Data & Knowledge Engineering 144 (2023) 102111

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

A conceptual model supporting decision-making for test automation in Agile-based Software Development

Shimza Butt [a,*], Saif Ur Rehman Khan [a], Shahid Hussain [b], Wen-Li Wang [b]

[a] *Department of Computer Science, COMSATS University Islamabad (CUI), 45550 Islamabad, Pakistan*

[b] *Department of Computer Science and Software Engineering School of Engineering, Penn State University, Behrend, USA*

A R T I C L E I N F O

Keywords:
Test automation
Agile Software Development
Conceptual model
Decision-making
Empirical investigation
Systematic literature review

A B S T R A C T

With the rise of agile development notion, software test automation is widely adopted by software organizations to reduce the testing time and costs during the software development process. However, it is essential to carefully implement the test automation to ensure the project's success. In the literature, several studies have indicated that decision-making is crucial for the adoption of test automation. For deciding about test automation adoption, several factors, including the skill level of testers, test tool, and development process, need to be considered since automated testing performance heavily depends on the mentioned factors and their interactions. Thus, it is of vital importance to ascertain the factors that influence test automation in the context of Agile-based Software Development (ASD). To accomplish this, we perform a systematic literature review and a questionnaire-based survey. In total, 21 factors are identified that significantly impact test automation in the ASD context. Additionally, the identified factors are categorized into five main classes. Following the identified factors' assessment, this work proposes a conceptual model that supports agile practitioners in deciding about the adoption of test automation in the context of ASD. Finally, this work validates the proposed model to assess its practical applicability. The validation of the proposed model is accomplished by using an expert-based validation approach. The attained results show that the proposed model supports agile practitioners in estimating the possibility of success when implementing test automation in the ASD context.

1. Introduction

Test automation is widely employed by software organizations with the underlying intention to reduce the cost of manual testing, which is regarded as a time-consuming and laborious phase of Software Development Life-Cycle (SDLC) [1]. Also, it is essential to facilitate to gauge in the international market [2]. Software testing and the test engineer's roles in product development constantly evolves; thereby, traditional software testing methods fail to meet the product's ongoing testing demands. The significance of automated testing in software engineering is greatly increased in the recent years, which can be seen through a growing number of organizations investing in automated testing tools to avoid defects during the development phase of SDLC [3]. Automated Testing is defined as "the use of a separate software from the testable application to control and execute test cases against defined specifications" [4].

According to the reported studies [5,6], test automation adoption by software organizations is generally declared to be 72%, and the total test case automation percentage is mentioned to have risen from 28% to 45% in a year (from 2014 to 2015). According to

* Corresponding author.
E-mail address: shimzabutt.96@gmail.com (S. Butt).

https://doi.org/10.1016/j.datak.2022.102111
Received 9 April 2022; Received in revised form 4 October 2022; Accepted 20 November 2022
Available online 23 November 2022

the ISTQB yearly test reports [7], test automation in 2015–2016 was 58.5%, and considerably increased to 64.4% in years 2017–2018. The global automated software testing market is supposed to reach $55 billion in 2022, as mentioned in [8]. Notice that the decision on the testing stream (i.e. manual or automated) heavily depends on the project's nature, management intentions, and available resources [9]. The test automation has several advantages in terms of resources (i.e. time and money) [10] and efforts required in test executions [2]. Furthermore, test automation accelerates the execution phase and provides a formal test coverage [1]. However, there are some disadvantages and challenges associated with test automation while scaling agile framework, as highlighted in the reported studies [11–15]. In the literature, the researchers have mentioned some challenges with test automation in agile context such as, training, high investment for setup and tool selection [2], and the higher management's specific interest in measuring its impact on overall cost, quality, and development time of software [1].

Agile-based Software Development (ASD) is regarded as a widely-employed development methodology, with agile approaches being adopted in 21% of development projects in 2013, as mentioned in a prior study [16]. Another study [17] has highlighted the importance of collaboration between developers and testers for successful test automation in ASD. For a fast-paced agile projects, testing of software products at every stage of agile development is required mainly due to rapid modifications in requirements and development [2]. Test automation is assumed to be a pre-requisite for a modern agile development team [4] as it accelerates the quality assurance process [3]. The concept of agile testing implies shifting towards automated testing [18]. Inspired by this, several fully automated software testing tools have been developed. However, automated testing is a crucial process; hence, several considerations must needs to be determined, such as which feature requires automated testing before any organization continues for test automation [1]. As automation does not come up for free; thereby, it must be carefully implemented to ensure the project's success [19]. Hence, high initial investments are required for test automation in scripting, tool procurement, training, software feature analysis, and so on. Thus, Return on Investment (ROI) must be determined before starting test automation [1].

Due to the challenges of automated testing, its proper implementation is rare [2]. Hence, the practitioners believed that automated testing is impractical for all possible situations [9]. Many software development organizations invest in test automation but fail to acquire the expected benefits [20]. More than 60% of IT organizations used and spent millions of dollars on developing automated testing tools; however, their final product(s) lack in meeting the success due to the selection of a wrong testing tool in the considered testing environment, which results in a complete project failure [6]. Thus, extensive planning is required in the context of automated testing. As poor decisions often lead to severe disappointments, they can result in high erroneous expenditures in terms of resources and efforts. Since this situation can jeopardize the successful completion of software projects within the allocated development resources. Therefore, it is required to investigate the following questions, including how much automation is enough? In the context of a given project, when and what should you automate? What prerequisites should be met before the test engineer automates software testing? [19].

In summary, the above-mentioned issues highlight the necessity of integrating agile-based development with automated software testing [2]. Due to this, there is a high demand to address the challenges, which requires high collaboration between agile and test automation. Moreover, various means to fully use agile practices and automation tools are required to attain the utmost benefits for the software industries [21]. On the other hand, the testing costs and time can be significantly reduced if test automation is properly implemented while improving the overall quality of a software product [22]. Generally speaking, automated software tests must be conducted at the proper time with the appropriate set of resources [2]. Therefore, it is required to consider the important factors while planning for test automation, which ultimately supports effective decision-making [18,23].

Motivated by this, current work aims at identifying and validating the factors that impacts on successful adoption of test automation in the context of ASD. Note that the validation of factors focuses on obtaining the viewpoint of practitioners involved in the adoption of test automation in ASD. In the targeted research context, decision-making includes both test planning and risk management to ensure the successful deployment of test automation. However, it remains a challenging and complex task since the agile framework involves little planning [9]. The inability to incorporate automated testing in an agile life-cycle is one of the leading causes of a project failure [16]. Therefore, the current work proposes a conceptual model that considers the relevant factors essential in decision-making process for test automation in ASD context. The main contributions of this study are:

- Identification and categorization of factors affecting successful test automation in agile-based development through a systematic literature review protocol.
- Analysis of the identified factors from the industry practitioners to provide the state-of-the-practice viewpoint.
- Propose a conceptual decision-making model for test automation in agile software development based on the identified classes of factors.
- Validation of the proposed conceptual model to assist the decision-makers in adopting test automation in agile development context.

The impediments [24] and factors affecting the decision-making questions [22,25,26] related to automated software testing are identified in the existing literature. However, to the best of our knowledge, this study differs from the current state-of-the-art in the following ways:

(a) The existing work lacks in systematically identifying the test automation factors, particularly in agile development environment. Moreover, there is no categorization of the factors affecting test automation in the context of ASD.

(b) Also, there is a need to validate the factors from the practitioners' perspective to gain a state-of-the-practice viewpoint.

(c) Still, there is no work that supports the decision-makers by providing a mechanism useful in estimating the possibility of success for test automation in agile development.

(d) The existing literature lacks in providing a conceptual model that considers all relevant factors to facilitate in decision-making for test automation in the ASD environment.

Table 1
Summary of related work.

Author(s)	Description of research	Methodology	Limitations
Wiklund et al. [24]	• Identified impediments	SLR	• The majority of reported impediments are of organizational related rather than technical in nature • Lacks in validating of the identified impediments • Lacks in focusing on the targeted research domain (i.e. ASD)
Garousi and Mäntylä [22]	• Identified and classified factors	MLR	• Lacks in validating the provided checklist • Does not focus on the targeted research domain
Polepalle and Kondoju [25]	• Identified 16 testability requirements, 15 factors triggering additional effort in GUI test automation, and 12 criteria for when to begin automation	Multi-case study (literature review and interview)	• Targeted only one specific company for the validation of the identified criteria • Lacks in focusing on the targeted research domain
Lindholm [26]	• Validated and modified a checklist • Established a decision tree for the selection of test cases for test automation	Literature review and interview practitioners	• Targeted only experts of one specific company for the validation of factors and hence evaluation of the decision tree • Focused only on factors related to the selection of test cases for automation • Lacks in focusing on the targeted research domain

The remaining part of this paper is organized as follows: Section 2 describes the related work, while the research methodology adopted in this study is reported in Section 3. The findings and analysis of the study are described in Section 4. Section 5 presents the proposed conceptual model, while Section 6 validates the proposed model. The detailed discussion concerning each research question is provided in Section 7. Section 8 reports the threats to validity about current study, while research implications are described in Section 9. Finally, Section 10 concludes this work, and outline the potential research pointers.

2. Related work

This section describes the reported work related to the identification of influencing factors that affects test automation in the context of ASD. Moreover, some studies have focused on proposing decision-making models/techniques for test automation in the ASD context.

Wiklund et al. [24] performed a Systematic Literature Review (SLR) and identified impediments to software test automation. The majority of the reported impediments are related to organizational nature, including behavioral, lack of time, people and funding, and so on. However, the authors lack in validating the identified impediments. In contrast, Garousi and Mäntylä [22] performed a Multi-Vocal Literature (MLR). The authors identified the factors related to test automation decision-making. They categorized the factors into five categories including software under test, test-tool, human and organization, cross-cutting, and test-related factors. Moreover, they also suggested classifying factors such as the project level, test case level, or business level for future work. However, their study lacks in providing the practitioners' viewpoint regarding the identified factors.

In comparison to the reported studies [22,24], Polepalle and Kondoju [25] conducted a multiple-case study (i.e. the combination of literature review and interviews) to examine the practitioners' viewpoints employing GUI-automation testing in the telecommunication industry. The authors recognized 16 testability requirements, 15 factors triggering additional effort in GUI test automation, and 12 criteria for when to begin automation. They highlighted the need to validate the applicability of the results for general software test automation, and suggested a classification of the identified variables by different levels of testing for future research work. Additionally, they mentioned the ranking of the identified factors to direct the practitioners' emphasis on high priority factors. Moreover, they emphasized on investigating the effect of Agile, DevOps, and continuous deployment on the decision-making process. However, their study lacks in validating the provided checklist.

In contrast, Lindholm [26] performed a literature review study. The author interviewed the practitioners to validate and modify the checklist provided in [22]. The main underlying intention was to establish a method that facilitates the test case's selection for automation purposes. After modifying the checklist, the factors considering only test case selection for automation were used to evaluate the decision tree. The results indicated that the method provides economic and organizational benefits in the industry. However, they lack in identifying important factors, particularly in the context of agile-based software development. Moreover, they considered the opinion of experts (of only one company) to prioritize the factors and evaluated the decision tree. Consequently, it arises serious validity threats about the generalizability of their results.

Table 1 summarizes the related work reported in the literature.

From Table 1, it is evident that the researchers have attempted on finding a limited set of factors affecting automated software testing. Moreover, they focused on the factors specifically selection of test cases for automation. However, the limitations in the existing literature also necessitate validation of the identified factors from the practitioners, particularly in the ASD context.

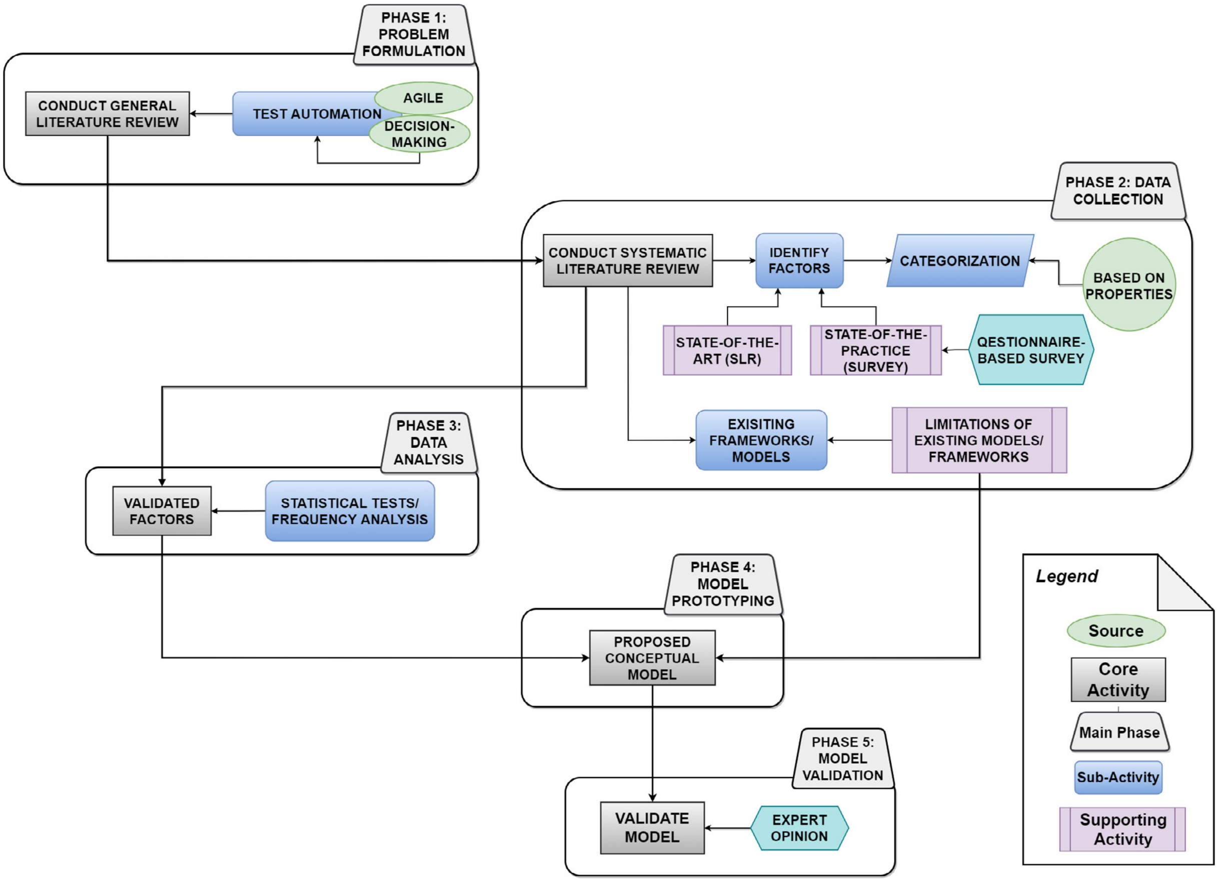

Fig. 1. Adopted research methodology.

3. Research methodology

This section describes the research methodology adopted for this work. The current study primarily focuses on the qualitative research method to propose a particular solution in the ASD context. Note that the qualitative research method relies on collecting and analyzing data obtained through interviews and questionnaires. Fig. 1 illustrates the adopted research methodology for current study. It mainly consists of five phases namely problem formulation, data collection, data analysis, model prototyping and model validation (Fig. 1).

A general literature study is carried out during the problem formulation phase in the targeted research context, i.e. decision-making for test automation in agile development. Following this step, data is gathered using various approaches with the underlying intention of obtaining the information about the factors affecting test automation in the ASD context. For this purpose, a systematic review of the literature and a survey (i.e. a questionnaire-based) are used. In the data analysis phase, a comparison is performed between the findings and outcomes of the review and survey using several statistical tests. This step generated a list of 21 factors, which were then categorized into five elements (classes) according to their properties. The elements indicate the primary categories. In contrast, the factors describe the essential details about each element, i.e. must be considered by the agile practitioners (decision-maker) while planning and analyzing the risk associated with successful test automation adoption. After completing the data analysis phase, a conceptual model grounded on the identified factors is developed during the model prototyping phase. Finally, phase five encompasses validating the proposed conceptual model. Moreover, expert validation (also known as expert opinion) approach has been adopted to validate the proposed conceptual model. In expert validation, the software industry experts are selected to validate the proposed conceptual model. Finally, the experts decide to reject, accept, or review the proposed model.

3.1. Systematic Literature Review (SLR)

This section provides the detailed information about the SLR's design and execution. The main motivation for conducting a SLR is that it provides an unbiased view and synthesize the reported work about a particular research area. In this work, we have followed the widely adopted SLR's guidelines as suggested by kitchenham et al. [27]. According to kitchenham et al. [27], a SLR comprises

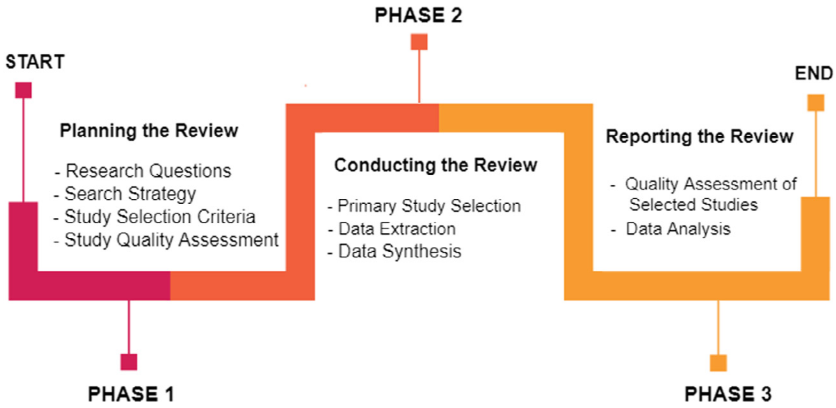

Fig. 2. The core phases of SLR.

three core phases: (i) planning, (ii) conducting, and (iii) reporting the review. Fig. 2 highlights the core phases and sub-phases of a SLR. The main motif of the performed SLR was on finding the factors affecting test automation in the context of ASD.

For planning phase of SLR (Fig. 1), we developed a research plan to devise the research questions, keywords, search string and repositories, and inclusion/exclusion criteria for selecting potential research articles.

3.1.1. Research questions

Based on the targeted research objective, we devised the following Research Questions (RQs):

RQ-1: What are the factors affecting test automation in agile development context mentioned in the existing literature?

- **RQ-1.1:** How to categorize the identified factors in the context of agile-based development?
- **RQ-1.2:** What are the critical factors affecting test automation in agile-based development?
- **RQ-1.3:** What are the current state-of-the-art decision-making methods, tools, or frameworks for test automation in ASD?

RQ-2: How to provide decision-making support for test automation in agile-based development?

RQ-3: What is the practical applicability of the proposed decision-making model for test automation in agile-based development?

3.1.2. Search strategy

In this step, a search string and repositories are identified and selected (respectively) to find the relevant studies.

(a) Search String

The query used to collect the data for the included studies is given below and was tailored to the selected databases. The search string was defined systematically. First of all, we determined the main keywords based on the formulated research questions. Next, the synonyms of the main keywords were identified. After that, the main keywords along with their synonyms were integrated using Boolean operators "AND", "OR". Finally, the formulated search string was applied in each selected source repository's titles, abstracts, and keywords. The formulated search string is mentioned as follows:

("Test Automation" OR "Automated Testing" OR "Software Test Automation") **AND** ("Decision Support" OR "Decision Making") **AND** ("Agile" OR "Agile Software Development" OR "Agile Development" OR "ASD" OR "SCRUM") **AND** (("Factors" OR "Impediments" OR "Criteria") **OR** ("Tools" OR "Framework" OR "Method"))

(b) Search Repository

In this work, different source repositories were chosen to identify the relevant articles. The current study retrieved the potential publications by using the following widely-used digital source repositories:

- ACM
- Science Direct
- Wiley Online Library
- IEEE
- Springer Link

3.1.3. Study selection criteria

A criteria to select the relevant studies was identified in this step. It includes two types of criteria, specifically inclusion and exclusion criteria, as illustrated in Table 2. The inclusion criteria are devised in accordance with the studies focusing on the targeted domain. Also in a SLR, recent studies are focused. Although in the current literature, the authors of a study [22] have conducted a multi-vocal literature review in 2015. Therefore, to provide the latest literature view, the inclusion criteria (IC4) focused on the studies published from 2015 to 2021.

Table 2
Study selection criteria.

Inclusion Criteria (IC)	Exclusion Criteria (EC)
IC1: The potential study published in peer-reviewed journals and conferences.	EC1: The study not explicitly focusing on test automation in agile-based software development is excluded.
IC2: Books discussing topics related to test automation in agile-based software development are also considered while studying the literature.	EC2: The study in which the full content is inaccessible is excluded.
IC3: The studies discussing factors affecting test automation are included.	EC3: The studies that are not written in English are excluded.
IC4: The studies published in years from 2015 to 2021 are included.	EC4: The duplications of similar results were also eliminated.

Table 3
Quality assessment questions.

QA criteria	Checklist questions
QA1	Is the research method well-defined and well-documented?
QA2	Is there a specific focus on test automation in agile software development in this study?
QA3	Does the study discuss any factor affecting test automation in agile-based software development?
QA4	Does the study discuss any tool, framework, or method for decision-making to test automation?
QA5	Does any of the stated research question addressed?
QA6	Does the study involve empirical evaluation of the proposed solution?
QA7	Does the study have any future implications?

3.1.4. Study quality assessment

Quality evaluation of the potential articles was performed alongside the point of extracting the data. A checklist of Quality Assessment (QA) criteria was created to evaluate the preferred research articles, as shown in Table 3. For every given point QA1 to QA7, the assessment was conducted as follows:

- One point was given to the studies that focused on the complete checklist questions.
- A score of 0.5 was given to the studies that had only a few answers to the checklist questions.
- the studies that did not answer any of the checklist questions received a score of zero.

3.2. Questionnaire-based survey

The current section describes the practitioners' measures and approaches for data collection in the questionnaire-based survey. Moreover, the participants' selection process is provided. Note that the survey design is based on designing the questionnaire and selecting participants for the responses. In contrast, data analytical approach involves an approach used for analyzing the data obtained from the survey responses. The factors identified from the existing literature were further validated to gain the perspective from project managers, software quality assurance engineers, or test automation engineers involved in agile-based software development.

3.2.1. Measures and procedures for data gathering

In this work, a questionnaire-based survey is conducted to obtain data from the practitioners based on the factors identified from the literature. Generally, the surveys are conducted to obtain a detailed picture of the present condition. In addition, surveys provide data, i.e. difficult to get using observational approaches. In this study, the practitioners were surveyed to obtain a more thorough investigation. The questionnaire included closed-ended questions to collect data from the target group, who had prior experience working in the ASD domain. The survey questionnaire was developed using Google forms[1] and involved two main sections. The first section of the questionnaire focused on the respondent's demographic and organizational-related information, while the second section covered test automation elements. The items in the questionnaire were graded on a five-point Likert scale, with one representing "strongly disagree", 2 representing "disagree", 3 representing "neutral", 4 representing "agree", and five representing "strongly agree". Experts conducted face validity and content validity tests to assess the survey questionnaire's reliability and correctness. Face validity entailed specialists examining the items listed in the questionnaire and determining whether or not respondents could grasp the questions. The experts who assessed the questionnaire items' completeness, accuracy, and readability were involved in content validity. It was done to get a consensus on which items should be kept in the final questionnaire. Based on the feedback of the experts, the questionnaire was reviewed, and improvements were made.

[1] https://forms.gle/UTpKyKkv4Xy3bZ5w8.

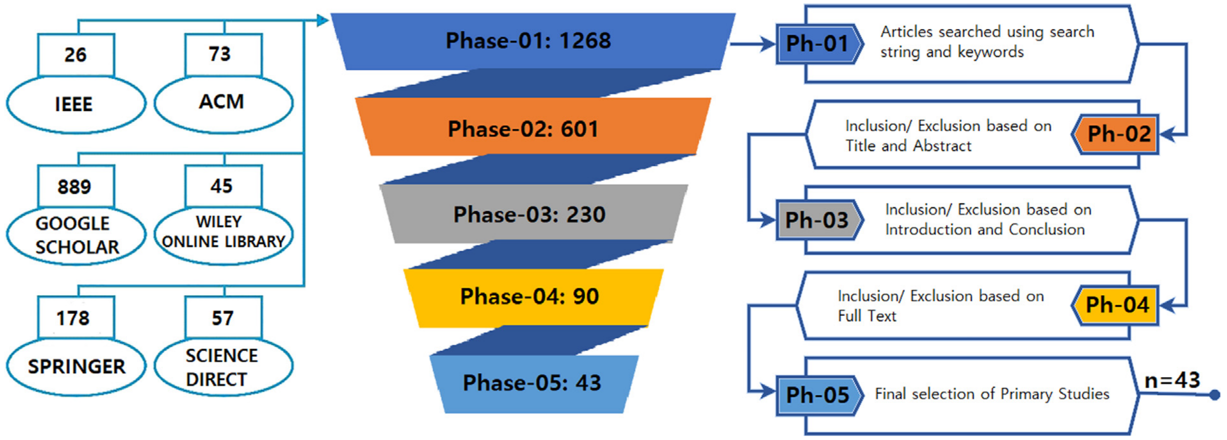

Fig. 3. Tollgate approach for selection of studies.

3.2.2. Participants

After assessing the survey questionnaire's reliability, the participants were approached using professional networking platform, i.e. LinkedIn. According to the study, the inclusion criterion of sample selection as participants of this study were all IT professionals involved in decision-making, testing, and ensuring the quality of ASD projects or had any expertise related to testing in the context of ASD [20]. First of all, the IT professionals were contacted through LinkedIn, and requested to take part in the research. Next, the questionnaire was provided to them through LinkedIn message based on their consent. Moreover, the practitioners were informed that the information gathered through the survey questionnaire would not be shared with anyone else and would be kept private. Note that we have collected the data from 16 May 2021 to 01 Aug 2021 from the participants on LinkedIn working in ASD projects. A reminder was also sent two weeks after the questionnaire was distributed. By the start of August, a total of 136 responses had been received. We have considered all received response as similar to previously conducted primary studies [28,29]. However, they considered a sample size of <100 related to a specific targeted population [28,29] that raises the concerns about generalizability of the results. In this work, we have targeted considerable number of agile practitioners (i.e. 136) to handle this validity threat.

4. Findings and analysis

This section provides the findings and conclusions for each of the devised research questions. The data from the selected primary studies were used to answer the research questions. Furthermore, the legitimacy of the produced results is assessed by analytical and empirical examination. Firstly, the findings of the SLR are discussed in detail in this section. After that, the findings of the questionnaire-based survey are discussed. Finally, a comparison between SLR and empirical research is provided.

4.1. Findings of SLR

This section explores the findings of SLR in detail. Moreover, it provides the analysis of the selected primary studies, identification and categorization of factors, and also reviews existing decision-making methods, tools, or frameworks.

4.1.1. Primary study selection

The tollgate approach was used to filter the research articles obtained during the initial study selection. Fig. 3 illustrates the selection process of potential primary studies.

Initially, 1268 articles were obtained from the selected online databases by employing the created search strings and the inclusion/exclusion criteria (Fig. 3). Google Scholar was used to verify any missing studies after selecting from other selected databases and applying the snowballing technique. Moreover, the duplications of the similar results were also eliminated. After implementing the tollgate approach and forward snowballing, we have shortlisted 43 potential articles. Notice that snowballing is a method used to search important articles relevant to the targeted context. Forward snowballing refers to considering citations of already identified primary studies. Finally, the shortlisted papers were evaluated using the devised quality assessment criteria.

4.1.2. Data synthesis

The articles were synthesized based on the parameters, i.e. primary study id, reference number, author name, research title, study type, databases, publication year, and publication venue associated with each article. The selected list of articles is reported in Appendix A. Moreover, the selected articles were analyzed on the basis of publication year and venue along with the type of studies.

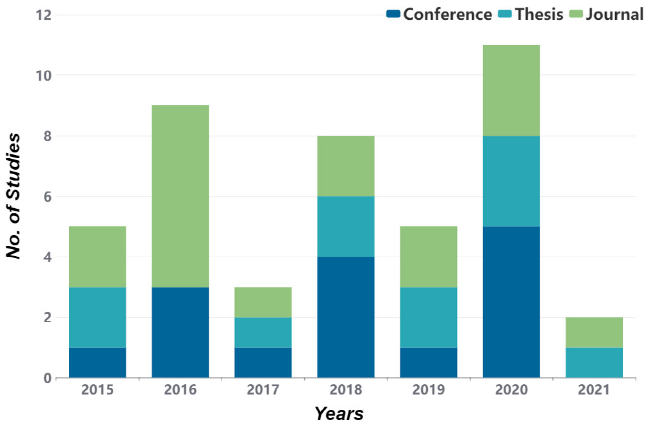

Fig. 4. Temporal distribution of selected articles.

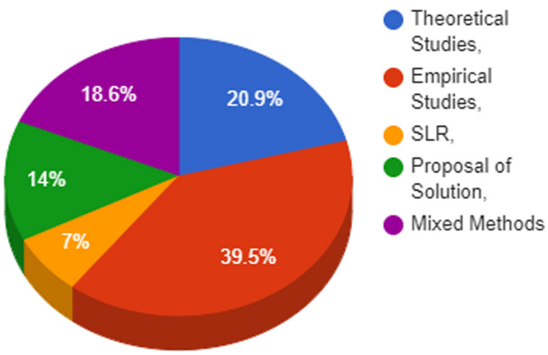

Fig. 5. Types of study adopted by selected articles.

(a) Temporal Distribution of Selected Articles

In this step, the temporal distribution of primary studies was used to analyze the data along with the publication venue of the selected studies. The selected primary studies were published from the years 2015 to 2021. The distribution of primary studies throughout time, along with the publication venue, is shown in Fig. 4. Out of the 43 selected studies, 11 (25.58%) of the selected articles were published in 2020. It indicates the growing publication trend of automated testing in 2020 according to the reported literature. The growing trend also validates the statistics of organizations investing in automated testing tools, highlighting the importance in academia and industry.

(b) Type of Studies

It was noticed that 43 selected studies were sorted into five types of studies according to the study [30], i.e. SLR, empirical study, theoretical studies, the proposal of a solution, and studies with mixed methods. Fig. 5 depicts the distribution of research methods used in the existing literature. As a result, 17 of them were empirical studies. Data collection strategies involved in the empirical studies include case studies, questionnaire-based surveys, and interviews. In contrast, three primary studies were systematic literature reviews, type of literature review that uses a structured approach to gather secondary data. Furthermore, nine of them were theoretical studies. These studies were utilized to analyze data from citations and practices from earlier studies in the literature. Besides, six of the primary studies belong to the category of studies that involve presenting a theoretical or conceptual framework and assessing the study's outcomes. The category is named 'proposal of a solution'. In addition, eight primary studies involved mixed methods, a combination of two or more of the methods mentioned above.

4.1.3. Quality assessment of the selected articles

Lastly, the potential studies were evaluated by applying the quality assessment criteria. The quality evaluation of the primary studies is displayed in Appendix B.

4.1.4. Identification of factors

The current section provides the factors affecting test automation in agile software development, identified through SLR. For the identification of factors, the articles identified through SLR were studied and the factors were extracted based on the suggestions of the authors. Additionally, the frequency analysis method is used for analyzing the data related to factors identified through SLR.

Table 4
Frequency analysis of the factors identified through SLR.

Factor id (TA-#)[a]	Factors	Frequency	RP (100-Scale)	References (PS-#)
TA-1	Maturity of SUT	8	67.5	PS-2, PS-8, PS-10, PS-32, PS-36, PS-27, PS-9, PS-42
TA-2	SUT life-cycle	3	55	PS-8, PS-11, PS-32
TA-3	SUT customizability	2	52.5	PS-8, PS-11
TA-4	SUT complexity	3	55	PS-8, PS-9, PS-36
TA-5	Test tool	21	100	PS-1, PS-3, PS-4, PS-5, PS-6, PS-8, PS-10, PS-11, PS-14, PS-16, PS-17, PS-24, PS-28, PS-29, PS-30, PS-31, PS-32, PS-33, PS-37, PS-38, PS-41
TA-6	Technical/Environmental	10	72.5	PS-1, PS-8, PS-9, PS-10, PS-27, PS-28, PS-30, PS-31, PS-35, PS-37
TA-7	Test process	12	77.5	PS-4, PS-6, PS-8, PS-10, PS-19, PS-27, PS-29, PS-30, PS-35, PS-36, PS-37, PS-39
TA-8	Need for regression testing	3	55	PS-1, PS-8, PS-11
TA-9	Test type	4	57.5	PS-1, PS-8, PS-11, PS-32
TA-10	Test Re-use/Repeatability	7	65	PS-8, PS-9, PS-11, PS-22, PS-27, PS-30, PS-42
TA-11	Test importance	5	60	PS-1, PS-4, PS-8, PS-11, PS-37
TA-12	Test oracle	5	60	PS-8, PS-9, PS-11, PS-25, PS-36
TA-13	Test stability	6	62.5	PS-8, PS-9, PS-11, PS-25, PS-31, PS-37
TA-14	Skill level of testers	18	92.5	PS-1, PS-6, PS-7, PS-8, PS-9, PS-10, PS-15, PS-19, PS-20, PS-24, PS-25, PS-27, PS-28, PS-31, PS-35, PS-37, PS-40, PS-43
TA-15	Behavioral	10	72.5	PS-2, PS-8, PS-9, PS-18, PS-21, PS-23, PS-27, PS-31, PS-32, PS-43
TA-16	Staffing	4	57.5	PS-9, PS-10, PS-32, PS-36
TA-17	Steering	5	60	PS-7, PS-8, PS-9, PS-11, PS-37
TA-18	Top management support	9	70	PS-1, PS-2, PS-8, PS-9, PS-11, PS-28, PS-29, PS-35, PS-36
TA-19	Economic	13	80	PS-1, PS-4, PS-8, PS-9, PS-10, PS-11, PS-25, PS-27, PS-30, PS-32, PS-35, PS-36, PS-39
TA-20	Automatability of testing	14	82.5	PS-1, PS-2, PS-8, PS-9, PS-10, PS-11, PS-22, PS-26, PS-32, PS-35, PS-36, PS-37, PS-39, PS-42
TA-21	Development process	11	75	PS-1, PS-4, PS-8, PS-9, PS-10, PS-11, PS-24, PS-32, PS-37, PS-39, PS-42

[a]TA is used as an abbreviation for Test Automation.

A frequency table is generated to represent the data frequency. A list of factors was generated using the information obtained from the 43 articles finally selected. A total of 40 out of 43 articles had the focus on the factors, while others specifically targeted existing decision-making methods, tools, or frameworks for test automation in ASD.

The method of frequency analysis used here is based on how many times the factor is discussed in literature, which ultimately results in calculating the frequency of that factor. Twenty-one factors affecting test automation in agile software development are identified from the previously cited literature. After frequency analysis of factors, the Relative Percentage (RP) of each factor was calculated based on a 100-scale relative percentage of the occurrence. Table 4 illustrates the list of identified factors along with the data frequency. The labels of Table 4 include the factor id, factor name, frequency, relative percentage of the occurrence, and the reference of the primary studies.

To analyze the significance of each factor, the criteria of frequency >70% as critical factors, frequency between 50 and 70% as factors with moderate impact, and frequency <50% as the least significant factors have been adopted. Among 21 identified factors, nine were found to have a critical impact on test automation according to the existing literature using the criteria mentioned above. These critical factors are TA-5 to TA-7 (test tool, technical/environmental, test process), TA-14 (the skill level of testers), TA-15 (behavioral), and TA-18 to TA-21 (top management support, economic, automatability of testing, and development process), respectively. The remaining 12 identified factors lie under the moderate impact on test automation as the frequency was between 50 and 70%. Moreover, none of the factors was categorized as least significant factors based on the above-mentioned criteria.

4.1.5. Categorization of factors

A total of 21 factors affecting test automation adoption were identified from the literature and further categorized into five categories (as shown in Fig. 6) based on their types and properties. The proposed conceptual model utilized the Factors Contribution Analysis (FCA) approach and the approach is based on the elements and factors that contribute to achieving the targeted objective, such as successful test automation. The elements indicate the primary categories, while the factors describe essential details about each element. Therefore, the classification of factors assists in the application of FCA approach. The main categories were further divided into different sub-categories. The factors along with their categories are further discussed in the following sections:

Software Under Test related factors

Software Under Test (SUT) properties have significant impacts on automation decisions. The most important in this category of factors is the maturity of SUT, which refers to product stability. In other words, SUT experiences major changes due to the re-implementation of new functionalities. Therefore, it would negatively affects on the benefits of automated testing as it will require

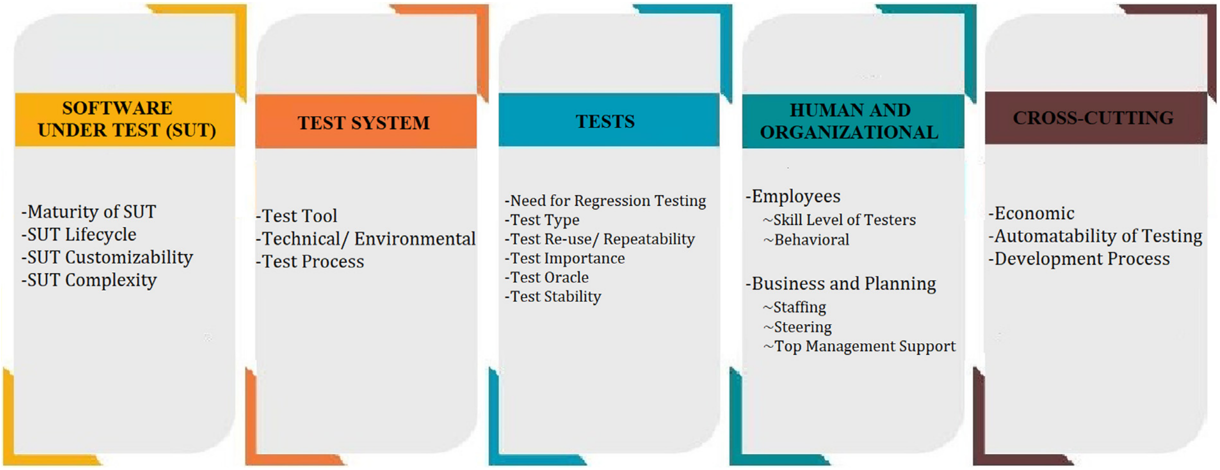

Fig. 6. A thematic taxonomy of the identified factors.

major efforts related to testing co-maintenance or test repair [22]. It is considered vital, but it does not stop one from automation as it depends on the impact of the change on the product [25]. Also, automation is recommended for stable applications. But if there is a small change, one can continue with automation, simultaneously resolving the issues. Other factors in this category are the properties of SUT that affect the automated testing decisions, such as SUT complexity due to its dependency on other products, SUT customizability, and SUT life cycle. These factors must be considered to make the best decision about SUT-related factors.

Test system related factors

A test system is created by combining test tools with a testbed. Notice that a testbed is defined as "an environment containing the hardware, instrumentation, simulators, software tools, and other support elements needed to conduct a test" used by testers, as well as automated test scripts used to test the application [24]. The automation delivers little or no value to the testers if the tools and test systems generate issues.

(a) Test Tool

An important factor identified for automated testing decisions is selecting and adopting suitable automation test tools before the beginning of the test automation process [22]. If a sufficient test automation tool for a certain situation is missing, and developing such a tool in-house is not practicable, test automation should be avoided. Though an additional effort is required to find and examine various available automation tools, it reaps benefits in the long run as compatible tools create tests effectively and quickly [25]. Therefore, one must be familiar with the test tool and its compatibility with the SUT.

(b) Test Process

The identified factor considers several aspects that greatly impact automation decisions. The scope of the test automation process should be well-defined and followed by all the teams before beginning automated testing to ensure success with test automation [25]. Structure of test suite architecture also has its importance as poorly structured test suite requires additional effort for maintaining the test scripts [25] and results in a risk to the quality of the product being tested [24]. Proper test scripting standards should be created and followed throughout the test process. Test data availability and manual testing before automation testing, specifically in the case of GUI testing, are also considered important aspects under this category of factors.

(c) Technical & Environmental

Limitations in test tool functionality is one of the important aspects considered under this category of factors. There could probably be issues with using generic or open-source tools if the application is customized or experience frequent modifications, which make the tool inadequate for the intended purpose and may require modification to become useful [24]. Therefore, this could significantly impact the organization's efficiency and the benefits of automated testing. Factors associated with the test environment, such as untested test environment and configuration management [25], also significantly impact automated testing decisions. If the test system's quality is poor, it will cost more to maintain and lower testing quality. The result is likely to be a decrease in the overall impression of test automation's value.

Tests related factors

The cost of designing tests often rises as test automation is used, but the cost of re-running a test falls [22]. Understanding the number of test re-runs required and the effort involved in test creation and maintenance is critical. Test cases and test suites have their characteristics that influence automation options. The identified category of factors includes several sub-factors to be considered for decision-making which are discussed below.

(a) Need for Regression Testing

It was observed that frequent regression testing is considered an essential or beneficial factor in test automation decisions due to the increase in the continuous and rapid releases of software.

(b) Test Type

Certain types of tests, such as performance and load testing, are suitable candidates for test automation, whereas others, such as user experience tests, are not. Due to this fact, test type is considered an important factor for the decision-making of test automation.

(c) Test Reuse/Repeatability

Regression testing is also influenced by characteristics such as test reuse and repeatability. They can refer to situations in which the same test might be reused as part of another test due to similarity. It needs to be done in several environments to improve test repeatability.

(d) Test Importance

Another factor stated in the literature for decision-making is test importance, which addresses the aspect of proper test-case selection; i.e. tests covering the most significant aspects of the SUT are extremely vital for user satisfaction; hence they are urged to be automated. The possibility of the test uncovering faults is another aspect of this factor.

(e) Test Oracle

The mechanism by which defects are identified, known as the test oracle, is also crucial for decision-making. Test oracle, which is not stable and predictable is not preferred to be automated as the automated comparison in such cases will be fragile, resulting in many false positives.

(f) Test Stability

Another important element to consider in this area is test stability because automating an unstable test is not a good option.

Human and organizational factors

The skill level and behavior of the employees and staffing, steering, and top management support related to organizational change are core components for the test automation decisions.

Employees

The identified category of factors includes factors related to the behavior and skill level of employees.

(a) The Skill Level and Expertise of Testers

Test automation is a complex task that requires different and often additional skills as replacing the manual work with automation will result in a significant change [24]. Therefore, it is important to consider the skill level and expertise of testers before beginning automated testing.

(b) Behavioral

It is observed that the behavior of the people working on a test automation project has a considerable impact on the project's outcome [24]. Resistance to the change against software test automation must be considered before beginning automated testing. The reasons for resistance could be fear of redundancy due to automation, old habits of doing manual testing [3,17], shifting blame between testing and development, and a low learning curve [4]. Too high expectations are also an important aspect under this category as the management of unrealistic expectations might lead to the termination of the automation. It may have a negative impact on the perceived value of the automation if the intended outcomes are not obtained within the predicted time frame [24]. Unrealistic expectations are largely the result of a lack of understanding of what automation can and cannot accomplish.

Business and Planning

Many firms generally overspend on software development, so there is no surprise that cost is seen as a barrier to automation [24]. Moreover, it is vital to recognize that the adoption of test automation is, at least in part, about undertaking a risky organizational change [22]. As a result, to effectively include this factor in decision-making, the introduction of automation must be well managed to reduce negativity that could lead to the automation project's failure.

(a) Staffing

Dependency on key people is the major subject considered in this category of factors as changing roles requires training, and it ultimately is associated with additional costs [31]. It is an organizational change that can disrupt development and may require testing by non-testers [24].

(b) Steering

The identified factor is related to organizational maturity, which involves the need for proper change management, i.e. the ability to control SUT changes [22] through an adequate test automation strategy [24].

(c) Top Management Support

In addition to many organizational and human factors, top management support for test automation is found to be the most important factor as lack of managerial support will enhance the possibility of change-related problems [24]. Test automation typically requires a high initial investment in terms of cost and time, i.e. acquiring tools and training people and so on before the benefits of test automation become apparent. Thus, tight schedules and/or budget pressure (time and resource constraints) might prevent the implementation of test automation.

Cross-cutting and other factors

Several appropriate factors in the context of more than one group lie under this category, named as cross-cutting factors. It also includes factors that could not be classified into any of the abovementioned categories [22]. These factors are further categorized into three groups named: economic, automatability of testing, and development process.

(a) Economic

Factors under this category relate mainly to the benefits of test automation in terms of effort and cost tradeoffs among manual and automated testing [22]. The identified factor mainly considers the aspect of test results bringing little value to a business, emphasizing the effort spent and benefits gained, such as Return-Of-Investment (ROI) calculations. Therefore, there is a need to evaluate the feasibility of automated testing [25] as automation for small products is considered much expensive [24]. If multiple iterations of testing are required, conventional wisdom holds that test automation is usually more cost-effective than manual testing. Automated testing has a higher upfront cost than manual testing, but the automated test execution cost is lower, specifically if the testing is done multiple times [22]. Another factor under this category is the availability of resources. According to the study [24], the ratio of the resources required for automation to the total available resources can inform automation decisions. For example, if automation is to be adopted successfully, "intensive training" may be required, which needs a budget, time, and priority in the work plan. When competencies are inadequate, and there are no resources available for the training required to obtain the skills, it may be preferable to avoid automating. Therefore, organizations must identify whether necessary resources are available for automating the tests [25]. Inefficient time exploitation also impacts automated testing due to the unnecessary waste of time, hence adding less value to the test [32]. As a result, testing must be tracked in terms of cost (effort) and benefits and calculate the ROI of automation to use cross-cutting factors successfully in decision-making.

(b) Automatability of Testing

Another element in this group is the automatability of testing, which relates to how simple it is to perform automated testing on the SUT. Automatability is influenced by various factors, including the test automation tool(s) used and the skill level of testers. The identified component considers that test automation will necessitate a significant amount of maintenance effort [14,22]. Focusing on testability from the earliest phases of the development enhances the effectiveness of the automation process and supports the practitioners to keep test effort under control [25]. It is also vital for product requirements to be clear to ensure that the system is testable. Due to the increase in demand for new functionality and growing maintenance issues, improving the testability greatly reduces the cost of testing. Low testability in the system under test, especially when it comes to automated testing, can generate issues and potentially costly workarounds to adopt automated testing [24]. As a result, automation success will be jeopardized if the SUT or test cases are difficult to automate.

(c) Development Process

The choice of the development process, such as agile methods, is considered another element while deciding on automated testing. An important aspect of the software development process (i.e. the number of releases of the product) is also considered under this category of factors as higher release frequency increases test automation benefits, according to the study [22]. It is noticed that some of the factors, for example, test reuse and repeatability have an implicit relationship to the subject that the development process influences the decision of automated testing as they are commonly used in Agile development processes where frequent testing iterations are conducted. Inadequate development practices also impact automated testing, which is also considered under this category [24].

4.1.6. Existing methods, tools, or frameworks

We have reviewed different methods, tools, or frameworks presented to highlight the effect of decision-making for test automation in agile-based software development. The review of the existing decision-making methods, tools, or frameworks for test automation in ASD is presented in Table 5. The review matrix includes all of the necessary details for each model and highlights the limitations. The labels include the reference number (Ref.), method (model/technique), evaluation measures, findings, research description, and limitations.

A checklist provided by Garousi and Mäntylä [22] was validated and modified by Lindholm [26] to establish a method to facilitate the test cases selection for automation. To achieve the objective mentioned above, Lindholm [26] conducted a literature review and interviewed practitioners. After modifying the checklist, the factors considering only test case selection for automation were used to evaluate the decision tree. The results of the study show that the method provides economic and organizational benefits in the industry. The study lacks the identification of important factors, particularly in the context of agile-based software development. Furthermore, the opinion of experts of a specific company was used to evaluate the decision tree, due to which their results cannot be generalized.

In contrast, Garousi and Pfahl [33] conducted a case study to demonstrate the applicability of the SD model and compared the performance of manual and automated testing. The model was proposed to aid decision-makers in determining the appropriate level of test automation in a specific project. They executed multiple simulations demonstrating various automated testing levels to determine the effect of automated test execution. The simulations were done with varying personnel allocations and additional test cases in selected test cycles. Other productivity factors such as experience level and time were not considered during the simulations. Moreover, the productivity factors were changed for few activities only, i.e. test case scripting and execution. The study results show that the test performance target can be achieved with fully automated test execution with less effort.

Baral et al. [34] evaluated the Automated Test Decision Framework (ATDF). They determined whether it could rank software components in terms of ROI (Return on Investment) while implementing automated tests. When implementing automated testing to a software project, ATDF is utilized to prioritize the restricted resources to determine where these resources can be applied. They assessed several legacy software projects by adapting ATDF to seven open-source projects. The projects were chosen based on some decided criteria. However, the study lacks in the identification and ranking of factors affecting decision-making for test automation.

Kazmi et al. [35] proposed a Production Possibility Curve Cost Model to determine test automation feasibility. They also explored how to combine automated and manual testing in the best way feasible. Within the cost restrictions, a production cost frontier-based technique is employed to determine the point of automation and manual testing. The paper discusses the Production Possibilities curve Frontier (PPF) in the context of deciding to automate a test case. The proposed approach successfully automates a test case that was calculated by replacing manual test cases and is accomplished by keeping the budget constraints in mind. The PPF curve depicts the best proportions of automation and manual test cases that can be achieved. Only single project decision-making is included in the

Table 5
Review matrix.

Ref.	Method	Measures	Findings	Limitations
[26]	Decision tree	• Survey strategy was used to validate the decision tree • Evaluated the decision tree by usage on regression test cases	• Provide economic and organizational benefits e.g., less human effort, reduction in cost, and shorter release cycles	• Targeted only experts of one specific company for the validation of factors and hence evaluation of the decision tree. • Focused only on factors related to the test cases selection for automation. • Lacks in focusing on the targeted research domain (ASD).
[33]	Process simulation model	• Conducted a case study	• The System Dynamics (SD) model may be used to calculate the actual effort generated by the two test process configurations with different manpower allocations. • It was observed that if test execution is entirely automated, the test performance objective can be achieved with less effort.	• Lacks in considering other productivity factors but only considered personnel allocation. • Changed the productivity factors for few activities only i.e. test case scripting and execution. • Evaluated the model in one specific company.
[34]	Automated test decision framework (ATDF)	• The Pearson correlation between the ATDF ranks and independent ROI measure has been calculated.	• For those components with a test code increase of more than ten percent, there was a moderate to significant link between the ATDF and the quality approach ROI metric. • Supports ATDF's ROI estimations for the projects under consideration.	• Lacks in focusing on the targeted research domain (ASD). • Used just for legacy components. • Rank components (which require test automation or not) not factors for decision making.
[35]	A production possibility curve cost model	• Replaced the manual test cases with automated ones within the cost constraints.	• The PPF curve depicts the best proportions of automation and manual test cases that can be achieved.	• The cost of testing is taken into account, but indirect costs are not. • Only single project decision-making is considered in this model. • Considered a single objective testing time. • Lacks in focusing on the targeted research domain (agile-based development).
[6]	Agent-based assisting system	• The correlation of the proposed apparatus with comparable works is examined	• Enhances Accuracy • Adaptable to coordinate the future testing details • Use of test code, i.e. dependent on the availability of portable code • Expand the execution and diminish the multifaceted nature of test executer agents.	• Test reusability may reduce efficiency. • Lacks in focusing on the targeted research domain (agile-based development).
[36]	A lean business case	• Several calculations were made by using example variables and generalized values	• Test automation is viewed as a valuable tool that makes the testing process simpler and more efficient.	• The research was focused on a specific product and company.

suggested approach. Moreover, the study also lacks indirect costs; for example, requirement stability, QA team skills, and available testing time are not focused on agile development.

An agent-based assisting system for determining a tool for software test automation according to the requirement is proposed in the study [6]. The authors empirically validated the proposed system by assessing the proposed apparatus's correlation with similar works. A model of the proposed device has been created and is executed to run a few tests in Java. The results are encouraging and support the structure's overall design. The proposed methodology provides several financial benefits, enhances accuracy, and diminishes the multifaceted nature of test executer agents. The proposed work has some deficiencies that are not evaluated in agile development, and test reusability may reduce efficiency. According to the study [36], test automation is viewed as a valuable tool that simplifies the testing process and makes it more efficient by handling more repetitious test cases and reducing the time and resources required for testing.

In another study [36], a Lean Business Case was built, backed up by calculations about test automation implementation and ROI based on an epic analysis using Scaled Agile Framework (SAF) current templates. Test automation's use, necessity, and benefits

Table 6
Demographic profile of respondents.

Demographics	Respondents	Frequency	Percentage
Position	Project Manager	43	31.62
	Tester	17	12.5
	Test Automation Engineer	35	25.74
	Quality Assurance Engineer	41	30.15
Total	–	136	100%
Working experience	Less than 1 year	17	12.5
	1–5 years	85	62.5
	5–10 years	29	21.32
	More than 10 years	5	3.68
Total	–	136	100%
Organization size	Small (1–50 employees)	39	28.68
	Medium (51–250 employees)	47	34.56
	Large (>250 employees)	50	36.76
Total	–	136	100%

were investigated through interviews. Several calculations were made using example variables and generalized values to propose a framework on the expenses of adopting the MVP (Minimum Viable Product) and the expected decrease in testing costs. The most prominent risks identified were automating more complicated scenarios, immature test environments and systems, and changes in system GUIs. Furthermore, the research focused on a single company and product.

In summary, the existing work lacks in providing decision support for test automation in agile-based software development. The current state-of-the-art only identified factors affecting automated software testing. The factors supporting specifically the selection of test cases for automation are focused on in the existing studies. However, the limitations in the existing literature necessitate the need to propose a conceptual model supporting decision-making for test automation in ASD, considering the factors affecting automated software testing.

4.2. Findings of survey

The results of an empirical investigation of factors are presented in this section, including a demographic profile of respondents and a quantitative analysis applied to their responses. A questionnaire-based survey was used for the empirical investigation of factors in the current study.

4.2.1. Demographic profile of respondents

During an empirical investigation, detailed information on a set of descriptive statistics is highly important. To infer more important results, there is a need to look closely at the practitioner's fundamental information and some pertinent organization-related information. Therefore, the current study acquired demographic and organization-related information from the respondents. It was done to acquire more significant and reliable survey responses. IT Professionals of 11 countries were targeted in this study, as given in Appendix C, based on which 136 responses were obtained.

A total number of 136 responses were used in this study based on similar primary studies [28,29], in which a sample size of <100 was used due to a specific targeted population. Based on which their results cannot be generalized. Therefore, due to the general population (i.e. agile practitioners) being targeted in this study, a sample size of 136 responses was used. Table 6 lists the respondents' demographic information, including job position, working experience, and organization size based on the number of employees. The values of these parameters were based on a similar study [20].

Project managers (31.62%), quality assurance engineers (30.15%), test automation engineers (25.74%), and testers (12.5%) were the most common positions acquired among the 136 respondents (12.5%). The respondents' job experience was important because this study primarily focused on the ASD environment. 17 (out of the 136 respondents) had less than a year of work experience (12.5%). In contrast, 85 (62.5%) of respondents have 1–5 years of work experience. Following that, 29 (21.32%) and 5 (3.68%) respondents, respectively, had work experience of 5–10 years and more than 10 years.

Mostly, small and medium-size organizations contain up to 250 employees [30]. In this research, 39 respondents were working in small-size organizations, including 1–50 employees. Whereas 47 were in medium-size organizations, including 51–250 employees. In contrast, 50 of them were in large organizations, including more than 250 employees. As a result, the study's demographic findings suggest that the respondents were well-positioned and had prior experience working in ASD-related organizations.

4.2.2. Quantitative data analysis

We performed a Levene's test, a T-test, and a Spearman correlation test for quantitative analysis. These statistical tests were employed to ensure the validity of the results and to discern whether the identified factors are as important as the literature indicates. The positive replies (i.e. agree and strongly agree) were picked from the survey data. Based on the type of data collected in this study, the tests mentioned above are selected to find the significant variances across the variables. Different researchers have used the same analysis method for similar nature of data [37].

Table 7
Levene's test.

Summary						
Groups	Count	Sum	Average	Variance		
SLR %age	21	1430	68.09524	168.0655		
Survey %age	21	1401.06	66.71714	161.8245		
ANOVA						
Source of variation	SS	df	MS	F	P-value	F crit
Between groups	19.94104	1	19.94104	0.120895	0.729889	4.084746
Within groups	6597.8	40	164.945			
Total	6617.741	41				

Table 8
T-test for equal variance.

	SLR %age	Survey %age
Mean	68.0952381	66.71714286
Variance	168.065476	161.8245114
Observations	21	21
Pooled variance	164.944994	
Hypothesized mean difference	0	
df	40	
t Stat	0.3476997	
P(T ≤ t) one-tail	0.3649444	
t Critical one-tail	1.68385096	
P(T ≤ t) two-tail	0.72988879	
t Critical two-tail	2.02107537	

Levene's test

The importance of the homogeneity of variance between the results attained from SLR and the Empirical study is quantified using Levene's test. The primary concern is with the significance value. If greater than significant 0.05 (i.e. $p > 0.05$), group variances can be treated unequally. According to Levene's Test results for RQ-1 (i.e., $p = 0.729889 > 0.05$), null hypothesis is accepted for differences assumed. Table 7 shows the resulting variance values as well as the percentages derived from literature and industry.

T-test

In addition to Levene's test, the Independent Sample t-test is also applied to compare the mean differences between the data obtained for RQ-1 through SLR and Empirical study, as displayed in Table 8. Based on the assumption, the results of the Independent Sample t-test ($t = 1.68$ and $p = 0.36 > 0.05$) demonstrate that the findings of SLR and the Empirical study are not significantly different. A smaller t-score indicates that the groups are the same as the t value should lie between (−2, 2).

Spearman test

A correlation analysis test, namely the Spearman correlation test, was used in addition to the t-test. Table 9 shows a comparison of the ranks obtained from SLR and the Empirical investigation. We used Spearman's rank-order correlation test to assess the significance of the discrepancies between the SLR and Empirical study outcomes. A Spearman correlation test indicates a positive correlation by a Spearman's coefficient (rs) value closer to 1. In contrast, a negative correlation is shown by a value of rs closer to −1. The Spearman coefficient for our study was 0.521514159, representing a strong positive connection between the ranks derived using SLR and the outcomes of empirical investigation.

4.3. Comparison of the findings

A comparison based on the results obtained from SLR and Empirical study was conducted. Fig. 7 represents the comparison of the percentages of factors acquired from SLR and the Empirical Study. The results were evaluated to determine the critical factors affecting test automation in the ASD context. Those factors were assumed critical and had frequency >70% in both the literature and an industrial perspective. With a frequency >70% in literature and industry, a highlighted factor is equally important for practitioners and must be considered in decision-making. The extraction of essential factors is of prime importance because these factors might significantly impact other factors. Fig. 8 highlights the essential factors (literature and industry) along with their percentages. The findings of the study explored that seven factors are the most influential in both SLR and the empirical study. The factors are TA-5 (test tool), TA-7 (test process), TA-14 (skill level of testers), and TA-18 to TA-21 (top management support, economic factors, automatability of testing, and development process), respectively. Whereas the findings of the empirical study also determined that three influencing factors are considered critical by the practitioners, these are not considered critical according to the cited literature. Moreover, two factors are considered critical in the cited literature but are considered moderate by the practitioners. At the same time, the practitioners considered two influencing factors to have the least impact on test automation but are not considered as least significant in the cited literature. Therefore, comparing the results allows the Practitioner to focus more on the critical factors while implementing automated testing in the ASD context. The study resulted in a list of influencing factors for test automation in agile software development.

Table 9
Comparative ranking of identified factors.

Factors	SLR %age	%Rank SLR	Survey %age	%Rank survey
Maturity of SUT	67.5	10	61.03	15
SUT life-cycle	55	19	63.97	13
SUT customizability	52.5	21	41.18	20
SUT complexity	55	19	55.15	18.5
Test tool	100	1	87.5	1
Technical/Environmental	72.5	7.5	60.66	16
Test process	77.5	5	72.61	9
Need for regression testing	55	19	73.53	5.5
Test type	57.5	16.5	66.18	12
Test reuse/Repeatability	65	11	73.53	5.5
Test importance	60	14	80.15	4
Test oracle	60	14	55.15	18.5
Test stability	62.5	12	38.24	21
Skill level of testers	92.5	2	83.09	2
Behavioral	72.5	7.5	62.74	14
Staffing	57.5	16.5	67.65	11
Steering	60	14	58.46	17
Top management support	70	9	72.79	8
Economic factors	80	4	73.28	7
Automatability of testing	82.5	3	72.55	10
Development process	75	6	81.62	3

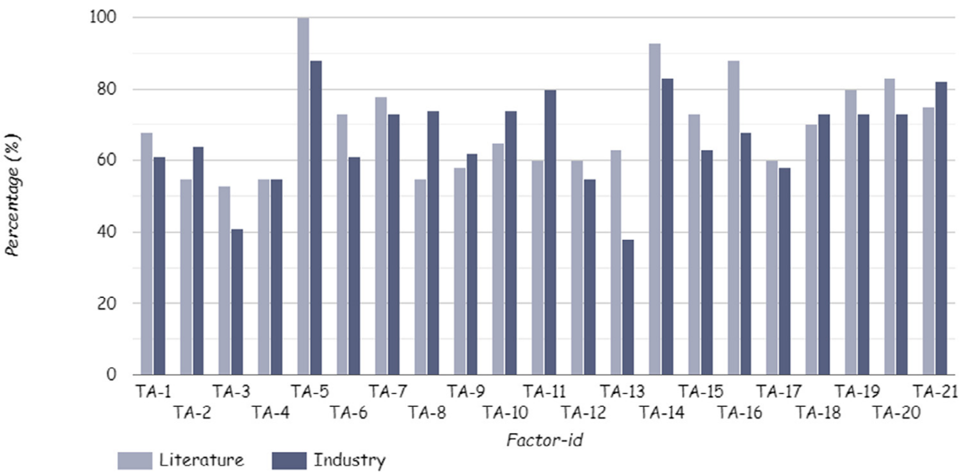

Fig. 7. Comparison of factors (SLR and empirical study).

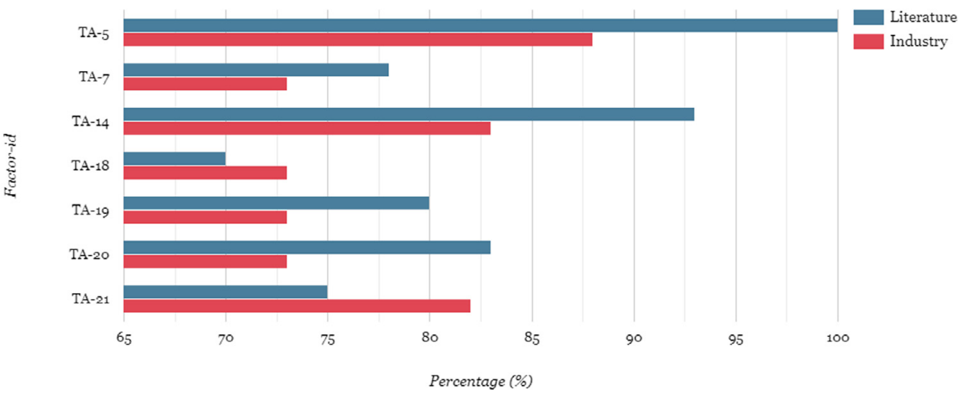

Fig. 8. Critical factors.

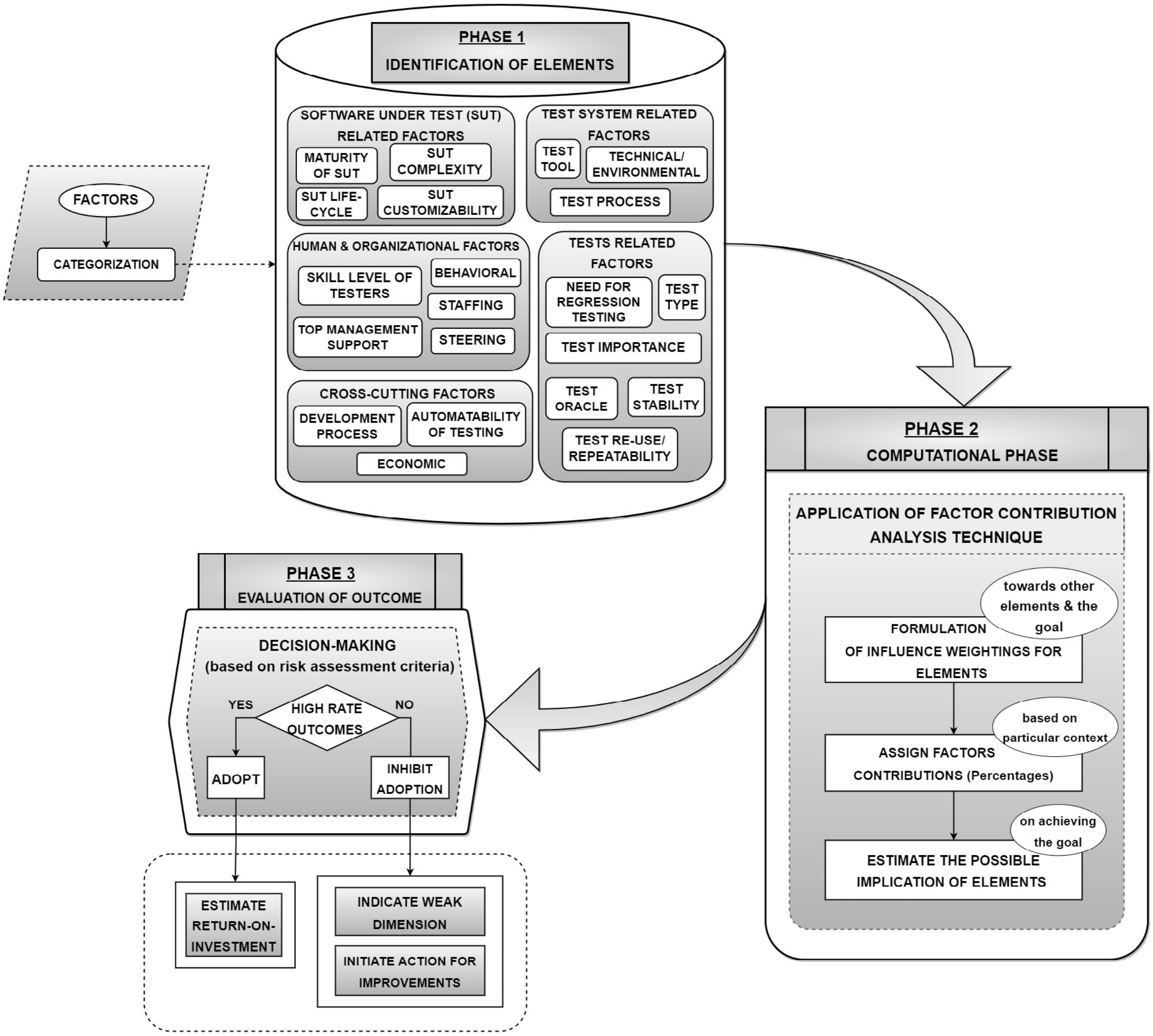

Fig. 9. Proposed conceptual model.

5. Proposed conceptual model

This section presents the proposed conceptual model. As already discussed, RQ-2 aims at providing decision-makers with a mechanism for test automation in agile-based development. Initially, the development procedure and phases of the proposed model are explained. After that, the model's applicability is demonstrated using two hypothetical scenarios. It allows decision-makers to target critical areas and quantify the risk of not implementing automated testing before and during testing. The conceptual model established in this research is based on evaluating the factors affecting automated software testing. Fig. 9 illustrates the model.

5.1. Development procedure

It is intended to integrate all research findings to develop a conceptual model that will help decision-making for test automation in ASD based on the risk associated with successful test automation. The motivation behind the selection and categorization of the factors in this model was, that the model is based on the concepts and their relationships. In addition, each phase required some inputs for its correct functioning. Three fundamental phases contribute to the development of the model.

- **Phase 1: Identification of Elements**
  This phase identifies and utilizes factors impacting test automation to aid decision-makers in implementing test automation in an ASD environment. The identified factors are further classified into five categories that are regarded as model elements.

- **Phase 2: Computational Phase**
  The output of the first phase is utilized in phase 2 to implement FCA. The FCA approach is used to determine the possible impact of elements on test automation success.
- **Phase 3: Evaluation of Outcome**
  Finally, the last phase analyzes the findings and determines the possibility of success based on the formulated criteria. This phase includes a decision point for test automation adoption.

As a result, the model considers all relevant factors to aid in decision-making for test automation in an ASD environment.

5.2. Phases of the model

The following sections provide a detailed explanation of the phases of the proposed model and their components:

5.2.1. Identification of elements and associated factors

This phase includes a list of elements and their associated factors that must be addressed before the start of test automation. These elements and their related factors were identified through SLR and validated by practitioners in the previous sections. Following empirical research, the 21 identified factors are further classified according to their properties. The elements indicate the primary categories, while the factors describe essential details about each element that the decision-maker must consider while planning and analyzing the risk associated with successful test automation adoption. The following sections discuss the identified elements and their associated factors.

Software Under Test (SUT)

The properties of a Software Under Test (SUT) have a significant influence on automation decisions. The most critical aspect in this category is the maturity of the SUT, which relates to the product's stability since automated tools are suggested for stable applications. However, if significant modifications occur (e.g., the re-implementation of new functionalities) considerable work will be required for test co-maintenance or test repair [22]. As a result, the advantages of automated testing will be diminished, and one may be unable to continue automating while addressing the issues. Additionally, this category includes factors affecting automated testing decisions about the SUT's properties, such as its complexity, customizability, and life cycle. These considerations must be made to make the best decision possible about SUT-related factors.

Test System

A test system is developed by integrating test tools and a testbed. It has all the necessary elements that reflect the typical steps in a testing process. If the tools and test systems cause complications, the automation adds little or no value to the testers. This category includes tool selection and adoption factors before starting the test automation process [22]. If an adequate test automation tool is not available for a particular case or requirement, and building one in-house is impractical, test automation should be avoided. As a result, familiarity with the test tool and its compatibility with the SUT is required. The technical or environmental element addresses issues in this category, such as test tool capability constraints and test environment configuration management. They substantially influence the company's efficiency and the advantages of automated testing. Another component in this category is the test process, which considers various factors when making automation decisions. The scope of the test automation process should be well-defined and adhered to by all teams before initiating automated testing [25]. Appropriate criteria for test scripting should be established and attached throughout the test process. Test data availability and manual testing practice before automated testing, particularly in the case of GUI testing, are also regarded as critical elements in this category.

Tests

When test automation is employed, the cost of developing tests often increases, while the cost of re-running a test decrease [22]. Understanding the number of tests re-runs necessary and the effort involved in test design and maintenance is vital. Each test case and test suite have unique properties that affect the automation choices available. This category covers various aspects to consider when deciding, such as the increased requirement for frequent regression testing due to continuous and quick software updates. Specific tests, such as performance and load testing, are considered acceptable for automation, while others, such as user experience testing, are not. As a result, test type is a critical component to consider when automating tests. Test reuse and repeatability relate to instances when the same test may be reused as a component of another test due to its similarities. It must be performed in several contexts to increase test repeatability. Additionally, this category includes test importance, which tackles the issue of proper test case selection. The process for identifying defects, referred to as the test oracle, is equally critical for decision-making in this category. Automating test oracles that are not stable and predictable is not recommended since the automated comparison will be unstable, resulting in many false positives. Another critical factor to consider in this area is test stability since automating an inconsistent test is not wise.

Human and Organizational

Test automation is a complicated activity that demands distinct and frequently new abilities since the automation of manual tasks will result in a significant shift [24]. As a result, it is critical to examine the testers' skill set and competence before initiating automated testing. Additionally, the behavior of the individuals involved in a test automation project has a substantial impact on the project's result. Before implementing automated testing, it is necessary to evaluate resistance to change. There are various possible causes for resistance, including established manual testing practices [3,17], assigning responsibility between development and test, and a low learning curve [4]. High expectations may also diminish the perceived value of automation if the desired outcomes are not attained within the anticipated period [24]. Since many businesses overspend on software development in general, it is predictable that cost is seen as a barrier to automation. Additionally, it is vital to acknowledge that adopting test automation entails, at least in part, hazardous organizational change [22]. Consequently, the implementation of automation must be carefully planned to prevent the negativity that might fail the automation project. As a result, various aspects such as staffing, steering, and top management support are listed in this category. Staffing is related to a company's reliance on key personnel since shifting jobs involves training and eventually costs more [31]. Steering is a measure of organizational maturity, which entails the need for effective change management. In the absence of top management assistance, the probability of change-related challenges increases [24]. Test automation often takes a significant initial commitment in terms of money and effort, such as procuring tools and training personnel, before the advantages become evident. Thus, tight deadlines and budget restrictions (time and resource restrictions) may inhibit the use of test automation.

Cross-Cutting

Multiple factors that apply to several groups fall under this category, entitled cross-cutting factors. Additionally, it covers elements that do not fit well into any of the preceding categories. These considerations include economics, the ability of testing to be automated, and the development process. Economic concerns are primarily affected by the advantages of test automation in terms of cost and effort trade-offs between manual and automated testing [22]. This element primarily evaluates the issue of whether test results provide minimal value to the company, highlighting the time used and rewards achieved, such as ROI (Return-On-Investment) estimates. As a result, it is necessary to assess the viability of automated testing. Additionally, this category considers the resource availability for automating the tests. Another element in this category is test automatability, which refers to how easily the SUT can be tested in an automated manner. This component considers that test automation will need considerable maintenance work [14,22]. Concentrating on testability from the start increases the efficacy of the automation process and assists practitioners in maintaining test effort under control [25]. For example, agile methodologies are also believed to affect the decision to use automated testing. It is noted that several of the factors, such as test reuse and repeatability, have an implicit relationship to the subject that the development process influences the decision to use automated testing. They are typically applied in Agile development processes that involve frequent testing iterations.

5.2.2. Phase 2: Application of FCA technique

As discussed in the previous sections, the model is being presented to aid decision-makers in adopting test automation in the context of ASD. In this research context, decision-making includes both test planning and risk management to ensure the successful deployment of test automation. Therefore, the Factor Contribution Analysis (FCA) approach is utilized in this phase to determine the potential impact of factors on successful test automation, as the technique is associated with both risk management and test planning [38]. FCA is based on the elements and factors that contribute to achieving the targeted objective, such as successful test automation. The approach evaluates the risk effect of possible changes because one or more elements' total contribution falls short of 100%. The discussion includes changes to the weightings of the elements' impact on the goal and the risk implications of accomplishing the goal.

The FCA approach is applied in two steps.

1. Initially, the decision-makers estimate an element's influence on another element or on accomplishing the objective in percentage terms, representing the possibility of not achieving that goal.
2. Assign factor contributions depending on the specific scenario and organizational environment, which eventually contributes to the goal's achievement.

A relationship's influence weighting establishes the amount to which one factor might influence other elements or the objective. Some, but not all, of one element's impact may be addressed by the influence of another element. The user's knowledge and experience determine the weighting of influence. Moreover, factors are a distinct group of items that relate to a single element. Each factor contributes a percentage to the element with which it is related. FCA approach is applied to the scenario represented in Fig. 10 to explain its working.

It consists of a goal, denoted by C0, and five major identified elements, represented by C1, C2, C3, C4, and C5. Each of the five recognized elements has a collection of factors that describe the element's details. The effects of each element are mapped to the objective or other elements through directional connections. These connections serve as a foundation for risk assessment before and during test automation. Each relationship is accompanied by an illustrated weighting indicated in percentages. The percentages in Fig. 10 are for demonstration and discussion only and were computed using the findings of RQ-1. These percentages will vary depending on the type of software to be tested within the organizational context.

The following essential criteria and information aid in the application of the factor contribution analysis technique.

1. The influence weighting assigned to each element will be between 10% and 90%.
    - Each element is related to the objective or another element and is evaluated separately to ascertain its influence.
    - Each element will have a minimum of one connection, with the maximum number specified by the number of elements and their relationships.
2. Factor contribution analysis is concerned with the overall factor contribution, not with the element's factor contributions.

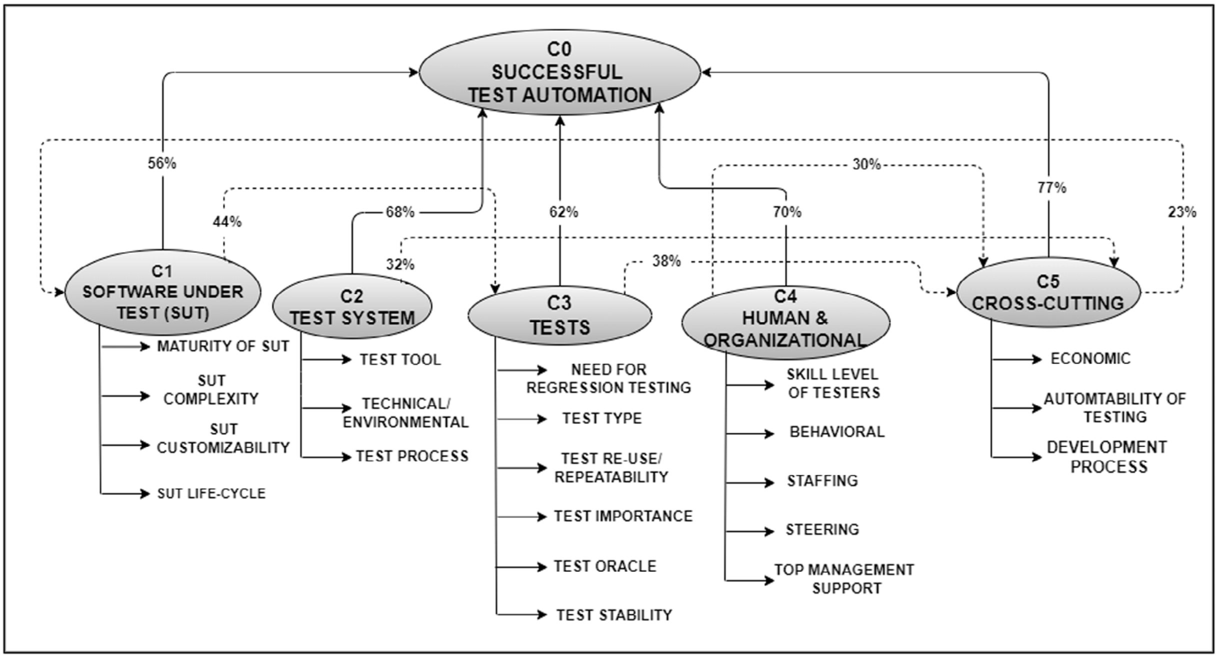

Fig. 10. Application of FCA technique.

- The total of the factors affecting each element must equal 100%. If the factor analysis does not provide a value of 100%, more or fewer factors for that element must be identified, analyzed, and allocated a contribution percentage equal to 100%.

3. If the loss of a factor contribution is equal to or more than 90%, it has been proven through a series of simulations that the objective will not be reached; hence, the maximum loss for a factor contribution cannot exceed 90%.
4. Risk management is objective-driven and does not apply to any particular element.
5. Backtracking between elements is not allowed since this would result in an infinite loop (as would be the case for C1 → C3 → C5 → C1 → C3).

After assigning influence weightings to connections, anything that impacts one or more influence weightings represents a risk in attaining the objective or success of an element and may be utilized for risk assessment. Thus, if the element's influence weighting percentage(s) is lower than its original percentage(s), it will affect the risk of not accomplishing the objective.

There is a maximum of 15 different possibilities based on the present structure, as shown in Fig. 10. The overall factor contribution and potential consequences of the factors are mentioned below.

(a) When the overall factor contribution to the C1 element is less than 100%, the following impacts occur:
C1 → C0
C1 → C3 → C0
C1 → C3 → C5 → C0
(b) When the overall factor contribution to the C2 element is less than 100%, the following impacts occur:
C2 → C0
C2 → C5 → C0
C2 → C5 → C1 → C0
(c) When the overall factor contribution to the C3 element is less than 100%, the following impacts occur:
C3 → C0
C3 → C5 → C0
C3 → C5 → C1 → C0
(d) When the overall factor contribution to the C4 element is less than 100%, the following impacts occur:
C4 → C0
C4 → C5 → C0
C4 → C5 → C1 → C0
(e) When the overall factor contribution to the C5 element is less than 100%, the following impacts occur:
C5 → C0
C5 → C1 → C0
C5 → C1 → C3 → C0

Table 10
Risk assessment criteria.

Risk category	Influence weighting criteria
Low	≥90%
Low–Medium	≥80% to <90%
Medium	≥70% to <80%
Medium–High	≥60% to <70%
High	<60%

5.2.3. Phase 3: Evaluation of outcome

Phase 3 analyzes the findings and estimates the possibility of success based on the specified criteria. This phase includes a decision point for test automation adoption. If the results fulfill the established requirements for high-quality outcomes, decision-makers are advised to use automated testing. Otherwise, adoption will be prohibited. In this scenario, the weak aspects are highlighted, and decision-makers are urged to take the necessary corrective actions. The assessment of risk categories for factor contribution analysis is shown in Table 10, along with how these categories are interpreted concerning changes in the influence weightings applied to the essential elements and the objective. Other researchers have employed the same risk assessment criteria when using the FCA approach in their context [38].

As previously stated, this phase includes a decision point on whether or not to use test automation. In this situation, the adoption criterion or success value will be assessed against a pre-defined threshold, which is more than or equal to 70% for adoption [39]. In the case of a poor outcome (less than 70%), the model will reveal weak aspects for future refinement, whereas it will direct toward adoption in the instance of a positive outcome.

5.3. Illustrative examples

The applicability and implementation of the model are explained through hypothetical examples.

Scenario 1

For example, in scenario 1, what would happen if the influence weightings for C2 and C4 were lowered by 10%?

The following effects are examined using the above-mentioned possible scenarios for the elements' total factor contribution effects. In other words, the changes to C2 and C4 will affect the objective. The effect of C2 is from C2 to C0, C5 to C0 via C2, and C1 to C0 via C5. The impact of the change to C4 adds to the effect of the change to C2. The effect of C4 is from C4 to C0, C5 to C0 via C4, and C1 to C0 via C5. The new influence weightings on the goal are:

- C1 → C0 – 44.8% (56 minus reduction: 10 + 10 = 20% of 56)
- C2 → C0 – 61.2% (68 minus reduction: 10% of 68)
- C3 → C0% – 62% (62 minus reduction: 0% of 62)
- C4 → C0% – 63% (70 minus reduction: 10% of 70)
- C5 → C0 – 61.6% (77 minus reduction: 10 + 10 = 20% of 77)

The overall influence on achieving the objective has been decreased. When the new influence weightings on the goal (292.6) are divided by the total of the previous influence weightings (333), the result is 87.87 percent. According to Table 10, the loss or change in factor contribution to C2 and C4 indicates that the influence weighting loss has a LOW–MEDIUM risk of not attaining the targeted objective. Therefore, according to the example mentioned above, the model will direct towards adoption of test automation based on the pre-defined standard, i.e. ≥ 70% for adoption [39], as there are more chances of success.

Scenario 2

For example, in Scenario 2, what would happen if the influence weightings for C2 were lowered by 15%, C4 and C5 by 25%?

The following influences are examined using the above-mentioned possible scenarios for the elements' total factor contribution effects. In other words, the changes to C2, C4, and C5 will affect the objective. The effect of C2 is from C2 to C0, C5 to C0 via C2, and C1 to C0 via C5. The effect of C4 is from C4 to C0, C5 to C0 via C4, and C1 to C0 via C5. The impact of change to C5 adds to the effect of the change to C2 and C4. The effect of C5 is from C5 to C0, C1 to C0 via C5, and C3 to C0 via C1. The new influence weightings on the goal are:

- C1 → C0 – 19.6% (56 minus reduction: 15 + 25 + 25 = 65% of 56)
- C2 → C0 – 57.8% (68 minus reduction: 15% of 68)
- C3 → C0 – 46.5% (62 minus reduction: 25% of 62)
- C4 → C0 – 52.5% (70 minus reduction: 25% of 70)
- C5 → C0 – 26.9% (77 minus reduction: 15 + 25 + 25 = 65% of 77)

The overall influence on reaching the objective has been decreased. When the new influence weightings on the goal (203.35) are divided by the total of the previous influence weightings (333), the result is 61.07 percent. According to Table 10, the loss or change in factor contribution to C2, C4, and C5 indicates that the influence weighting loss has a MEDIUM–HIGH risk of not attaining the targeted objective. As a result, the model did not recommend adoption since the possibility of success is less than 50%.

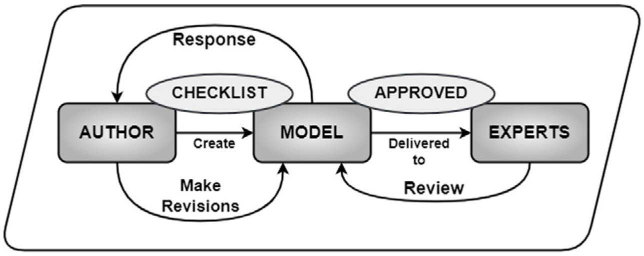

Fig. 11. Expert opinion methodology.

Table 11
Criteria-based questions for model validation.

Criteria	Questions	Description
Design	Is the overall design of the proposed conceptual model good enough, or does it need improvements?	This question is designed to obtain feedback on the design of the proposed conceptual model, considering visualization as an essential aspect of the understandability of the model.
Readability	Is the text of the proposed conceptual model readable or not?	This question was designed to get reviews on the readability of all the components involved in the model.
Relevancy	All phases of the proposed conceptual model contain the relevant components?	The purpose of this question was to assess that either the proposed model contains the relevant information or not.
Logical connection	Is the relationship among the phases logical?	This question addresses the logical connectivity among the phases of the proposed model. Logical connectivity here deals with the order between the phases.
Understandability	Is the proposed conceptual model easy to understand?	This question aims to get feedback in terms of the understandability of the proposed model.
Labeling	Are the phases of the model labeled correctly?	The question was designed to assess the labeling of the phases and components involved in the proposed model.
Acceptability	Is the proposed conceptual model acceptable for decision-makers?	This question assesses that whether the proposed conceptual model is acceptable for practitioners.

The work reported in [20] stated that large organizations with an in-house or outsourced test team that automates a significant proportion of test cases and adheres to agile or DevOps practices attain a considerable level of test automation maturity. Hence, it can be seen that the influence of each element in the model varies case-by-case. For instance, the skill level of testers is considered a critical aspect in test automation. Still, in specific contexts, the skill level of testers may be the primary impediment. In contrast, in cases of those that have already done a significant amount of automated testing, that element may be overlooked entirely. Therefore, the influence of elements should be evaluated based on the particular scenario.

6. Validation of proposed conceptual model

This section discusses the methodology used to accomplish the validation of the proposed model. Moreover, the results of the validation are also described here. To answer RQ-3, expert opinion methodology is used. However, before sharing the proposed conceptual model with the industry experts, initial validation is performed. In the initial validation process, the questionnaire and version 1 of the proposed model is shared with the academic experts Dr. Saif Ur Rehman Khan and Dr. Shahid Hussain employed as Assistant Professors at COMSATS University Islamabad (CUI), and Penn State University, USA, respectively. The items listed in the questionnaire were examined and determined whether or not respondents could grasp the questions. After the initial validation phase, the expert opinion methodology is followed. Experts are interviewed using a questionnaire including questions based on pre-defined criteria. The complete procedure is visually represented in Fig. 11 and the details are described in the sub sections.

6.1. Expert opinion

This study employs an expert validation procedure to validate the proposed conceptual model. A Google forms-based questionnaire[2] was used to interview the experts. The questionnaire also contained a conceptual model illustration. The author constructs the model and distributes it together with the checklist to industry experts. The list comprises the questions that will be used to validate the proposed model. For each question, a description and criteria are provided in Table 11.

Afterwards, the experts performed a review of the model and provide feedback to the author. The author revises the model based on the experts' responses, eventually approving the model. The industry experts were chosen to evaluate the model and determine

[2] https://forms.gle/fDY2a3YzDhdx2CDP8.

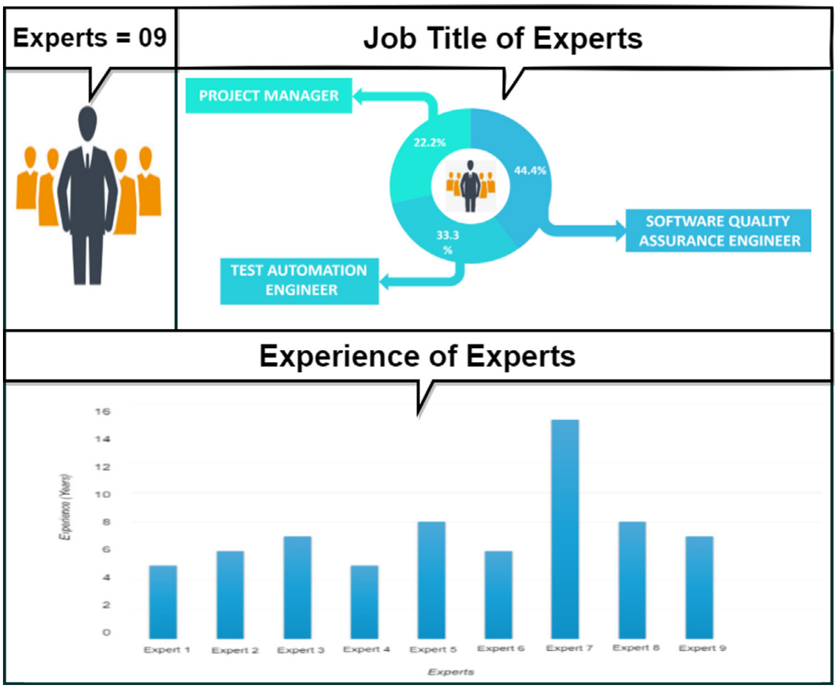

Fig. 12. Demographics of the experts.

if it should be accepted, rejected, or refined. Additionally, the experts were chosen based on pre-established criteria. The experts must meet the following requirements:

1. must presently work as a software quality assurance engineer, project manager, or test automation engineer.
2. the experience of potential expert must be of at least five years.

Notice that the prior studies [40,41] used the same expert selection and model validation criteria. The summary of the demographics of the experts is illustrated in Fig. 12.

As a consequence of the pre-defined criteria, nine experts were identified. Nine experts were considered enough for expert validation based on the existing studies [25,26], and [40] which used the same approach for validation with less than five experts. All the experts were having an experience of more than five years. Additionally, four of them were presently working as a software quality assurance engineer, three were test automation engineers and two of them were software project managers.

We obtained positive responses from the considered experts. Moreover, some experts also provided some suggestions for the improvement of the model. Fig. 13 depicts the general representation and summary of the expert's responses. The expert reviews were incorporated into the model's refinement, and the final version of the model is reported in this study.

7. Results and discussion

The current study aimed at identifying the factors influencing automated testing in the ASD context. In this study, we first performed a systematic literature review to find the factors affecting test automation in the ASD context. Then, the identified factors are validated through a questionnaire-based survey. And at last, a comparison based on the findings of SLR and empirical study is made. The ultimate purpose of the research is to develop a model supporting decision-making for test automation in ASD based on the influencing factors.

This section extensively discusses about the findings related to the devised research questions in this section. Regarding RQ-1, a Systematic Literature Review is performed, and it is concluded that several influencing factors should be considered while decision-making for test automation in the ASD context. The current work identified 21 influencing factors from 43 primary studies, out of which 40 focused on those factors, which was further validated in the empirical study. It is recognized that test tool, technical/environmental, test process, the skill level of testers, behavioral, top management support, economic, automatability of testing, and development process are the most critical factors based on their frequencies according to the cited literature.

To address RQ-1.1, the 21 identified factors were further categorized into five elements: SUT-related Factors, Test System-related Factors, Tests-related Factors, Human and Organizational Factors, and Cross-Cutting Factors based on their properties. The Software Under Test (SUT)-related factors include four factors, SUT customizability, life-cycle, complexity, and maturity of SUT. Among the 21 identified factors, test tool, technical/environmental, and test process were included in Test System-related factors. The category

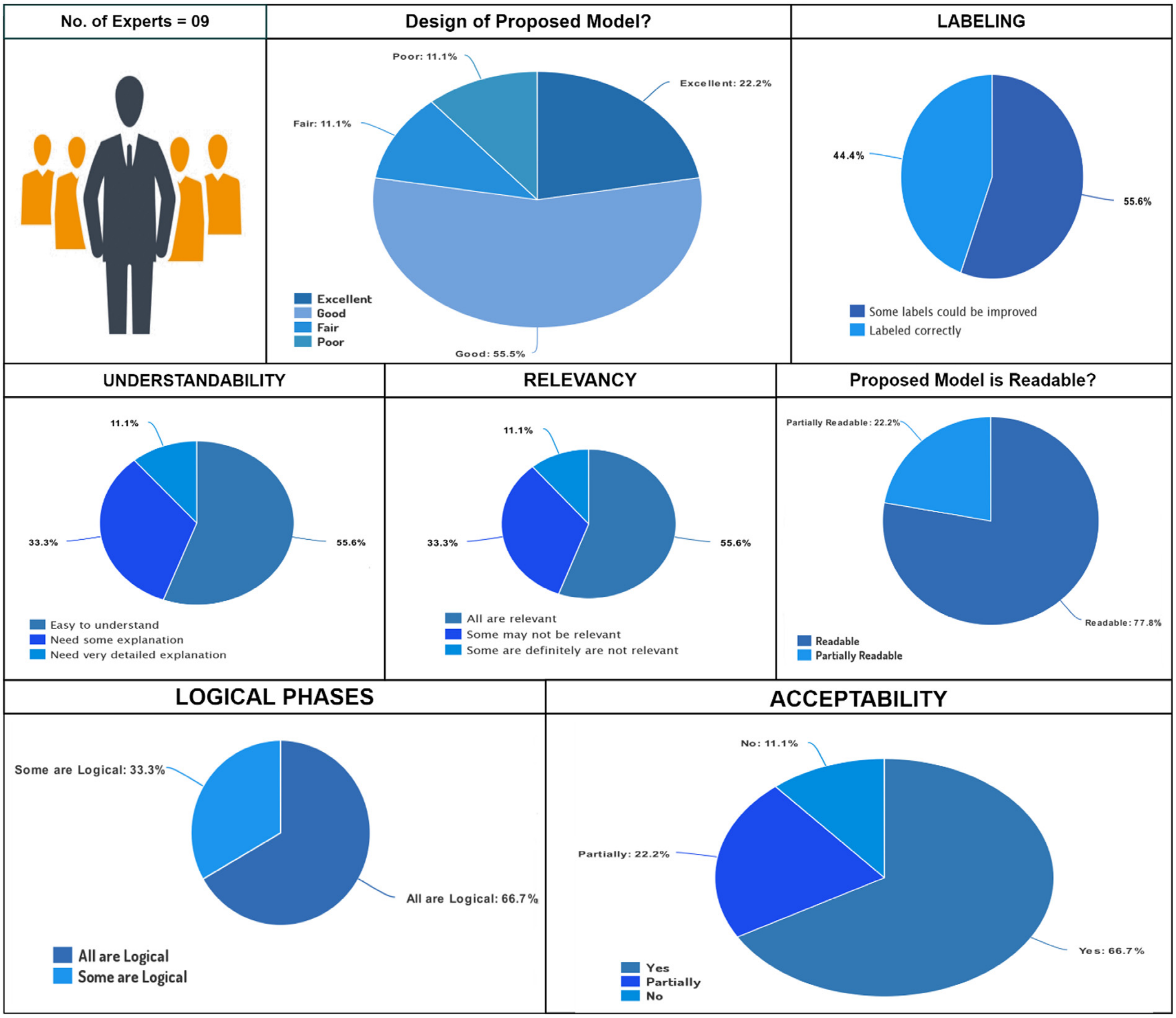

Fig. 13. Summary of the expert's responses.

named Tests-related factors involved the factors, such as the need for regression testing, test re-use/repeatability, test type, test importance, test stability, and test oracle. The category of human and organizational factors included factors related to employees, such as the skill level of testers and the behavior of the employees. It also involved factors related to business and planning, such as staffing, steering, and top management support. And the factors related to automatability of the testing, development process, and economic factors were placed under the last cross-cutting factors category.

Regarding RQ-1.2, the authors further validated the 21 factors identified from the cited literature to gain the perspective from project managers, software quality assurance engineers, or test automation engineers involved in agile-based software development. Afterward, 136 IT Professionals from various countries were selected through LinkedIn to obtain the industrial perspective and validate the identified factors. For the empirical investigation of factors, the data obtained were analyzed by applying statistical tests, such as Levene's test, independent sample t-test, and Spearman's correlation rank order test. After the quantitative analysis, the SLR and empirical study results are compared to identify the critical factors affecting test automation in agile development according to the cited literature and the empirical study. To analyze the importance of each factor, the criteria of frequency >70% as critical factors are adopted in this study. Moreover, a comparison was made based on which seven critical factors were identified from literature and industry. Test tool, test process, the skill level of testers, top management support, economic factors, automatability of testing, and development process are the seven factors considered to have a critical impact on test automation in ASD context based on the comparison.

To answer RQ-1.3, the existing decision-making methods, tools, or frameworks proposed for test automation in ASD are identified through the SLR. After reviewing those methods, it is concluded that the existing decision-making methods, tools, or frameworks lack in considering all the influencing factors. Their evaluations revolved around specific organizations and products, which cannot be generalized. Moreover, they lack in considering the targeted research domain, i.e. agile-based development. Ignoring the influencing factors can result in inaccurate decision-making for test automation in the ASD context. As a result, the elements must be considered to limit the negative impact of automated testing. However, based on the preceding discussion, it is determined that there is a need for researchers and practitioners to improve test automation decision-making in the context of ASD.

These findings show that a model based on the identified factors affecting test automation is needed. As a result, we developed a conceptual model to give decision-makers a mechanism to adopt test automation in agile development which addresses RQ-2. The proposed model estimate the possibility of success for test automation suitable for a particular project by including the recognized influencing elements. The FCA approach is applied in the model to determine the possible impact of elements on test automation success. As a result, the model considers all relevant factors to aid in decision-making for test automation in an ASD environment.

Regarding RQ-3, the authors adopted an expert opinion approach. The industry experts are chosen to validate the proposed conceptual model. Nine experts were identified based on the pre-defined criteria of experts having experience of at least five years and must be working as a software quality assurance engineer, project manager, or test automation engineer. This question evaluated the practical applicability of the proposed model. Positive responses are obtained from the experts. Moreover, some experts also provided some suggestions for the improvement of the model. The expert reviews were incorporated into the model's refinement, and the final version of the model is reported in this study.

8. Threats to validity

To ensure the accuracy and appropriateness of the findings from the conducted research, the primary types of validity threats were also considered. The four major types of validity threats are discussed below:

8.1. Internal validity

A validity threat to this study could be that the selected primary studies did not provide the causes for a certain factor's existence, which ultimately threatening internal validity. There was difficulty in mitigating this threat because the sources of the determinants were not formally specified in the primary studies. The application scenarios and their results presented in Section 4.3 are meant to illustrate how the proposed model can be applied in a software company to assist decision-makers in determining the possibility of success for test automation adoption in the ASD context. This research work demonstrates only the applicability of the proposed conceptual model. The percentages for influence weightings used in the scenarios are derived from the review and are used just for illustration purposes. Thus, the specific results cannot be generalized.

8.2. External validity

Another potential threat associated with the analysis of this study is that the primary study authors are from academia, which means they may lack in-depth expertise and comprehension of contemporary ASD developments. However, this threat was mitigated by confirming the criteria with practitioners. Both academic and industrial views were covered due to this, which strengthened this research. As for external validity, the only method used for the validation of the model is expert opinion. Thus, the obtained results cannot be transferred to the software companies quickly. Several case studies need to be conducted based on different parameters to assess the performance of the proposed model in real-world scenarios.

8.3. Construct validity

A systematic literature review is conducted to acquire an unbiased view of the targeted domain. Therefore, to tackle this threat, the study selection strategy was designed on fairgrounds. Specifically, the search string was refined on an iterative basis to ensure that it may not exclude relevant studies. Moreover, the experts were involved in evaluating and ensuring the accuracy and reliability of the questionnaire. The experts observed the items mentioned in the questionnaire to acknowledge the clarity and completeness of the questions. Based on the responses of the experts, the questionnaire was reviewed, and improvements were made. However, the questionnaire used for expert validation lacks in specifying the details for the responses of the experts. Therefore, this threat is mitigated in the extension of this work. The questionnaire has been revised and an open-ended question is included for enhancing the usefulness of the approach.

8.4. Conclusion validity

The devised objectives of this study have been achieved by conducting a systematic review and questionnaire-based survey. Consequently, it supports other researchers to replicate and advance the literature review. Surely, it is possible to miss some relevant studies, but this has a minor effect on the general categorization of influencing factors.

9. Research implications

This section provides key practical implications of the study's findings, which may benefit researchers and practicing professionals. By conducting a comprehensive analysis of the existing state-of-the-art and state-of-the-practice in ASD context and developing a conceptual model to aid in test automation decision-making, it is believed that this will assist in bridging the gaps between researchers and practitioners. It brings the researchers with too much room and opportunities to broaden the study.

Table A.1
List of selected articles through the performed systematic review.

PS-#	Ref-#	Study type	Database	Year	Publication venue
PS-1	[2]	Mixed method	Google Scholar	2020	Journal
PS-2	[4]	SLR	Google Scholar	2018	Thesis
PS-3	[5]	Mixed method	ACM	2016	Journal
PS-4	[6]	Proposal of solution	Google Scholar	2018	Journal
PS-5	[7]	Theoretical study	Google Scholar	2020	Journal
PS-6	[18]	Theoretical study	Springer Link	2020	Journal
PS-7	[20]	Empirical study	Google Scholar	2020	Conference
PS-8	[22]	MLR	Science Direct	2016	Journal
PS-9	[24]	SLR	Wiley Online Library	2017	Journal
PS-10	[25]	Mixed method	Google Scholar	2017	Thesis
PS-11	[26]	Mixed method	Google Scholar	2019	Thesis
PS-12	[33]	Empirical study	Wiley Online Library	2016	Journal
PS-13	[34]	Empirical study	IEEE	2019	Conference
PS-14	[42]	Empirical study	Springer Link	2018	Conference
PS-15	[43]	Empirical study	IEEE	2020	Conference
PS-16	[44]	Theoretical study	Google Scholar	2020	Conference
PS-17	[45]	Theoretical study	ACM	2017	Conference
PS-18	[46]	Empirical study	Springer Link	2020	Conference
PS-19	[47]	Theoretical study	Springer Link	2019	Journal
PS-20	[48]	Theoretical study	Springer Link	2018	Conference
PS-21	[49]	Empirical study	ACM	2020	Conference
PS-22	[50]	Empirical study	Google Scholar	2019	Thesis
PS-23	[51]	Mixed method	Google Scholar	2020	Thesis
PS-24	[52]	Mixed method	Google Scholar	2018	Thesis
PS-25	[32]	Empirical study	Google Scholar	2015	Thesis
PS-26	[53]	Proposal of solution	Google Scholar	2016	Journal
PS-27	[28]	Empirical study	Science Direct	2016	Journal
PS-28	[8]	Empirical study	Google Scholar	2018	Journal
PS-29	[29]	Empirical study	ACM	2018	Conference
PS-30	[54]	Empirical study	ACM	2016	Conference
PS-31	[55]	Mixed method	Springer Link	2016	Conference
PS-32	[31]	Empirical study	Google Scholar	2015	Thesis
PS-33	[56]	Theoretical study	Google Scholar	2019	Journal
PS-34	[35]	Proposal of solution	Google Scholar	2016	Journal
PS-35	[57]	Proposal of solution	Google Scholar	2015	Journal
PS-36	[58]	Theoretical study	Science Direct	2015	Conference
PS-37	[59]	Theoretical study	Google Scholar	2015	Journal
PS-38	[60]	Proposal of solution	Google Scholar	2020	Thesis
PS-39	[61]	Proposal of solution	IEEE	2018	Conference
PS-40	[62]	Empirical study	Springer Link	2016	Conference
PS-41	[63]	Mixed method	Google Scholar	2020	Thesis
PS-42	[36]	Empirical study	Google Scholar	2021	Thesis
PS-43	[64]	Empirical study	Springer Link	2021	Journal

9.1. State-of-the-art

The conceptual model proposed in this study is based on the identified factors influencing test automation in the ASD context. As the last phase of the proposed model involves a decision point, it suggests adopting test automation or inhibiting adoption based on the outcomes of phase 2. In the scenario of inhibiting adoption, weak dimensions are highlighted as the factors which cause the risk of not achieving the goal. The weak dimensions could support the researchers in determining strategies to overcome the negative impact of those highlighted factors for successful test automation. As a result, it would aid novice researchers in better understanding the targeted study context and advancing the current state-of-the-art in this domain. Moreover, the model can be extended to the domain of Agile Global Software Development (AGSD).

While the influences on the model's factors may vary, depending on the type of software being tested and within the organizational context, this will provide novice researchers with another avenue for future research. They may conduct several experiments and empirically investigate the proposed model's performance. Empirical studies may be executed against various parameters, such as time, budget, and quality. This aspect enables them to define and do ROI calculations more quickly. Additionally, different scenarios may execute investigations, including organizations of varying sizes, maturity levels, and project types. These may be used to determine if test automation is a suitable candidate for a specific company or a project.

9.2. State-of-the-practice

The advantage of the proposed model is that it gathers all critical elements and factors into a single model that represents a holistic perspective of the problem at hand. Another research implication is that a conceptual model for test automation

Table B.1
Quality assessment questions score (See Table a and b).

(a) QA score

Ref-#	QA 1	2	3	4	5	6	7	Total
[61]	1	0	1	1	1	0	0	0.5
[43]	1	0	1	1	0	1	1	0.5
[34]	1	1	0	1	1	1	1	1
[18]	0	1	1	1	1	0	0	0.5
[42]	1	0	1	1	0	1	0	0.5
[46]	1	0	1	0	1	1	0	0.5
[47]	1	0	1	1	0	1	1	0.5
[48]	1	0	1	1	0	1	0	0.5
[55]	1	0	1	1	1	0	1	0.5
[62]	0	1	1	1	0	1	0	0.5
[64]	1	1	1	1	0	1	0	0.5
[24]	1	1	1	0	0	1	1	0.5
[33]	1	1	0	1	1	1	1	0.5
[22]	1	1	1	1	1	1	1	0.5
[28]	0	0	1	0	1	1	1	0.5
[58]	1	0	1	1	1	1	0	0.5
[2]	1	1	1	1	1	1	0	0.5
[4]	1	1	1	1	0	0	1	0.5
[6]	1	1	1	1	1	1	1	1
[7]	0	0	1	0	1	1	0	0.5
[8]	0	0	1	0	1	0	1	0.5
[20]	0	0	1	1	1	1	0	0.5

(b) QA score (continued)

Ref-#	QA 1	2	3	4	5	6	7	Total
[25]	0	0	1	1	1	1	0	0.5
[26]	1	1	1	1	1	1	1	1
[44]	0	0	1	1	0	0	0	0.5
[50]	0	0	1	1	1	1	0	0.5
[51]	1	0	1	0	0	1	0	0.5
[52]	1	0	1	1	0	1	0	0.5
[32]	1	1	1	1	0	1	1	0.5
[53]	0	1	1	1	1	0	1	0.5
[31]	1	0	1	1	1	1	1	0.5
[56]	0	1	1	1	1	0	0	0.5
[35]	1	1	1	1	1	1	1	1
[57]	1	1	1	1	1	1	1	1
[59]	1	0	1	0	1	1	1	0.5
[60]	0	0	1	1	0	1	0	0.5
[63]	1	1	1	0	1	0	0	0.5
[36]	1	1	1	1	1	1	1	1
[5]	0	0	1	1	1	0	1	0.5
[45]	0	1	1	1	1	0	1	0.5
[49]	0	1	1	0	1	1	1	0.5
[29]	1	1	1	1	0	1	1	0.5
[54]	1	1	1	0	1	1	1	0.5

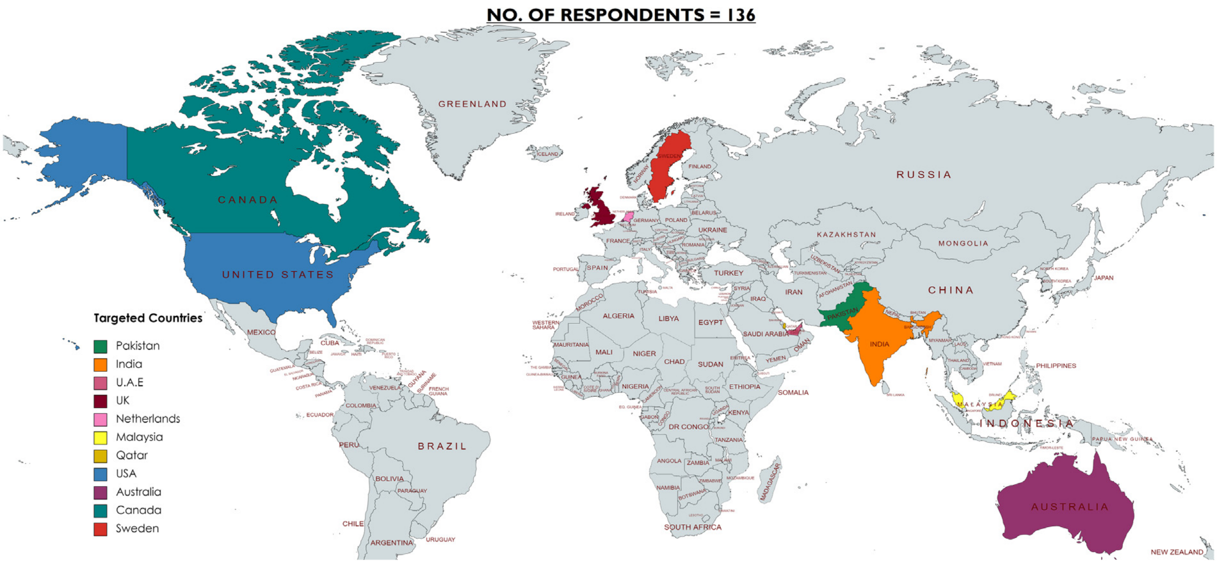

Fig. C.1. Demographic profile of respondents.

decision-making in ASD is proposed based on the identified factors, which will aid practitioners in the targeted area in terms of implementation. As previously stated, ignorance of these factors may result in significant erroneous resource and effort expenditures and eventually project failure. As a result, the influencing factors identified in this research assist practitioners in considering the influence of all of these affecting factors while implementing test automation in the context of ASD.

10. Conclusion and future work

In this study, the factors affecting test automation in the Agile-based Software Development (ASD) context are addressed. The decision about test automation adoption heavily depends on several factors and their interactions. Therefore, there is a need to determine those influencing factors for test automation in the ASD context. To achieve the devised research objective, 21 factors were identified affecting test automation in ASD. Subsequently, the factors were organized into five elements. Moreover, existing decision-making methods, tools, or frameworks for test automation were also considered, and their limitations were identified in the study. It is found that the existing methods lack in focusing decision-making for the adoption of test automation and considering all the factors affecting test automation. Therefore, we proposed a conceptual model focusing on decision-making for test automation in ASD based on the identified elements. The model's main phases include identifying the elements, applying the FCA technique, and evaluating the outcomes.

As already discussed, the underlying intention of the proposed conceptual model is to aid decision-makers in adopting test automation in the context of ASD. In this research context, decision-making includes both test planning and risk management to ensure the successful deployment of test automation. Therefore, the Factor Contribution Analysis (FCA) approach is utilized in this model to determine the potential impact of factors on successful test automation, as the technique is associated with both risk management and test planning. FCA is based on the influences of the elements and the associated factors that contribute to estimating the possibility of success to achieve the goal, in this instance, successful test automation. The current research also focuses on the validation of the proposed model. For the validation of the proposed model, a technique named expert opinion is used in this study. The results of the survey authenticated the applicability of the proposed model. The expert reviews were incorporated into the model's refinement, and the final version of the model is reported in this study.

The future work intends to focus on evaluating the performance of the proposed model by conducting several real-world case studies and estimating ROI calculations. It is noticed that the influence weightings of elements are calculated only by the experts based on their experience and skills. This approach can involve biases in the results. Depending on the identified limitations of the conducted study, we found that there is a need to provide several guidelines or a systematic approach to estimate the influence weightings for each element. Moreover, the proposed model involves only one decision-maker for assessing the contributions of the factors. Hence, incorporating several decision-makers in the proposed model can be regarded as another interesting research dimensions for the future research perspective.

CRediT authorship contribution statement

Shimza Butt: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Saif Ur Rehman Khan:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Resources, Writing – review & editing, Visualization, Supervision. **Shahid Hussain:** Formal analysis, Investigation, Writing – review & editing, Supervision. **Wen-Li Wang:** Writing – review & editing, Visualization, Supervision, Project administration.

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.

Acknowledgments

The authors appreciate the members of the Software Reliability Engineering Group (SREG) at COMSATS University Islamabad (CUI). The latter provided them with their comments and continued assistance during this research. Furthermore, they would like to express their gratitude to the IT Professionals (i.e. agile practitioners) who took part in the conducted survey.

Appendix A

This section represents the list and details of the articles selected through the performed SLR. The labels of Table A.1 includes primary study id, reference number, study type, database, publication year and venue.

Appendix B

This section represents the quality assessment score of the articles selected through the SLR. The labels of Table B.1 includes reference number, each individual question QA score, and total QA score.

Appendix C

This section represents the demographic profile of respondents obtained through the survey used in data collection phase as shown in Fig. C.1.

References

[1] D. Kumar, K.K. Mishra, The impacts of test automation on software's cost, quality and time to market, Procedia Comput. Sci. 79 (2016) 8–15, http://dx.doi.org/10.1016/j.procs.2016.03.003.

[2] B. Maharjan, S. Kautish, Challenges of automated regression testing in agile software development - a qualitative study of selected it companies of Nepal, 2, (1) (2020) 104–118.

[3] E.F. Collins, V.F. De Lucena, Software test automation practices in the agile development environment: An industry experience report, in: 7th Int. Work. Autom. Softw. Test, AST 2012 - Proc., 2012, pp. 57–63, http://dx.doi.org/10.1109/IWAST.2012.6228991.

[4] T. Virtanen, Literature Review of Test Automation Models in Agile Testing (M.S. Thesis), Tampere University of Technology, 2018, [Online]. Available: https://trepo.tuni.fi/handle/123456789/25869.

[5] P. Raulamo-Jurvanen, Decision support for selecting tools for software test automation, ACM SIGSOFT Softw. Eng. Not. 41 (6) (2017) 1–5, http://dx.doi.org/10.1145/3011286.3011304.

[6] K. Hussain, I. Mazhar, An agent-based assisting system for choosing the right, Int. J. Comput. Sci. Telecommun. 9 (2) (2018) 7–11.

[7] D. Ateşoğulları, A. Mishra, Automation testing tools: A comparative view, Int. J. Inf. Comput. Secur. 12 (2020) 63–76.

[8] A.Z. Khan, S. Iftikhar, R.H. Bokhari, Z.I. Khan, Issues/challenges of automated software testing: A case study, Pak. J. Comput. Inf. Syst. 3 (2) (2018) 61–75.

[9] R. Nawaz Bashir, et al., Testing automation in agile software development, Int. J. Innov. Appl. Stud. 9 (2) (2014) 541–546, [Online]. Available: http://www.ijias.issr-journals.org/.

[10] B. Oliinyk, V. Oleksiuk, Automation in software testing, can we automate anything we want? 2019, pp. 224–234.

[11] V. Garousi, M. Felderer, S. Eldh, M. Kuhrmann, Exploring the industry' s challenges in software testing: An empirical study, J. Softw.: Evol. Process. (2020) 1–23, http://dx.doi.org/10.1002/smr.2251.

[12] M. Felderer, V. Garousi, Together we are stronger: Evidence-based reflections on industry-academia collaboration in software testing, in: D. Winkler, S. Biffl, D. Mendez, Bergsmann J. (Eds.), Software Quality: Quality Intelligence in Software and Systems Engineering. SWQD 2020, vol. 371, Springer, 2020, http://dx.doi.org/10.1007/978-3-030-35510-4_1.

[13] A. Putta, M. Paasivaara, C. Lassenius, Benefits and challenges of adopting the scaled agile framework (SAFe): Preliminary results from a multivocal, in: International Conference on Product-Focused Software Process Improvement, Vol. 1, Springer International Publishing, 2018.

[14] J. Kasurinen, O. Taipale, K. Smolander, Software test automation in practice: Empirical observations software test automation in practice: Empirical observations, Adv. Softw. Eng. 2010 (2010) 620836, http://dx.doi.org/10.1155/2010/620836, 18 pages.

[15] W. Rice, R.C. Solutions, Surviving the top ten challenges of software test automation, 2003, pp. 1–14.

[16] Venkat Moncompu, Agile test automation: transition challenges and ways to overcome them, in: Pacific NW Softw. Qual. Conf., 2013, pp. 1–9.

[17] E. Collins, A. Dias-Neto, V.F.d. Lucena Jr., Strategies for agile software testing automation: An industrial experience, in: 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops, 2012, pp. 440–445, http://dx.doi.org/10.1109/COMPSACW.2012.84.

[18] Z. Nikolova, Testing strategies in an agile context, in: S. Goericke (Ed.), The Future of Software Quality Assurance, 2020, http://dx.doi.org/10.1007/978-3-030-29509-7_9.

[19] V. Garousi, F. Elberzhager, Test automation: Not just for test execution, IEEE Softw. 34 (2) (2017) 90–96, http://dx.doi.org/10.1109/MS.2017.34.

[20] Y. Wang, M.V. Mäntylä, S. Demeyer, K. Wiklund, S. Eldh, T. Kairi, Software test automation maturity: A survey of the state of the practice, in: ICSOFT 2020 - Proc. 15th Int. Conf. Softw. Technol., 2020, pp. 27–38, http://dx.doi.org/10.5220/0009766800270038.

[21] B. Gonen, D. Sawant, Significance of agile software development and SQA powered by automation, in: Proc. - 3rd Int. Conf. Inf. Comput. Technol. ICICT 2020, 2020, pp. 7–11, http://dx.doi.org/10.1109/ICICT50521.2020.00009.

[22] V. Garousi, M.V. Mäntylä, When and what to automate in software testing? A multi-vocal literature review, Inf. Softw. Technol. 76 (2016) 92–117, http://dx.doi.org/10.1016/j.infsof.2016.04.015.

[23] F. Ricca, A. Stocco, Web test automation: Insights from the grey literature, in: T. Bureš, et al. (Eds.), SOFSEM 2021: Theory and Practice of Computer Science. SOFSEM 2021, in: Lecture Notes in Computer Science, vol. 12607, Springer, Cham, 2021, http://dx.doi.org/10.1007/978-3-030-67731-2_35.

[24] K. Wiklund, S. Eldh, D. Sundmark, K. Lundqvist, Impediments for software test automation: A systematic literature review, Softw. Test. Verify Reliab. 27 (8) (2017) 1–20, http://dx.doi.org/10.1002/stvr.1639.

[25] C. Polepalle, R.S. Kondoju, Evidence and Perceptions on GUI Test Automation (M.S. Thesis), Blekinge Institute of Computing, Sweden, 2017, [Online]. Available: www.bth.se. [Accessed: Mar. 17, 2021].

[26] D. Lindholm, Economics of Test Automation (M.S. Thesis), Linköping University, 2019.

[27] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - A systematic literature review, Inf. Softw. Technol. 51 (1) (2009) 7–15, http://dx.doi.org/10.1016/j.infsof.2008.09.009.

[28] E. Alégroth, R. Feldt, P. Kolström, Maintenance of automated test suites in industry: An empirical study on visual GUI testing, Inf. Softw. Technol. 73 (2016) 66–80, http://dx.doi.org/10.1016/j.infsof.2016.01.012.

[29] B. Maqbool, F.U. Rehman, M. Abbas, S. Rehman, Implementation of software testing practices in Pakistan ' s software industry, in: Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS 2018), Association for Computing Machinery, New York, NY, USA, 2018, pp. 147–152, http://dx.doi.org/10.1145/3180374.3181340.

[30] G. Ammad, U. Iqbal Janjua, T.M. Madni, M.F. Cheema, A.R. Shahid, An empirical study to investigate the impact of communication issues in GSD in Pakistan's IT industry, IEEE Access 7 (2019) 171648–171672, http://dx.doi.org/10.1109/ACCESS.2019.2953008.

[31] E. Alégroth, Visual GUI Testing: Automating High-Level Software Testing in Industrial Practice (Phd. Dissertation), Chalmers University of Technology, 2015.

[32] S. Spasov, A System for Semi-Automated Testing in Industrial Practice (M.S. Thesis), Chalmers University of Technology, University of Gothenburg, Sweden, 2015.

[33] V. Garousi, D. Pfahl, When to automate software testing? A decision-support approach based on process simulation, J. Softw. Evol. Process 28 (4) (2016) 272–285, http://dx.doi.org/10.1002/smr.1758.

[34] K. Baral, R. Mohod, J. Flamm, S. Goldrich, P. Ammann, Evaluating a test automation decision support tool, in: Proc. - 2019 IEEE 12th Int. Conf. Softw. Testing, Verify. Valid. Work. ICSTW 2019, 2019, pp. 69–76, http://dx.doi.org/10.1109/ICSTW.2019.00034.

[35] R. Kazmi, I. Ghani, R. Mohamad, I.S. Bajwa, Trade-off between automated and manual testing: A production possibility curve cost model, Int. J. Adv. Soft Comput. Appl. 8 (2016) 12–27.

[36] J. Heikkinen, A Benefit Analysis of Utilizing Test Automation in a Telecommunications Company from a Business Perspective (MSc Thesis), University of Helsinki, 2021.

[37] J.A. Khan, S.U.R. Khan, J. Iqbal, I.U. Rehman, Empirical investigation about the factors affecting the cost estimation in global software development context, IEEE Access 9 (2021) 22274–22294, http://dx.doi.org/10.1109/ACCESS.2021.3055858.

[38] D. Larkman, R. Jentzsch, M. Mohammadian, Software testing – factor contribution analysis in a decision support framework, in: J. Watada, G. Phillips-Wren, L.C. Jain, R.J. Howlett (Eds.), Intelligent Decision Technologies. Smart Innovation, Systems and Technologies, Vol 10, Springer, Berlin, Heidelberg, 2011, http://dx.doi.org/10.1007/978-3-642-22194-1_88.

[39] S. Ali, N. Ullah, M.F. Abrar, Z. Yang, J. Huang, Fuzzy multicriteria decision-making approach for measuring the possibility of cloud adoption for software testing, Sci. Program. 2020 (2020) 6597316, http://dx.doi.org/10.1155/2020/6597316, 24 pages.
[40] J.A. Khan, S.U.R. Khan, T.A. Khan, I.U.R. Khan, An amplified COCOMO-II based cost estimation model in global software development context, IEEE Access 9 (2021) 88602–88620, http://dx.doi.org/10.1109/ACCESS.2021.3089870.
[41] F. Aizaz, S.U.R. Khan, J.A. Khan, I.U.R. Khan, A. Akhunzada, An empirical investigation of factors causing scope creep in agile global software development context: A conceptual model for project managers, IEEE Access 9 (2021) 88602–88620, http://dx.doi.org/10.1109/access.2021.3089870.
[42] A. Stouky, B. Jaoujane, R. Daoudi, H. Chaoui, Improving software automation testing using jenkins, and machine learning under big data, in: J. Jung, P. Kim, K. Choi (Eds.), Big Data Technologies and Applications. BDTA 2017, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 248, Springer, 2018, http://dx.doi.org/10.1007/978-3-319-98752-1_10.
[43] M. Sánchez-gordón, R. Colomo-palacios, Beyond technical skills in software testing automated versus manual testing, in: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20), Association for Computing Machinery, New York, NY, USA, 2020, pp. 161–164, http://dx.doi.org/10.1145/3387940.3392238.
[44] A.S. Gadwal, Comparative review of the literature of automated testing tools, 2020, http://dx.doi.org/10.13140/RG.2.2.36836.19848.
[45] P. Raulamo-Jurvanen, M. Mäntylä, V. Garousi, Choosing the right test automation tool: a grey literature review of practitioner sources, in: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (EASE'17), Association for Computing Machinery, New York, NY, USA, 2017, pp. 21–30, http://dx.doi.org/10.1145/3084226.3084252.
[46] L. Paruch, V. Stray, R. Florea, The human factors of the agile software tester, in: M. Shepperd, F. Brito e Abreu, A. Rodrigues da Silva, R. Pérez-Castillo (Eds.), Quality of Information and Communications Technology. QUATIC 2020, in: Communications in Computer and Information Science, vol. 1266, Springer, Cham, 2020, pp. 474–487, http://dx.doi.org/10.1007/978-3-030-58793-2_38.
[47] R. Florea, V. Stray, The skills that employers look for in software testers, Softw. Qual. J. 27 (2019) 1449–1479, http://dx.doi.org/10.1007/s11219-019-09462-5.
[48] R. Florea, V. Stray, Software tester, we want to hire you! An analysis of the demand for soft skills, in: J. Garbajosa, X. Wang, A. Aguiar (Eds.), Agile Processes in Software Engineering and Extreme Programming. XP 2018, in: Lecture Notes in Business Information Processing, vol. 314, Springer, Cham, 2018, http://dx.doi.org/10.1007/978-3-319-91602-6_4.
[49] R. Florea, V. Stray, A qualitative study of the background, skill acquisition, and learning preferences of software testers, in: Proceedings of the Evaluation and Assessment in Software Engineering (EASE '20), Association for Computing Machinery, New York, NY, USA, 2020, pp. 299–305, http://dx.doi.org/10.1145/3383219.3383252.
[50] T. Rautavesi, Test Automation As a Part of Agile Software Development (Bachelor's Thesis), Metropolia University of Applied Sciences, 2019.
[51] L. Paruch, N. Sciences, The Human Factors and the Hard Skills Shaping the Role of the Software Tester in Agile Teams a Grounded Theory Study (M.S. Thesis), University of Oslo, 2020.
[52] S. Azarkerdar, Agile Testing in Scrum, A Study of Major Factors for Successful Agile Testing Implementations in the View of Agile Testers (M.S.Thesis), Uppsala University, 2018.
[53] J. Shim, H. Kwon, H. Jung, M. Hwang, Design of acceptance test process with the application of agile development methodology, Int. J. Control Autom. 9 (2) (2016) 343–352.
[54] A. Rodrigues, Relevance and impact of critical factors of success in software test automation lifecycle: A survey, in: Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing, SAST, Association for Computing Machinery, New York, NY, USA, 2016, 6, http://dx.doi.org/10.1145/2993288.2993302, 1–10.
[55] S. Tyagi, R. Sibal, B. Suri, Adopting test automation on agile development projects: A grounded theory study of Indian software organizations, in: H. Baumeister, H. Lichter, M. Riebisch (Eds.), Agile Processes in Software Engineering and Extreme Programming. XP 2017, in: Lecture Notes in Business Information Processing, vol. 283, Springer, Cham, 2017, http://dx.doi.org/10.1007/978-3-319-57633-6_12.
[56] H.V. Gamido, M.V. Gamido, Comparative review of the features of automated software testing tools, Int. J. Electr. Comput. Eng. 9 (2019) 4473–4478, http://dx.doi.org/10.11591/ijece.v9i5.
[57] P. Sabev, K. Grigorova, Manual to automated testing: An effort-based approach for determining the priority of software test automation, Int. J. Comput. Electr. Autom. Control Inf. Eng. 9 (12) (2015) 2456–2462.
[58] A. Rauf, E.M. Reddy, Software test automation: An algorithm for solving system management automation problems, in: Procedia - Procedia Comput. Sci. Icict 2014, Vol. 46, 2015, pp. 949–956, http://dx.doi.org/10.1016/j.procs.2015.01.004.
[59] K. Shaukat, U. Shaukat, F. Feroz, S. Kayani, A. Akbar, Taxonomy of automated software testing tools, Int. J. Comput. Sci. Innov. 1 (2017) 7–18.
[60] J. Walker, Enhancing the Test and Evaluation Process: Implementing Agile Development, Test Automation, and Model-Based Systems Engineering Concepts (Doctoral dissertation), Colorado State University, 2020.
[61] D. Gafurov, A.E. Hurum, M. Markman, Achieving test automation with testers without coding skills: an industrial report, in: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE 2018), Association for Computing Machinery, New York, NY, USA, 2018, pp. 749–756, http://dx.doi.org/10.1145/3238147.3240463, 749–756.
[62] P.R. Jurvanen, K. Kakkonen, M. Mäntylä, Using surveys and web-scraping to select tools for software testing consultancy, in: P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, T. Mikkonen (Eds.), Product-Focused Software Process Improvement. PROFES 2016, in: Lecture Notes in Computer Science, vol. 10027, Springer, Cham, 2016, http://dx.doi.org/10.1007/978-3-319-49094-6_18.
[63] M.A. Inam, Test Automation Assessment (MS. Thesis), National University of Computer and Emerging Sciences, 2019, http://dx.doi.org/10.13140/RG.2.2.24583.75688.
[64] V. Stray, R. Florea, L. Paruch, Exploring human factors of the agile software tester, Softw. Qual. J. (2021) http://dx.doi.org/10.1007/s11219-021-09561-2.

Shimza Butt received the BS degree in Information Technology from GC Women University Sialkot, Pakistan, in 2018. Currently, she is pursuing the MS degree in Software Engineering from COMSATS University Islamabad (CUI), Islamabad Pakistan. Her research interests include software testing, software project management, software process improvement, and artificial intelligence.

Saif Ur Rehman Khan received his bachelor degree in Computer Science from Mohammad Ali Jinnah University (MAJU) Islamabad Pakistan in 2005, Master's degree in Software and System Engineering from MAJU, Islamabad Pakistan in 2007, and the Ph.D. degree in Computer Science (Software Engineering) from University of Malaya (UM), Kuala Lumpur, Malaysia, in 2018. He is in the faculty of Computer Science Department of COMSATS University Islamabad (CUI), Islamabad Pakistan since 2005, and in-charge of Software Reliability Engineering Group (SREG) at CUI. His research interests in software engineering include verification and validation, search-based software engineering, cyber–physical systems, requirements engineering, and software project management. He has published numerous research articles in high-impact journals and peer-reviewed conferences. He has been in several expert review panels, both locally and internationally. He was a recipient of the Best Paper Presentation Award from UM in 2014, and a Certificate of Outstanding Contribution in Reviewing from FGCS journal in 2018.

Shahid Hussain completed his Ph.D. in Computer Science from City University of Hong Kong, MS (Software Engineering) from City University of Science and Information Technology, Peshawar, Pakistan and MSc (Computer Science) from Gomal University, DIK, Pakistan. He is currently working as an Assistant Professor at the Computer Science and Software Engineering Department, School of Penn State University Behrend, Erie, PA, USA. Before Joining PSU, he has worked as a Pro temp (NTTF) at the Computer and Information Science Department, University of Oregon, Eugene, Oregon, USA. Prior to this, he was working as an Assistant Professor in COMSATS University, Islamabad. He has worked as course chair and acting Director Administration at Namal College, Mianwali an affiliated College of University of Bradford, UK. Moreover, he has worked as a faculty member of University of Peshawar, Center of Applied Epistemics, and CECOS University. He has worked as programmer with SRDC, British Council, Peshawar, Pakistan. He is a research collaborator at SERC, MID Lab, Smart Intelligence team, SERC-KUST, SE-CU (HK) team, SE-NUAA (CN) from September 2014 to date. His current research focus is to explore the software engineering practices for High Performance Computing (HPC) software packages and applications to improve their process and benchmark their quality levels, to automate the document generation for open-source software, and to explore the implications of Machine learning, Artificial Intelligence (AI), Big data analysis and soft computing technique in analysis and design of softwares.

Dr. Wen-Li Wang is an associate professor of Computer Science and Software Engineering at Penn State Behrend. He received the B.S. in Management Information Systems from NCCU, Taipei, Taiwan in 1991, and both the M.S. and Ph.D. in Computer Science from the State University of New York SUNY at Albany in 1996 and 2002 respectively where he received the best thesis award in 2003 for his work. Dr. Wang joined the Penn State Behrend faculty in August of 2002. Dr. Wang's first engineering job was as a LAN administrator for the Department of MIS in Taipei, Taiwan in 1993–1994. He then worked as a graduate assistant and teaching assistant while completing his education at SUNY-Albany. Dr. Wang is a member of the Institute of Electrical and Electronic Engineers IEEE.