Kuchen, H., 2019b. Towards behavior-driven graphical user interface testing. ACM SIGAPP Appl. Comput. Rev. 19 (2), 5–17.
Bussenot, R., Leblanc, H., Percebois, C., 2018. Orchestration of domain specific test languages with a behavior driven development approach. In: 2018 13th Annual Conference on System of Systems Engineering. SoSE, IEEE, pp. 431–437.
Butler, M., Dghaym, D., Hoang, T.S., Omitola, T., Snook, C., Fellner, A., Schlick, R., Tarrach, T., Fischer, T., Tummeltshammer, P., 2019. Behaviour-driven formal model development of the etcs hybrid level 3. In: 2019 24th International Conference on Engineering of Complex Computer Systems. ICECCS, IEEE, pp. 97–106.
Byron, W., Kevin, S., Carla, B., Joseph, L., Trikalinos, T.A., 2012. Deploying an interactive machine learning system in an evidence-based practice center: Abstrackr. In: In Proc. of the ACM International Health Informatics Symposium. IHI, pp. 819–824.
Carrera, Á., Iglesias, C.A., Garijo, M., 2014. Beast methodology: An Agile testing methodology for multi-agent systems based on behaviour driven development. Inform. Syst. Front. 16 (2), 169–182.
Carter, J.D., Gardner, W.B., 2016a. BHive: Behavior-driven development meets B-method. In: Quality Software Through Reuse and Integration. Springer, pp. 232–255.
Carter, J., Gardner, W.B., 2016b. BHive: Towards behaviour-driven development supported by B-method. In: 2016 IEEE 17th International Conference on Information Reuse and Integration. IRI, IEEE, pp. 249–256.
Carvalho, R.A.d., Carvalho e Silva, F.L.d., ManhÑes, R.S., Oliveira, G.L.d., 2013. Implementing behavior driven development in an open source ERP. In: Enterprise Information Systems of the Future. Springer, pp. 242–249.
Cauchi, A., Colombo, C., Francalanza, A., Micallef, M., Pace, G., 2016. Using Gherkin to extract tests and monitors for safer medical device interaction design. In: Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems. pp. 275–280.
Cavalcante, M.G., Sales, J.I., 2019. The behavior driven development applied to the software quality test. In: 2019 14th Iberian Conference on Information Systems and Technologies. CISTI, IEEE, pp. 1–4.
Chen, L., Babar, M.A., 2011. A systematic review of evaluation of variability management approaches in software product lines. Inf. Softw. Technol. 53 (4), 344–362.
Cisneros, L.A., Reis, C.I., Maximiano, M., Quiña, J.A., 2018. An experimental evaluation of ITL, TDD and BDD. In: ICSEA 2018, the Thirteenth International Conference on Software Engineering Advances. ThinkMind, pp. 20–24.
Contan, A., Miclea, L., Dehelean, C., 2017. Automated testing framework development based on social interaction and communication principles. In: 2017 14th International Conference on Engineering of Modern Electric Systems. EMES, IEEE, pp. 136–139.
de Carvalho, R.A., e Silva, F., Manhaes, R.S., 2012. Business language driven development: Joining business process models to automated tests. Adv. Enterprise Inform. Syst. II 167–177.
de Souza, P.L., do Prado, A.F., de Souza, W.L., Pereira, S.M.d.S.F., Pires, L.F., 2017. Combining behaviour-driven development with scrum for software development in the education domain. In: ICEIS,Vol. 2. pp. 449–458.
Deng, Y., Lou, G., Zheng, X., Zhang, T., Kim, M., Liu, H., Wang, C., Chen, T.Y., 2021. BMT: Behavior driven development-based metamorphic testing for autonomous driving models. In: 2021 IEEE/ACM 6th International Workshop on Metamorphic Testing. MET, IEEE, pp. 32–36.
Diepenbeck, M., Kühne, U., Soeken, M., Drechsler, R., 2014. Behaviour driven development for tests and verification. In: International Conference on Tests and Proofs. Springer, pp. 61–77.
Diepenbeck, M., Kühne, U., Soeken, M., Grosse, D., Drechsler, R., 2018. Behaviour driven development for hardware design. IPSJ Trans. Syst. LSI Des. Methodol. 11, 29–45.
Diepenbeck, M., Soeken, M., Grobe, D., Drechsler, R., 2013. Towards automatic scenario generation from coverage information. In: 2013 8th International Workshop on Automation of Software Test. AST, IEEE, pp. 82–88.
Diepenbeck, M., Soeken, M., Große, D., Drechsler, R., 2012. Behavior driven development for circuit design and verification. In: 2012 IEEE International High Level Design Validation and Test Workshop. HLDVT, IEEE, pp. 9–16.
Dimanidis, A., Chatzidimitriou, K.C., Symeonidis, A.L., 2018. A natural language driven approach for automated web api development: Gherkin2oas. In: Companion Proceedings of the the Web Conference 2018. pp. 1869–1874.
Dookhun, A.S., Nagowah, L., 2019. Assessing the effectiveness of test-driven development and behavior-driven development in an industry setting. In: 2019 International Conference on Computational Intelligence and Knowledge Economy. ICCIKE, IEEE, pp. 365–370.
Dyba, T., Kitchenham, B.A., Jorgensen, M., 2005. Evidence-based software engineering for practitioners. IEEE Softw. 22 (1), 58–65.
Egbreghts, A., 2017. A literature review of behavior driven development using grounded theory. In: 27th Twente Student Conference on IT. University of Twente, Netherlands.
Elshandidy, H., 2019. Behaviour-driven requirements engineering for Agile product line engineering. In: 2019 IEEE 27th International Requirements Engineering Conference. RE, IEEE, pp. 434–439.
Elshandidy, H., Mazen, S., Hassanein, E., Nasr, E., 2021. Using behaviour-driven requirements engineering for establishing and managing agile product lines. Int. J. Adv. Comput. Sci. Appl. 12 (2), 591–596.
Esposito, F., Wang, J., Contoli, C., Davoli, G., Cerroni, W., Callegati, F., 2018. A behavior-driven approach to intent specification for software-defined infrastructure management. In: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks. NFV-SDN, IEEE, pp. 1–6.
Faragó, D., Friske, M., Sokenou, D., 2020. Towards a taxonomy for applying behavior-driven development (BDD). Softwaretechnik-Trends 40 (3), 3–7.
Fazzolino, R., de Faria, H.M., Amaral, L.H.V., Canedo, E.D., Rodrigues, G.N., Bonifácio, R., 2018. Assessing Agile testing practices for enterprise systems: A survey approach. In: Proceedings of the III Brazilian Symposium on Systematic and Automated Software Testing. pp. 29–38.
Fazzolino, R., Rodrigues, G.N., 2019. Feature-trace: Generating operational profile and supporting testing prioritization from BDD features. In: Proceedings of the XXXIII Brazilian Symposium on Software Engineering. pp. 332–336.
Fischer, T., Dghyam, D., 2019. Formal model validation through acceptance tests. In: International Conference on Reliability, Safety, and Security of Railway Systems. Springer, pp. 159–169.
Gao, N., Li, Z., 2016. Generating testing codes for behavior-driven development from problem diagrams: A tool-based approach. In: 2016 IEEE 24th International Requirements Engineering Conference. RE, IEEE, pp. 399–400.
Gil, J.P., Garces, M., Broguiere, D., Shen, T.-C., 2016. Behavior driven testing in ALMA telescope calibration software. In: Software and Cyberinfrastructure for Astronomy IV, Vol. 9913. SPIE, pp. 128–141.
Giorgi, F., Paulisch, F., 2019. Transition towards continuous delivery in the healthcare domain. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice. ICSE-SEIP, IEEE, pp. 253–254.
Gohil, K., Alapati, N., Joglekar, S., 2011. Towards behavior driven operations (BDOps). In: 3rd International Conference on Advances in Recent Technologies in Communication and Computing. ARTCom 2011, IET, pp. 262–264.
Goulart, A.E.E., 2014. Using behavioral driven development (BDD) in capstone design projects. In: 2014 ASEE Annual Conference & Exposition. pp. 24–1322.
Güncan, D., Onay Durdu, P., 2021. A user-centered behavioral software development model. J. Softw. Evol. Process 33 (2), e2274.
Gupta, A., Poels, G., Bera, P., 2019. Creation of multiple conceptual models from user stories–a natural language processing approach. In: International Conference on Conceptual Modeling. Springer, pp. 47–57.
Gutiérrez Rodríguez, J.J., Ramos Román, I., Mejías Risoto, M., Arévalo Maldonado, C., Sánchez Begines, J.M., Lizcano, D., 2017. Modelling Gherkin scenarios using UML. In: ISD 2017: 26th International Conference on Information Systems Development (2017). Association for Information Systems.
Häser, F., Felderer, M., Breu, R., 2016. Is business domain language support beneficial for creating test case specifications: A controlled experiment. Inf. Softw. Technol. 79, 52–62.
Hatko, R., Mersmann, S., Puppe, F., 2014. Behaviour-driven development for computer-interpretable clinical guidelines.. In: 10th Workshop on Knowledge Engineering and Software Engineering. KESE10, CEUR Workshop Proceedings.
Hesenius, M., Griebe, T., Gruhn, V., 2014. Towards a behavior-oriented specification and testing language for multimodal applications. In: Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems. pp. 117–122.
Hoisl, B., Sobernig, S., Strembeck, M., 2014. Comparing three notations for defining scenario-based model tests: A controlled experiment. In: 2014 9th International Conference on the Quality of Information and Communications Technology. IEEE, pp. 180–189.
Irshad, M., Britto, R., Petersen, K., 2021. Adapting behavior driven development (BDD) for large-scale software systems. J. Syst. Softw. 177, 110944.
Kamalakar, S., Edwards, S.H., Dao, T.M., 2013. Automatically generating tests from natural language descriptions of software behavior. In: International Conference on Evaluation of Novel Software Approaches to Software Engineering. ENASE, pp. 238–245.
Kannengiesser, U., Krenn, F., Stary, C., 2020. A behaviour-driven development approach for cyber-physical production systems. In: 2020 IEEE Conference on Industrial Cyberphysical Systems, Vol. 1. ICPS, IEEE, pp. 179–184.
Karagöz, G., Sözer, H., 2017. Reproducing failures based on semiformal failure scenario descriptions. Softw. Qual. J. 25 (1), 111–129.
King, T.M., Nunez, G., Santiago, D., Cando, A., Mack, C., 2014. Legend: An Agile dsl toolset for web acceptance testing. In: Proceedings of the 2014 International Symposium on Software Testing and Analysis. pp. 409–412.
Klotins, E., Unterkalmsteiner, M., Gorschek, T., 2019. Software engineering in start-up companies: An analysis of 88 experience reports. Empir. Softw. Eng. 24 (1), 68–102.
Kudo, T.N., Bulcão-Neto, R.F., Vincenzi, A.M., 2019. A conceptual metamodel to bridging requirement patterns to test patterns. In: Proceedings of the XXXIII Brazilian Symposium on Software Engineering. pp. 155–160.
Lai, S.-T., Leu, F.-Y., Chu, W.C.-C., 2014. Combining IID with BDD to enhance the critical quality of security functional requirements. In: 2014 Ninth International Conference on Broadband and Wireless Computing, Communication and Applications. IEEE, pp. 292–299.

Lazăr, I., Motogna, S., Pârv, B., 2010. Behaviour-driven development of foundational UML components. Electron. Notes Theor. Comput. Sci. 264 (1), 91–105.
Lenka, R.K., Kumar, S., Mamgain, S., 2018. Behavior driven development: Tools and challenges. In: 2018 International Conference on Advances in Computing, Communication Control and Networking. ICACCCN, IEEE, pp. 1032–1037.
Li, N., Escalona, A., Kamal, T., 2016. Skyfire: Model-based testing with cucumber. In: 2016 IEEE International Conference on Software Testing, Verification and Validation. ICST, IEEE, pp. 393–400.
Li, T., Tsubota, S., Hirono, K., 2017. Gherkin syntax extension for parameterization of network switch configurations in test specification. In: 2017 IEEE International Symposium on Software Reliability Engineering Workshops. ISSREW, IEEE, pp. 12–14.
Liao, C.-F., Cheng, C.-J., Chen, K., Lai, C.-H., Chiu, T., Wu-Lee, C., 2017. Toward a service platform for developing smart contracts on blockchain in bdd and tdd styles. In: 2017 IEEE 10th Conference on Service-Oriented Computing and Applications. SOCA, IEEE, pp. 133–140.
Lillnor, R., He, B., 2020. A Systematic Mapping Study of Behavior Driven Development. (Bachelor Thesis). KTH School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden.
Lima, H.R., Souza, K.C., de Paula, L.V., e Martins, L.M., Giozza, W.F., de Sousa, R.T., 2021. Acceptance tests over microservices architecture using behaviour-driven development. In: 2021 16th Iberian Conference on Information Systems and Technologies. CISTI, IEEE, pp. 1–6.
Lopes de Souza, P., Lopes de Souza, W., Ferreira Pires, L., 2021. ScrumOntoBDD: Agile software development based on scrum, ontologies and behaviour-driven development. J. Braz. Comput. Soc. 27 (1), 1–45.
Lopez-Pellicer, F.J., Latre, M.Á., Nogueras-Iso, J., Zarazaga-Soria, F.J., Barrera, J., 2014. Behaviour-driven development applied to the conformance testing of INSPIRE web services. In: Connecting a Digital Europe Through Location and Place. Springer, pp. 325–339.
Lübke, D., van Lessen, T., 2016. Modeling test cases in BPMN for behavior-driven development. IEEE Softw. 33 (5), 15–21.
Lucassen, G., Dalpiaz, F., van der Werf, J.M.E., Brinkkemper, S., Zowghi, D., 2017. Behavior-driven requirements traceability via automated acceptance tests. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 431–434.
Lucena, C., 2017. A semantic wiki approach to enable behaviour driven requirements management. J. Control Eng. Appl. Inform. 19 (4), 72–83.
Ma, S.-P., Chuang, Y., Lan, C.-W., Chen, H.-M., Huang, C.-Y., Li, C.-Y., 2018. Scenario-based microservice retrieval using Word2Vec. In: 2018 IEEE 15th International Conference on E-Business Engineering. ICEBE, IEEE, pp. 239–244.
Ma, S.-P., Fan, C.-Y., Chuang, Y., Liu, I.-H., Lan, C.-W., 2019. Graph-based and scenario-driven microservice analysis, retrieval, and testing. Future Gener. Comput. Syst. 100, 724–735.
Mahalakshmi, G., Vani, V., 2017. Theoretical verification of test cases for behavior driven development. In: 2017 Second International Conference on Recent Trends and Challenges in Computational Models. ICRTCCM, IEEE, pp. 309–314.
Malik, M.I., Sindhu, M.A., Khattak, A.S., Abbasi, R.A., Saleem, K., 2021. Automating test oracles from restricted natural language Agile requirements. Expert Syst. 38 (1), e12608.
Manuaba, I.B.K., 2019. Combination of test-driven development and behavior-driven development for improving backend testing performance. Procedia Comput. Sci. 157, 79–86.
Matula, J., Hunka, F., 2018. Enterprise ontology-driven development. In: Workshop on Enterprise and Organizational Modeling and Simulation. Springer, pp. 3–15.
Matula, J., Zacek, J., 2018a. Ontological syntax highlighting. In: AIP Conference Proceedings, Vol. 1978, no. 1. AIP Publishing LLC, p. 060006.
Matula, J., Zacek, J., 2018b. Reengineering legacy systems with transaction model. In: AIP Conference Proceedings, Vol. 2040, no. 1. AIP Publishing LLC, p. 030016.
Mello, P., Ximenes, P., Lemos, R., Bessa, A.L., Cortés, M.I., Rocha, C.L., 2018. On the applicability of BDD in a business intelligence project: experience report. In: Proceedings of the 17th Brazilian Symposium on Software Quality. pp. 296–304.
Mens, T., Decan, A., Spanoudakis, N.I., 2019. A method for testing and validating executable statechart models. Softw. Syst. Model. 18 (2), 837–863.
Montesi, M., Lago, P., 2008. Software engineering article types: An analysis of the literature. J. Syst. Softw. 81 (10), 1694–1714.
Morrison, P., Holmgreen, C., Massey, A., Williams, L., 2013a. Proposing regulatory-driven automated test suites. In: 2013 Agile Conference. IEEE, pp. 11–21.
Morrison, P., Holmgreen, C., Massey, A., Williams, L., 2013b. Proposing regulatory-driven automated test suites for electronic health record systems. In: 2013 5th International Workshop on Software Engineering in Health Care. SEHC, IEEE, pp. 46–49.
Moult, D., Krijnen, T., 2020. Compliance checking on building models with the Gherkin language and continuous integration. In: Proceedings of the EG-ICE 2020 Workshop on Intelligent Computing in Engineering. Technische UniversitäT Berlin.
Mujtaba, S., Petersen, K., Feldt, R., Mattsson, M., 2008. Software product line variability: A systematic mapping study. School of Engineering, Blekinge Institute of Technology.
Mwakyanjala, M.B., Nieto-Peroy, C., Emami, M.R., van de Beek, J., 2020. Concurrent development and verification of an all-software baseband for satellite ground operations. Int. J. Satell. Commun. Netw. 38 (2), 209–227.
Nascimento, N., Santos, A.R., Sales, A., Chanin, R., 2020. Behavior-driven development: A case study on its impacts on agile development teams. In: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. pp. 109–116.
Nezhad, A.S., Lukkien, J.J., Mak, R.H., 2018. Behavior-driven development for real-time embedded systems. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation, Vol. 1. ETFA, IEEE, pp. 59–66.
Nguyen, D.-M., Huynh, Q.-T., Ha, N.-H., Nguyen, T.-H., 2020. Automated test input generation via model inference based on user story and acceptance criteria for mobile application development. Int. J. Softw. Eng. Knowl. Eng. 30 (03), 399–425.
North, D., 2006. Introducing BDD. Better Software Magazine.
O'Brien, M., 2018. Toward leveraging Gherkin controlled natural language and machine translation for global product information development. In: 21st Annual Conference of the European Association for Machine Translation. European Association for Machine Translation.
Okolnychyi, A., Fögen, K., 2016. A study of tools for behavior-driven development. Full-Scale Softw. Eng./Curr. Trends Rel. Eng. 7, 7–12.
Okubo, T., Kakizaki, Y., Kobashi, T., Washizaki, H., Ogata, S., Kaiya, H., Yoshioka, N., 2014. Security and privacy behavior definition for behavior driven development. In: International Conference on Product-Focused Software Process Improvement. Springer, pp. 306–309.
Oliveira, G., Marczak, S., 2017. On the empirical evaluation of BDD scenarios quality: Preliminary findings of an empirical study. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops. REW, IEEE, pp. 299–302.
Oliveira, G., Marczak, S., 2018. On the understanding of bdd scenarios' quality: Preliminary practitioners' opinions. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. Springer, pp. 290–296.
Oliveira, G., Marczak, S., Moralles, C., 2019. How to evaluate BDD scenarios' quality? In: Proceedings of the XXXIII Brazilian Symposium on Software Engineering. pp. 481–490.
Oran, A.C., Nascimento, E., Santos, G., Conte, T., 2017. Analysing requirements communication using use case specification and user stories. In: Proceedings of the 31st Brazilian Symposium on Software Engineering. pp. 214–223.
Oruç, A.F., Ovatman, T., 2016. Testing of web services using behavior-driven development. In: CLOSER,Vol. 2. pp. 85–92.
Pandit, P., Tahiliani, S., Sharma, M., 2016. Distributed Agile: Component-based user acceptance testing. In: 2016 Symposium on Colossal Data Analysis and Networking. CDAN, IEEE, pp. 1–9.
Pang Fung, C., Pang, W., Naja, I., Markovic, M., Edwards, P., 2021. Towards accountability driven development for machine learning systems. In: Proceedings of the SICSA EXplainable Artifical Intelligence Workshop. CEUR Workshop Proceedings.
Patkar, N., Chiş, A., Stulova, N., Nierstrasz, O., 2021. Interactive behavior-driven development: A low-code perspective. In: 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion. MODELS-C, IEEE, pp. 128–137.
Pereira, L., Sharp, H., de Souza, C., Oliveira, G., Marczak, S., Bastos, R., 2018. Behavior-driven development benefits and challenges: Reports from an industrial study. In: Proceedings of the 19th International Conference on Agile Software Development: Companion. pp. 1–4.
Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering. EASE 12, pp. 1–10.
Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18.
Pfleeger, N.F.S., Glass, R., 1994. Science and substance: A challenge to software engineers. IEEE Softw. 11 (4), 86–95.
Pinto, T.D., Gonçalves, W.I., Costa, P.V., 2019. User interface prototype generation from Agile requirements specifications written in concordia. In: Proceedings of the 25th Brazillian Symposium on Multimedia and the Web. pp. 61–64.
Plat, N., van Katwijk, J., Toetenel, H., 1992. Application and benefits of formal methods in software development. Softw. Eng. J. 7 (5), 335–346.
Purkayastha, S., Goyal, S., Phillips, T., Wu, H., Haakenson, B., Zou, X., 2020. Continuous security through integration testing in an electronic health records system. In: 2020 International Conference on Software Security and Assurance. ICSSA, IEEE, pp. 26–31.

Pyshkin, E., Mozgovoy, M., Glukhikh, M., 2012. On requirements for acceptance testing automation tools in behavior driven software development. In: Proceedings of the 8th Software Engineering Conference in Russia. CEE-SECR.
Raharjana, I.K., Harris, F., Justitia, A., 2020. Tool for generating behavior-driven development test-cases. J. Inform. Syst. Eng. Bus. Intell. 6 (1), 27.
Rahman, M., Gao, J., 2015. A reusable automated acceptance testing architecture for microservices in behavior-driven development. In: 2015 IEEE Symposium on Service-Oriented System Engineering. IEEE, pp. 321–325.
Rocha, T., Borba, P., Santos, J.P., 2019. Using acceptance tests to predict files changed by programming tasks. J. Syst. Softw. 154, 176–195.
Rocha, F.G., Souza, L.S., Silva, T.S., Rodríguez, G., 2021. Enhancing the student learning experience by adopting TDD and BDD in course projects. In: 2021 IEEE Global Engineering Education Conference. EDUCON, IEEE, pp. 1116–1125.
Rocha Silva, T., Winckler, M., Bach, C., 2020. Evaluating the usage of predefined interactive behaviors for writing user stories: An empirical study with potential product owners. Cogn. Technol. Work 22 (3), 437–457.
Rocha Silva, T., Winckler, M., Trætteberg, H., 2019a. Ensuring the consistency between user requirements and graphical user interfaces: A behavior-based automated approach. In: International Conference on Computational Science and Its Applications. Springer, pp. 616–632.
Rocha Silva, T., Winckler, M., Trætteberg, H., 2019b. Ensuring the consistency between user requirements and GUI prototypes: A behavior-based automated approach. In: IFIP Conference on Human-Computer Interaction. Springer, pp. 644–665.
Rocha Silva, T., Winckler, M., Trætteberg, H., 2020a. Ensuring the consistency between user requirements and task models: A behavior-based automated approach. Proc. ACM Hum.-Comput. Interact. 4 (EICS), 1–32.
Santos, E.C.S., Beder, D.M., Penteado, R.A.D., 2015. A study of test techniques for integration with domain driven design. In: 2015 12th International Conference on Information Technology-New Generations. IEEE, pp. 373–378.
Santos, J.P., Rocha, T., Borba, P., 2019. Improving the prediction of files changed by programming tasks. In: Proceedings of the XIII Brazilian Symposium on Software Components, Architectures, and Reuse. pp. 53–62.
Santos, E.C.d., Vilain, P., 2018. Automated acceptance tests as software requirements: An experiment to compare the applicability of fit tables and Gherkin language. In: International Conference on Agile Software Development. Springer, pp. 104–119.
Sarinho, V.T., 2019. BDD assemble!: A paper-based game proposal for behavior driven development design learning. In: Int International Conference on Entertainment Computing and Serious Games. Springer.
Sathawornwichit, C., Hosono, S., 2012. Consistency reflection for automatic update of testing environment. In: 2012 IEEE Asia-Pacific Services Computing Conference. IEEE, pp. 335–340.
Scandaroli, A., Leite, R., Kiosia, A.H., Coelho, S.A., 2019. Behavior-driven development as an approach to improve software quality and communication across remote business stakeholders, developers and QA: Two case studies. In: 2019 ACM/IEEE 14th International Conference on Global Software Engineering. ICGSE, IEEE, pp. 105–110.
Schneider, M., Hippchen, B., Abeck, S., Jacoby, M., Herzog, R., 2018. Enabling IoT platform interoperability using a systematic development approach by example. In: 2018 Global Internet of Things Summit. GIoTS, IEEE, pp. 1–6.
Schoeneman, L., Liu, J.B., 2013. Integrating behavior driven development and programming by contract. In: International Conference on Computational Science and Its Applications. Springer, pp. 590–606.
Schulz, H., Okanović, D., van Hoorn, A., Ferme, V., Pautasso, C., 2019. Behavior-driven load testing using contextual knowledge-approach and experiences. In: Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. pp. 265–272.
Schur, A., Williams, M., 2017. Augmented system verification using automated testing. In: International Conference on Applied Human Factors and Ergonomics. Springer, pp. 64–70.
Shafiee, S., Hvam, L., Haug, A., Wautelet, Y., 2018. Behavior-driven development in product configuration systems. In: 20th Configuration Workshop.
Silva, T.R., Fitzgerald, B., 2021a. Empirical findings on BDD story parsing to support consistency assurance between requirements and artifacts. In: Evaluation and Assessment in Software Engineering. pp. 266–271.
Silva, T.R., Fitzgerald, B., 2021b. Parsing BDD stories for automated verification of software artefacts. Electron. Commun. EASST 79, 2–9.
Silva, T.R., Hak, J.-L., Winckler, M., 2016. Testing prototypes and final user interfaces through an ontological perspective for behavior-driven development. In: Human-Centered and Error-Resilient Systems Development. Springer, pp. 86–107.
Silva, T.R., Hak, J.-L., Winckler, M., 2017a. A behavior-based ontology for supporting automated assessment of interactive systems. In: 2017 IEEE 11th International Conference on Semantic Computing. ICSC, IEEE, pp. 250–257.
Silva, T.R., Hak, J.-L., Winckler, M., 2017b. A formal ontology for describing interactive behaviors and supporting automated testing on user interfaces. Int. J. Semant. Comput. 11 (04), 513–539.
Silva, T.R., Winckler, M., Trætteberg, H., 2019. Extending behavior-driven development for assessing user interface design artifacts. In: The 31st International Conference on Software Engineering & Knowledge Engineering. SEKE 2019.
Siqueira, F.L., de Sousa, T.C., Silva, P.S.M., 2017. Using BDD and SBVR to refine business goals into an event-B model: A research idea. In: 2017 IEEE/ACM 5th International FME Workshop on Formal Methods in Software Engineering. FormaliSE, IEEE, pp. 31–36.
Sivanandan, S., et al., 2014. Agile development cycle: Approach to design an effective model based testing with behaviour driven automation framework. In: 20th Annual International Conference on Advanced Computing and Communications. ADCOM, IEEE, pp. 22–25.
Snook, C., Hoang, T.S., Dghaym, D., Fathabadi, A.S., Butler, M., 2021. Domain-specific scenarios for refinement-based methods. J. Syst. Archit. 112, 101833.
Snook, C., Hoang, T.S., Dghyam, D., Butler, M., Fischer, T., Schlick, R., Wang, K., 2018. Behaviour-driven formal model development. In: International Conference on Formal Engineering Methods. Springer, pp. 21–36.
So, C., 2017. Intuitive design: Framing a software test system as a status reporting tool for business. J. Eng. Des. Technol. 15 (2), 149–165.
Soeken, M., Wille, R., Drechsler, R., 2012. Assisted behavior driven development using natural language processing. In: International Conference on Modelling Techniques and Tools for Computer Performance Evaluation. Springer, pp. 269–287.
Solis, C., Wang, X., 2011. A study of the characteristics of behaviour driven development. In: 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications. IEEE, pp. 383–387.
Souza, P.L.d., Prado, A.F.d., Souza, W.L.d., Santos Forghieri Pereira, S.M.d., Pires, L.F., 2018. Improving Agile software development with domain ontologies. In: Information Technology-New Generations. Springer, pp. 267–274.
Storer, T., Bob, R., 2019. Behave nicely! automatic generation of code for behaviour driven development test suites. In: 2019 19th International Working Conference on Source Code Analysis and Manipulation. SCAM, IEEE, pp. 228–237.
Tuglular, T., Coşkun, D.E., 2021. Behavior-driven development of software product lines. In: 2021 8th International Conference on Dependable Systems and their Applications. DSA, IEEE, pp. 230–239.
Wanderley, F., Silva, A., Araújo, J., 2015. Evaluation of BehaviorMap: A user-centered behavior language. In: 2015 IEEE 9th International Conference on Research Challenges in Information Science. RCIS, IEEE, pp. 309–320.
Wanderley, F., da Silveria, D.S., 2012. A framework to diminish the gap between the business specialist and the software designer. In: 2012 Eighth International Conference on the Quality of Information and Communications Technology. IEEE, pp. 199–204.
Wang, Y., Degutis, D.R., Wagner, S., 2018. Speed up BDD for safety verification in Agile development: A partially replicated controlled experiment. In: Proceedings of the 19th International Conference on Agile Software Development: Companion. pp. 1–8.
Wang, Y., Wagner, S., 2018a. Combining stpa and bdd for safety analysis and verification in Agile development. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. IEEE/ACM.
Wang, Y., Wagner, S., 2018b. Combining stpa and bdd for safety analysis and verification in Agile development: A controlled experiment. In: International Conference on Agile Software Development. Springer, pp. 37–53.
Wiecher, C., Fischbadh, J., Greenyer, J., Vogelsang, A., Wolff, C., Dumitrescu, R., 2021a. Integrated and iterative requirements analysis and test specification: A case study at kostal. In: 2021 ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems. MODELS, IEEE, pp. 112–122.
Wiecher, C., Greenyer, J., Aydemir, F., Gralha, C., Daneva, M., Groen, E., Herrmann, A., 2021b. BeSoS: A tool for behavior-driven and scenario-based requirements modeling for systems of systems. In: REFSQ Workshops.
Wiecher, C., Greenyer, J., Wolff, C., Anacker, H., Dumitrescu, R., 2021c. Iterative and scenario-based requirements specification in a system of systems context. In: International Working Conference on Requirements Engineering: Foundation for Software Quality. Springer, pp. 165–181.
Wiecher, C., Japs, S., Kaiser, L., Greenyer, J., Dumitrescu, R., Wolff, C., 2020. Scenarios in the loop: Integrated requirements analysis and automotive system validation. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. pp. 1–10.
Wieringa, R., Maiden, N., Mead, N., Rolland, C., 2006. Requirements engineering paper classification and evaluation criteria: A proposal and a discussion. Requir. Eng. 11 (1), 102–107.
Williams, D.M., Darwish, S., Schneider, S., Michael, D.R., 2020. Legislation-driven development of a gift aid system using event-B. Form. Asp. Comput. 32 (2), 251–273.

Winkler, D., Meixner, K., Novak, P., 2019. Efficient and flexible test automation in production systems engineering. In: Security and Quality in Cyber-Physical Systems Engineering. Springer, pp. 227–265.
Wolde, B.G., Boltana, A.S., 2020. Behavior-driven re-engineering for testing the cloud. In: 2020 Seventh International Conference on Software Defined Systems. SDS, IEEE, pp. 75–82.
Wolde, B.G., Boltana, A.S., 2021. Behavior-driven quality first Agile testing for cloud service. Softw. Eng. 9 (1), 9.
Wynne, M., Hellesoy, A., 2012. The Cucumber Book. Pragmatic Programmers, LLC.
Xu, J., Du, Q., Li, X., 2021. A requirement-based regression test selection technique in behavior-driven development. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference. COMPSAC, IEEE, pp. 1303–1308.
Yang, A.Z., da Costa, D.A., Zou, Y., 2019. Predicting co-changes between functionality specifications and source code in behavior driven development. In: 2019 IEEE/ACM 16th International Conference on Mining Software Repositories. MSR, IEEE, pp. 534–544.
Yen, Y.-C., Wang, B.-Y., Zhong, X.-Z., Chiang, W.-Y., Hsieh, C.-Y., Cheng, Y.C., 2021. Applying problem frames in behavior-driven development for smart cone system. In: 2021 28th Asia-Pacific Software Engineering Conference. APSEC, IEEE, pp. 566–567.
Zaeske, W., Durak, U., 2020. Leveraging semi-formal approaches for DepDevOps. In: International Conference on Computer Safety, Reliability, and Security. Springer, pp. 217–222.
Zaeske, W., Durak, U., Torens, C., 2021. Behavior driven development for airborne software engineering. In: AIAA Scitech 2021 Forum. p. 1917.
Zafar, M.N., Afzal, W., Enoiu, E., 2021. Towards a workflow for model-based testing of embedded systems. In: Proceedings of the 12th International Workshop on Automating TEST Case Design, Selection, and Evaluation. pp. 33–40.
Zampetti, F., Di Sorbo, A., Visaggio, C.A., Canfora, G., Di Penta, M., 2020. Demystifying the adoption of behavior-driven development in open source projects. Inf. Softw. Technol. 123, 106311.

Leonard Peter Binamungu received his Ph.D. in Computer Science from the University of Manchester, UK, in 2020. He also has a M.Sc. in Computer Science from Osmania University, Hyderabad, India (2012), and a B.Sc. in Computer Science from the University of Dar es Salaam, Tanzania (2009). He is currently a Lecturer at the Department of Computer Science and Engineering, College of Information and Communication Technologies (CoICT), University of Dar es Salaam (UDSM). He mainly teaches, researches and consults in the areas of Software Engineering and Information Systems. His specific areas of research in software engineering include Behavior-Driven Development, agile testing, mining software repositories, software maintenance, and empirical software engineering.

Salome Maro is a Lecturer at the Department of Computer Science and Engineering, College of Information and Communication Technologies (CoICT), University of Dar es Salaam (UDSM). Her main areas of research interest are software traceability, software development processes and program comprehension. She has a Ph.D. in Computing Science from the University of Gothenburg, Sweden (2020), a M.Sc. in Software Engineering from the University of Gothenburg, Sweden (2014) and a B.Sc. in Computer Science from the University of Dar es salaam (2010).