IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 8, 2019, accepted May 8, 2019, date of publication May 20, 2019, date of current version August 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917913

The Acceptance of Search-Based Software Engineering Techniques: An Empirical Evaluation Using the Technology Acceptance Model

VITALIY MEZHUYEV[1], MOSTAFA AL-EMRAN[2], MOHD ARFIAN ISMAIL[1], LUIGI BENEDICENTI[3], AND DURKAHPUVANESVARI A/P CHANDRAN[1]

[1]Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang, Gambang 26300, Malaysia
[2]Applied Computational Civil and Structural Engineering Research Group, Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[3]Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada

Corresponding author: Mostafa Al-Emran (al.emran@tdtu.edu.vn)

This work was supported by Universiti Malaysia Pahang, Malaysia, under Grant RDU190311.

ABSTRACT Numerous topics in software engineering can be formulated as optimization problems. Due to the large scale of modern software systems, the methods of mathematical optimization have high computational complexity, and their application in many cases is not possible. To overcome this problem, search-based software engineering (SBSE) develops and applies metaheuristic search techniques to find near-optimal solutions. Despite the significant development that the SBSE techniques (SBSET) achieved in the last years, the level of SBSET use is very low. Therefore, this paper proposes and tests a model which evaluates and predicts the acceptance of SBSET by software engineering practitioners. The model is based on the technology acceptance model (TAM) and extended by the SBSET desired properties. A total of 163 practitioners participated in the study. The perceived ease of use, perceived usefulness, and organizational and team-based factors were all found to have significant positive effects on the actual use of the SBSET. This study also analyses the obstacles of SBSET acceptance and provides several proposals for its improvement in the software industry.

INDEX TERMS Search-based software engineering, techniques, acceptance, technology acceptance model.

I. INTRODUCTION

Since the formulation of the term Search-Based Software Engineering (SBSE) [1], SBSE has attracted many research theories and applications. SBSE converts a software engineering problem into a computational search problem. The technique is based on the definition of the pool of possible solutions, which is known as the search space. Taking into account the complexity of modern software systems, the search space is too large to be explored exhaustively. To find a sufficiently good solution, SBSE follows the procedure that allows the selection of a potential solution from the search space based on given criteria.

SBSE techniques (SBSET) have been applied at almost all the phases of the software development process starting from the requirement analysis [2], following software design, development, and refactoring [3], debugging and testing [4], and ending by the software product configuration and maintenance. Despite the large number of available SBSET and its validation on several benchmark problems, their adoption by the software development industry is quite limited [5]. An explanation of the above-mentioned phenomenon is typically restricted by the complexity of SBSET and the possible inertia of the software industry. For many software engineering research, it is argued that there is a limited impact of academic research results on industrial practices [6]. In our vision, the lack of theoretical foundations for the adoption and corresponding empirical results limits the potential contribution of SBSET to the modern software development companies.

To overcome this limitation, this paper is an attempt to explore the reasons about why software engineers may or may not accept SBSET. More specifically, the main contribution of this paper is to explore the factors affecting the SBSET acceptance by software engineering practitioners. To achieve this aim, this paper develops and evaluates a model, obtained

The associate editor coordinating the review of this manuscript and approving it for publication was Wasif Afzal.

by expanding the Technology Acceptance Model (TAM) that was developed by Davis [7] with the factors related to SBSET. These factors were deducted from the SBSE-related literature and other studies, and linked to the empirical evaluation of software technologies.

II. LITERATURE REVIEW

A. SEARCH-BASED SOFTWARE ENGINEERING

According to the literature, it is pointed out that the most widely used SBSE algorithms are genetic algorithms, simulated annealing, and hill climbing [8]. It is also argued that testing is the most used application domain of SBSE, followed by software product configuration [9]. Research indicates that SBSE is natural in software product lines to solve the problems of featuring the same characterization. The most commonly used SBSE techniques in this domain are the multi-objective evolutionary algorithms [10]. Promising results were achieved in the application of an ant colony algorithm to optimize the selection of users' requirements [11].

According to a systematic literature review on the selection of software requirements and their prioritization using SBSET [12], SBSET was successfully applied to the optimization of software structure (e.g., clustering software packages) and the different aspects of project management. To reduce a stakeholder dissatisfaction, an assessment of cost, value, and risk was incorporated into the multi-objective selection of requirements [13]. To assist a software release planning, a multi-objective search has been integrated into the requirements management tool [14]. Research also explains the use of SBSE for software refactoring by providing examples of ill-defined fitness functions [15].

There is a number of systematic literature reviews and surveys conducted on the different aspects of SBSE. The technical report written by Harman and Jones [8] is one of the first that provides a detailed analysis of trends, techniques, and applications of SBSE. Close to the context of SBSE research, Mantere and Alander [16] conducted a review on the evolutionary software engineering. Furthermore, Harman and his colleagues conducted a taxonomy and tutorial of SBSET concerning the use of SBSET [17]. Another review study was conducted to review the search-based testing for non-functional system properties [18]. Moreover, another review study was carried out to review the applications of search-based test case generation [19]. Scholars also systematically analyzed interaction in SBSE [20]. Harman and his research team also discussed the achievements, open problems, and challenges for search-based software testing [21].

As a result, there are numerous research reports made by the academy. As it can be noticed, SBSET allows to reduce software design time, decrease the cost of software testing and maintenance, and therefore, it can produce more reliable software. It is also anticipated that SBSET can help avoiding error-prone work in the different aspects of software engineering process.

B. EVALUATION CRITERIA FOR SBSET

Since the application of SBSET to search the near-optimal solutions is quite effective, the use of SBSET has acquired high elaboration. The effectiveness of SBSET refers to its ability to obtain an optimal result, while the efficiency indicates the capability to maintain the performance in producing the optimal result in several case studies. Researchers reported that SBSET outperforms many existing analytical optimization methods and random search algorithms. At the same time, in the implementation perspective, SBSET lends itself naturally to algorithms parallelization. It is pointed out that SBSE techniques can help to automate a specific problem-solving efficiently [22]. SBSET has the ability to solve very complex problems when exact solutions cannot be found in a reasonable time, and this refers to one of the strengths that leads SBSE to success [23].

Research also indicates the advantages of SBSE which include robustness, scalability, possibility to create links between apparently unconnected SE disciplines, and the source of insight [24]. The application of SBSET can take place across different phases of the SE life cycle. As such, SBSET could help in solving software engineering problems due to its robust nature, the capability to manage conflicting objectives, and to find a comprehensive resolution [22].

It is argued that no specific knowledge regarding search algorithms is needed for practitioners to use SBSET [23]. In that, it is concluded that SBSE tools can be used as "black boxes" because the internal details are hidden, and no understanding of search algorithms is required. In line with this, Marculescu and his research team conducted an industrial evaluation of interactive search-based testing for embedded software by domain experts, who had a little or no knowledge of SBSE [25].

At the same time, the strength of SBSE comes with the consideration of these models as "white-boxes", where the users play the central role. Such type of search makes the possibility of an interactive reformulation of a fitness function, user-defined software metrics, effective managing conflicting objectives, and making hybridization of algorithms to achieve the best performance.

Simultaneously, many researchers discussed the properties of SBSET, which prevent its adoption. SBSET needs clear understanding and quantification of what should be measured (especially for the definition of objective functions). The problem is that SE practitioners attempt to find a solution based on quantitative measurements. Research has already discussed the problem "what actually software metric represents" with respect to "what exactly should be measured", and "how to make measurable what is not measurable" [26]. The role of human interactivity is considered very prominent in the metaheuristic search [27]. Here, it is quite typical to combine a quantitative fitness, determined by software metrics, with qualitative developer evaluation [28].

While there are benefits in using SBSET, the barrier is tuning, where parameters need to be set correctly, else it will result in the impossibility to solve a problem [23].

TABLE 1. SBSET properties.

Perceived usefulness	Perceived usefulness includes various properties such as user's expectations, performance, automation, a reasonable time to find a solution, managing conflicting objectives, scalability, possibility to solve complex problems, comprehensive resolution, robustness, validity, and lack of the trust to SBSET (i.e., too many approaches and week predictability of results).	[22]–[24].
Perceived ease of use	The properties of perceived ease of use include simplicity, no specific knowledge needed, short learning curve, difficulties in the definition of software metrics and measurement, the complexity of parameters tuning, difficulties in interpretability and predictability of results, and lack of practical examples (i.e., the absence of SBSE stability and mature tools).	[23], [24], [33], [35], [25]–[32].
Organizational and team-based factors	The properties of organizational and team-based factors include the absence of an organizational support, lack of engineer's competencies, the link between SBSE disciplines and software life cycle applicability, and academy-industry agreements.	[24], [31], [37], [38].

SBSET might find it hard to solve the highly constrained problems, although it can provide good solutions for simple constrained optimization problems. For example, the strength of constraints for requirements optimization is measured based on the number and complexity of interactions between the requirements. The problem is more difficult to solve as the constraints become tighter [29]. Another barrier to the SBSET adoption is a lack of software engineer trust to existing tools [30]. Due to the significant range of the competencies of a software engineer, achieving trust in SBSE tools is non-trivial [31]. Simons emphasized the necessity and areas of industrial adoption of SBSE. According to a study conducted to evaluate the industrial relevance in SBSE research [32], it is pointed out that in order to accept SBSET, good tools are needed.

Another problem is the absence of industrial examples as some academicians might tend to focus on the published papers only. It is also emphasized that a common and frequent complaint from software practitioners is that academic research doesn't meet their requirements or expectations [33]. At the same time, it is highly imperative to mention that some SBSE tools have been successfully adopted, and EvoSuite [34] and Sapienz [4] are the prime examples of SBSE tools that are highly effective in practice.

Research showed that the availability of reliable tools was perceived as the most influential factor in the adoption of model-driven engineering (MDE) [35]. For the evaluation of MDE adoption, it is found that perceived usefulness, ease of use, and maturity of the tools are the most important determinants that affect the MDE adoption [36]. Despite the emergence of scale and complexity challenges, it is also discovered that organizational, cultural, and team-based challenges predominate the SE development process [37]. Thus, there is a clear gap between SBSE innovation and SBSET adoption by practitioners. Table 1 summarises the properties of SBSET which may encourage or prevent its adoption and actual use. It generalizes the properties into groups reflecting the SBSET perceived usefulness (performance, effectiveness, and robustness), perceived ease of use (simplicity, maturity, and process integrability), and organizational and team-based factors.

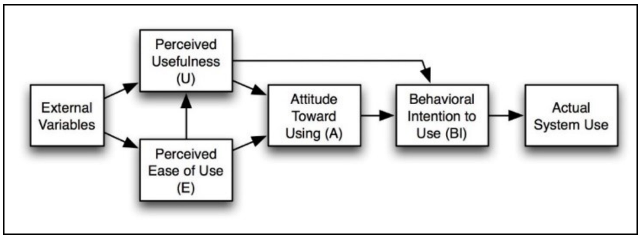

FIGURE 1. TAM model [39].

C. TECHNOLOGY ACCEPTANCE MODEL

The Technology Acceptance Model (TAM) [39] was developed on the basis of the Theory of Reasoned Action (TRA) [40] and the Theory of Planned Behaviour (TPB) [41]. Figure 1 illustrates the TAM model.

The intention of Davis was to develop a model for testing the acceptance of a new information system (IS) by the end-users [39]. TAM tends to explain why individuals adopt or not adopt a particular IS [42]. Since 1986, TAM was expanded and applied by many researchers to the different domains beyond IS [43]. In Software Engineering (SE), TAM was employed to check the acceptance of CASE systems [44], Object-Oriented Programming [45], software measures [46], and metamodeling [47] among many others. Researchers in SE proved that TAM could explain not only the acceptance of an IS or a particular software but also the processes involved under these systems. For instance, TAM has been applied to the software process improvement initiatives because the reasons for accepting a new initiative is similar to that for accepting a new technology [48].

According to TAM, two main factors could affect the adoption of technology, and these two factors include Perceived Ease of Use (PEOU) and Perceived Usefulness (PU) [7]. PEOU refers to the degree to which acquiring a technology will require minimum physical or mental efforts. PU refers to the beliefs of an individual in improving the job performance when using a specific technology. The assumption of TAM is that if an individual believes that the use of a particular technology will not be complex and will increase the usefulness of the final product, he/she would be more likely to use this technology.

This paper analyses the effect of both PEOU and PU on SBSET acceptance. SBSET should be perceived as useful and easy to use, or else SE practitioners will not adopt it. Understanding why SE practitioners may or may not accept SBSET is an important step to increase the effectiveness of the software development industry.

There are many studies devoted to the predictors of PU and PEOU in an attempt to facilitate technology acceptance [7]. There are several examples of the PU indicators. For instance, the use of technology can increase the productivity of work and job performance. The advantages of technology can overweight the disadvantages. The use of technology can provide valuable results and can assist the work in the future. Similarly, there are various issues regarding the PEOU indicators. For example, the use of technology can make the work easier. The use of technology can make it easier to produce valuable results. Also, it is possible to use a technology without expert help. These indicators are general and can be applied to the evaluation of adopting any information technology [7]. Many researchers argued the necessity of understanding what specific properties of technology will influence PU and PEOU. It is pointed out that these properties can't be fully managed to encourage individuals' perception [46]. This creates a continuous need for understanding the factors affecting these two constructs (i.e., PU and PEOU) to achieve higher levels of acceptance.

III. RESEARCH MODEL AND HYPOTHESES DEVELOPMENT

According to the prior studies in the literature and building upon the TAM, we were able to identify the predictors that could affect the perceived usefulness and perceived ease of use of SBSET. In that, perceived performance, perceived effectiveness, and perceived validity and robustness were suggested to have a significant impact on perceived usefulness of SBSET. It is also assumed that perceived simplicity, perceived maturity, and perceived integrability may have a significant effect on perceived ease of use of SBSET. In addition, the organizational and team-based factors, perceived usefulness, and perceived ease of use are posited to have a significant impact on SBSET acceptance. Figure 2 illustrates the proposed research model and its corresponding hypotheses.

A. PERCEIVED PERFORMANCE

Perceived performance (PP) is about "tuning so that the average user on a multi-user system perceives that the system is operating faster for them" [49]. In the SE context, PP is regarded as a critical factor that facilitates the process of SE projects [50]. Despite its importance in the SE literature, there is no research devoted to exploring its effect on SE acceptance in general, nor its relation to TAM constructs in specific. Therefore, this study suggests that PP may have a stronger effect on perceived usefulness of SBSET. Accordingly, the following is hypothesized:

H1: PP has a positive effect on perceived usefulness of SBSET.

B. PERCEIVED EFFECTIVENESS

Perceived effectiveness (PE) refers to "the degree to which using a technology will provide benefits to consumers in performing certain activities" [51]. In terms of SE, research indicated that PE is an essential factor in studying the distributed requirements engineering [52]. It is also pointed out that PE had a positive influence on metamodeling acceptance [47] and computer-aided software engineering technology [53]. In this study, it is suggested that the higher the effectiveness of SBSET, the higher the usefulness of SBSET would be. Therefore, this leads to the following hypothesis:

H2: PE has a positive effect on perceived usefulness of SBSET.

C. PERCEIVED VALIDITY AND ROBUSTNESS

Perceived validity and robustness (PVR) refers to "how well respondents thought the selection methods predicted future job performance" [54]. Research argued that PVR is an essential property of perceived usefulness [22]. In this study, it is assumed that PVR may have a stronger effect on perceived usefulness of SBSET. Accordingly, we hypothesize the following:

H3: PVR has a positive effect on perceived usefulness of SBSET.

D. PERCEIVED SIMPLICITY

Perceived simplicity (PS) is the key success indicator of products' ease of use [55]. It is believed that PS is an essential factor that affects the perceived ease of use of different systems [56]. In this study, it is suggested that the higher the simplicity of SBSET, the higher the easiness of the SBSET would be. Accordingly, we hypothesize the following:

H4: PS has a positive effect on perceived ease of use of SBSET.

E. PERCEIVED MATURITY

Perceived maturity (PM) refers to "the degree to which tools are perceived as mature and suitable for the tasks in hand" [36]. In the SE context, research indicated that PM had a positive effect on the acceptance of object-oriented programming [57], model-driven engineering [36], [58], and metamodeling [47]. In the current study, it is posited that the higher the reliability and maturity of SBSET, the higher the easiness of SBSET would be. Thus, this leads to the following hypothesis:

H5: PM has a positive effect on perceived ease of use of SBSET.

F. PERCEIVED INTEGRABILITY

Perceived integrability or integration (PI) refers to "the ability to make the separately developed components of the system work correctly together" [59]. In the SE context, PI is a critical element that precedes the development of any

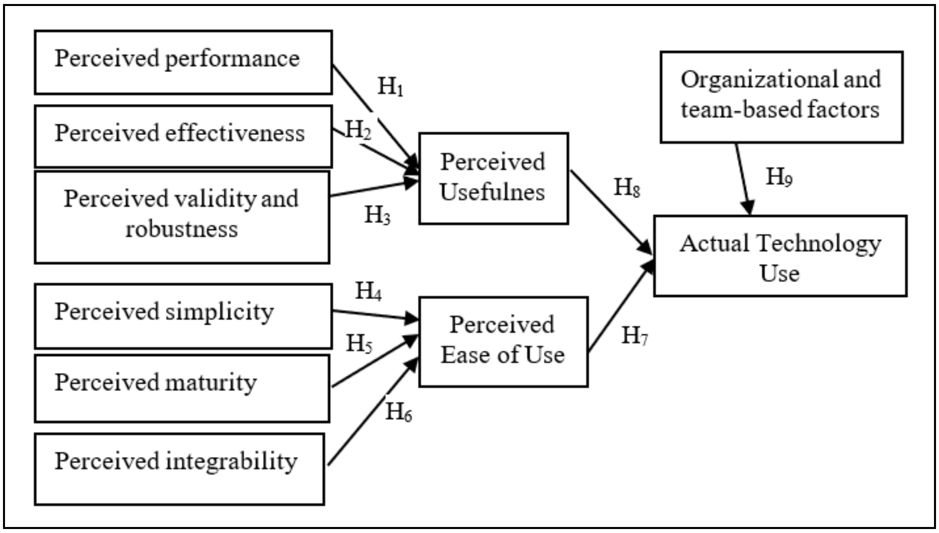

FIGURE 2. Research model and hypotheses.

software [60]. It is also assumed that PI is an essential factor in SE environments in which the integration of user-interface, software functions, and data format can be smoothly performed [61]. In this study, it is assumed that PI may have a stronger impact on perceived ease of use of SBSET. We, therefore, hypothesize the following:

H6: PI has a positive effect on perceived ease of use of SBSET.

G. PERCEIVED EASE OF USE

Perceived ease of use (PEOU) refers to "the degree to which a person believes that using a particular system would be free from effort" [7]. For many research in technology acceptance, PEOU has been found to affect the acceptance of several technologies including social software [62], software utilization [63], e-payment systems [64], Google classroom [65], and cloud computing [66], among many others. In the SE context, research indicated that PEOU has a positive influence on metamodeling acceptance [47] and software measures [46]. Given its great influence on different technologies, this study suggests that PEOU may have a stronger effect on SBSET acceptance. We, therefore, hypothesize the following:

H7: PEOU has a positive effect on SBSET acceptance.

H. PERCEIVED USEFULNESS

Perceived usefulness (PU) is defined as "the degree to which a person believes that using a particular system would enhance his or her job performance" [7]. In the SE context, research revealed that PU had a significant effect on the acceptance of several technologies such as metamodeling [47], software measures [46], and electronic process guide [38]. In this study, it is assumed that PU may have a stronger effect on SBSET acceptance. Thus, we hypothesize the following:

H8: PU has a positive effect on SBSET acceptance.

I. ORGANIZATIONAL AND TEAM-BASED FACTORS

Organizational and team-based factors (OTBF) is regarded as "the key factor in predicting the acceptance of several technologies and systems" [47]. In the context of SE, research indicated that OTBF has a positive effect on metamodeling acceptance [47] and model-driven engineering [67], [68]. In this study, it is posited that if the organization provides sufficient support and training for SBSET, there is a great chance that the end-users' acceptance of SBSET will be increased. Thus, the following is hypothesized:

H9: OTBF has a positive effect on SBSET acceptance.

IV. RESEARCH METHODOLOGY

A. DATA COLLECTION

The target respondents in this study are academics, researchers, and software engineers working in different software development companies. The emails of the respondents were allocated through the scientific papers and technical reports published in the last five years on the different aspects of SBSE. An email including a letter and a reference to the online survey was sent to 1307 participants. The letter includes a short motivation for the survey as well as the aim of the study. For those who didn't respond to the survey after a few days from the first email, a follow-up email as a gentle reminder has been sent to them. Out of 1307 emails sent, only 183 responses were received, resulting in a response rate of 14%. 20 responses were discarded as they were incomplete. Thus, the total valid responses which can be used for further analysis is 163.

B. SURVEY INSTRUMENT

The questionnaire survey consists of four different sections. The first section is related to the personal information of the respondent (i.e., gender, age, country, educational level, and occupation). The second section comprises of details regarding the employment and possible type of SBSE projects.

The third section consists of the research model constructs which were measured using a 5-point Likert scale, ranging from "1 = strongly disagree" to "5 = strongly agree". The fourth section includes open-ended questions regarding the problems preventing the acceptance of SBSET and proposals to increase the level of SBSET acceptance. The constructs and their corresponding items are illustrated in the Appendix.

V. RESULTS AND DISCUSSION

A. DATA ANALYSIS

The data analysis in this study was carried out using the partial least squares-structural equation modeling (PLS-SEM) through SmartPLS V.3.2.7 software [69]. This study follows the guidelines for using PLS-SEM in information systems research [70]. In that, the collected data were analyzed using a two-step assessment approach including the measurement model and structural model [71]. Since this study is exploratory in nature, PLS-SEM is regarded as the best choice to suit such type of studies [72].

B. DESCRIPTIVE ANALYSIS

A total of 163 valid responses were received. Of those responses, 66.87% were males, and 33.13% were females. The majority of respondents' age range varied between 26 and 35 years old, with 45.4% (N = 74), this is followed by those who range between 18 and 25 years old, with 25.77% (N = 42), 36 and 45 years old, with 20.86%, and those who were above 46 years old, with 7.97% (N = 13). Moreover, most of the respondents are bachelor's degree holders with 42.33%, this is followed by doctorate's degree holders with 31.9%, master's degree holders with 17.79%, and undergraduate degrees with 7.97%. It is worth mentioning that 53.37% of the respondents are working in software development companies, whereas 46.63% of them are working in academia (i.e., academics or researchers). Table 2 shows the respondents' demographics details.

It is also worth noticing that the respondents are working in different companies and academic institutions, in which most of these environments have 250 employees and above. Most of the software development companies' respondents are working in mature software companies, which age over seven years old. In terms of experience with SBSE, the majority of the respondents has 3-6 years of practice (46.8%), while 29.8% of them have more than 6 years of experience. With respect to the SBSET application domains, most of the mainstream approaches are related to the software testing and debugging, software design, while the requirements prioritization and the software maintenance share the third place (see Figure 3). PhD projects and other research projects are considered the dominant ones as compared with the other types of SBSE applications (see Figure 4).

At the same time, most of the respondents indicated that there is a high relevance between SBSE-related projects and their industrial applications (see Figure 5). To analyse the effort on developing new SBSET or applying existing ones, the respondents were asked to answer the question "Did you invent a new SBSE algorithm or tool?". Figure 6 shows that 77.7% of the respondents reported the use of an existing SBSET, whereas only 22.3% were involved in the development of new algorithms or tools. Concerning the use of SBSET in an interactive way, the respondents were asked to answer the question "Did you modify an existing SBSE algorithm to improve its performance or search results?". Figure 7 shows that the majority of the respondents (74.7%) reported their efforts on performing modifications on SBSE algorithms; thus, this shows a clear preference of using SBSET as "white-boxes" models.

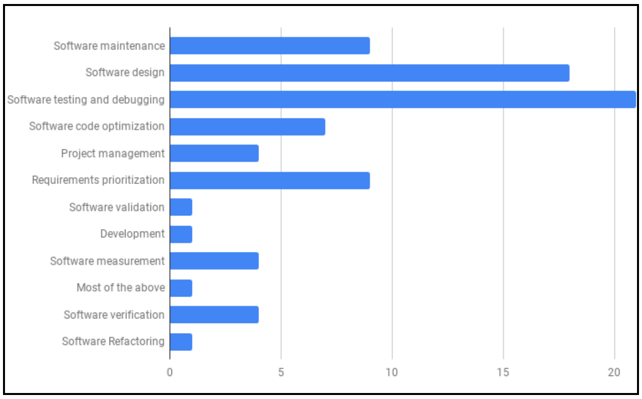

FIGURE 3. SBSET application domains.

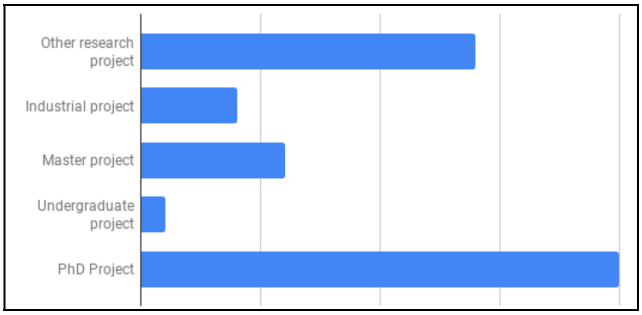

FIGURE 4. Types of SBSE projects.

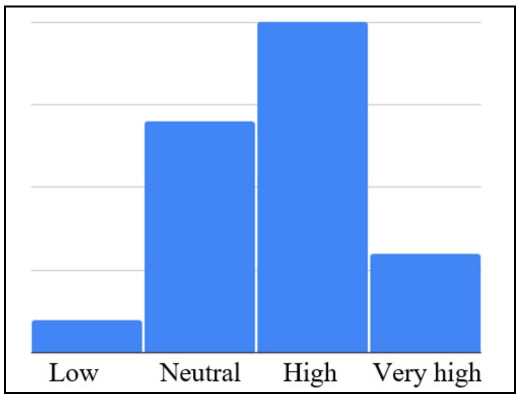

FIGURE 5. Evaluation of industrial relevance of the projects.

TABLE 2. Respondents' demographics.

Items	Values	Frequency	Percentage
Gender	Male	109	66.87
	Female	54	33.13
Age	From 18 to 25	42	25.77
	From 26 to 35	74	45.40
	From 36 to 45	34	20.86
	Above 46	13	7.97
Educational Level	Undergraduate	13	7.97
	Bachelor	69	42.33
	Master	29	17.79
	Doctorate	52	31.90
Type of Employment	Academia	76	46.63
	Software industry	87	53.37

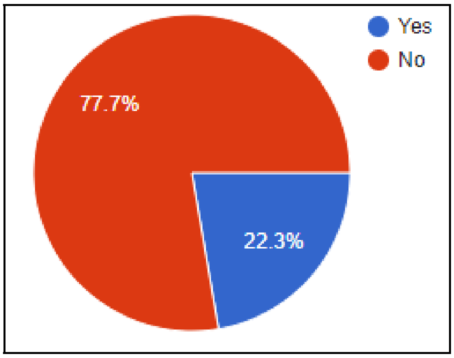

FIGURE 6. The usage of existing or invention of new SBSET.

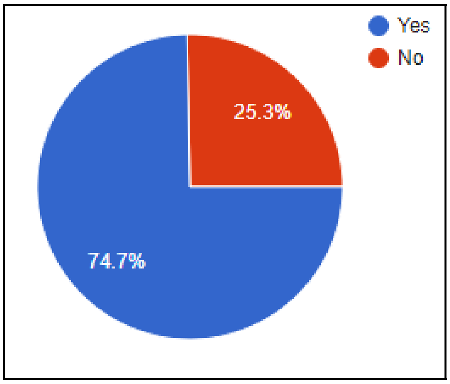

FIGURE 7. Efforts in modifying SBSE algorithms.

C. MEASUREMENT MODEL ASSESSMENT

Before testing the hypotheses in the structural model, the evaluation of the measurement model should be confirmed. The reason for analyzing the measurement model is to ensure that the measures used are sound and they provide the understood theoretical parts adequately. The assessment of the measurement model includes measuring the reliability (Cronbach's Alpha and composite reliability) and validity (convergent and discriminant validity).

For testing the internal consistency reliability, the results in Table 3 reveal that the Cronbach's Alpha values are ranged between 0.733 and 0.932 which were all higher than the threshold value of 0.7 [71]. The results also revealed that the composite reliability (CR) values are ranged between 0.836 and 0.957 which were all higher than the suggested value of 0.7. Therefore, the internal consistency reliability in terms of Cronbach's Alpha and CR is confirmed.

Convergent validity refers to the measurement of which multiple items are similar in concept. In this study, factor loadings and average variance extracted (AVE) were tested to examine the convergent validity [73]. AVE refers to the grand mean value of the squared loadings of the indicators related to the construct [74]. As per Table 3, the values of factor loadings have fulfilled the requirements as all of these values were higher than the recommended value of 0.7. The results in Table 3 also indicate that the AVE values are ranged between 0.631 and 0.881 which were all higher than the suggested value of 0.5 [71]. Given these results, the convergent validity is ascertained.

Discriminant validity is generally an accepted precondition for analyzing the relationships between latent variables. It aims to ensure that a reflective construct has the strongest relationships with its own indicators in comparison with other constructs in the PLS path model [71]. Henseler and his research team have proposed a new criterion for assessing the discriminant validity, which is known as the "Heterotrait-Monotrait ratio (HTMT)" [75]. In that, the HTMT ratio has been shown to have higher performance compared to the previous criteria. Based on the previous assumptions and according to the recommendations of Hair *et al.* [73], this study employs the HTMT criterion for assessing the discriminant validity. According to Table 4, it can be deduced that the HTMT criterion is met as all the values were less than the threshold value of 0.85 [75]. Thus, this indicates that the discriminant validity is established.

TABLE 3. Constructs reliability and validity.

Constructs	Items	Factors loading	Cronbach's Alpha	Composite reliability	Average variance extracted
Perceived performance	PP1	0.847	0.794	0.880	0.709
	PP2	0.791			
	PP3	0.886			
Perceived effectiveness	PE1	0.864	0.841	0.903	0.756
	PE2	0.885			
	PE3	0.859			
Perceived validity and robustness	PVR1	0.828	0.847	0.907	0.766
	PVR2	0.890			
	PVR3	0.905			
Perceived simplicity	PS1	0.932	0.932	0.957	0.881
	PS2	0.984			
	PS3	0.898			
Perceived maturity	PM1	0.857	0.733	0.836	0.631
	PM2	0.786			
	PM3	0.736			
Perceived integrability	PI1	0.899	0.929	0.955	0.877
	PI2	0.986			
	PI3	0.923			
Perceived usefulness	PU1	0.888	0.893	0.933	0.823
	PU2	0.913			
	PU3	0.920			
Perceived ease of use	PEOU1	0.813	0.796	0.881	0.711
	PEOU2	0.873			
	PEOU3	0.843			
Organizational and team-based factors	OTBF1	0.893	0.767	0.863	0.679
	OTBF2	0.804			
	OTBF3	0.770			
Actual Use	AU1	0.766	0.761	0.863	0.678
	AU2	0.828			
	AU3	0.872			

TABLE 4. HTMT ratio results.

	AU	OTBF	PEOU	PE	PI	PM	PP	PS	PU	PVR
AU										
OTBF	0.255									
PEOU	0.202	0.597								
PE	0.640	0.484	0.589							
PI	0.446	0.558	0.628	0.731						
PM	0.603	0.180	0.458	0.327	0.457					
PP	0.751	0.415	0.497	0.814	0.620	0.319				
PS	0.149	0.354	0.814	0.356	0.332	0.203	0.297			
PU	0.715	0.378	0.546	0.817	0.682	0.232	0.832	0.340		
PVR	0.529	0.443	0.591	0.569	0.731	0.395	0.649	0.536	0.811	

D. STRUCTURAL MODEL ASSESSMENT

The explanatory power of the proposed model was evaluated by measuring the discrepancy amount in the dependent variables. According to Hair et al. [71], the coefficient of determination (R^2) and path coefficients are considered as the essential measures for assessing the structural model. In terms of path analysis, Figure 8 and Table 5 demonstrate the path coefficients, t-values, and p-values for each hypothesis. It can be noticed that all the hypotheses are supported, which in turn indicates that all the paths are significant between the independent and dependent variables.

Hypothesis 1 ($B = 0.253$, $p < 0.05$) describes the path between perceived performance and perceived usefulness; indicating that the perceived performance enhances the perceived usefulness of SBSET. Hypothesis 2 ($B = 0.359$, $p < 0.01$) shows the path between perceived effectiveness and perceived usefulness; representing that the perceived effectiveness leverages the perceived usefulness of SBSET. Hypothesis 3 ($B = 0.396$, $\text{p} < 0.01$) demonstrates the path between perceived validity and robustness and perceived usefulness; revealing that the perceived validity and robustness positively influences the perceived usefulness of SBSET.

Hypothesis 4 ($B = 0.556$, $\text{p} < 0.001$) describes the path between perceived simplicity and perceived ease of use; indicating that the perceived simplicity significantly affects the perceived ease of use of SBSET. Hypothesis 5 ($B = 0.567$, $p < 0.001$) describes the path between perceived maturity and perceived ease of use; indicating that

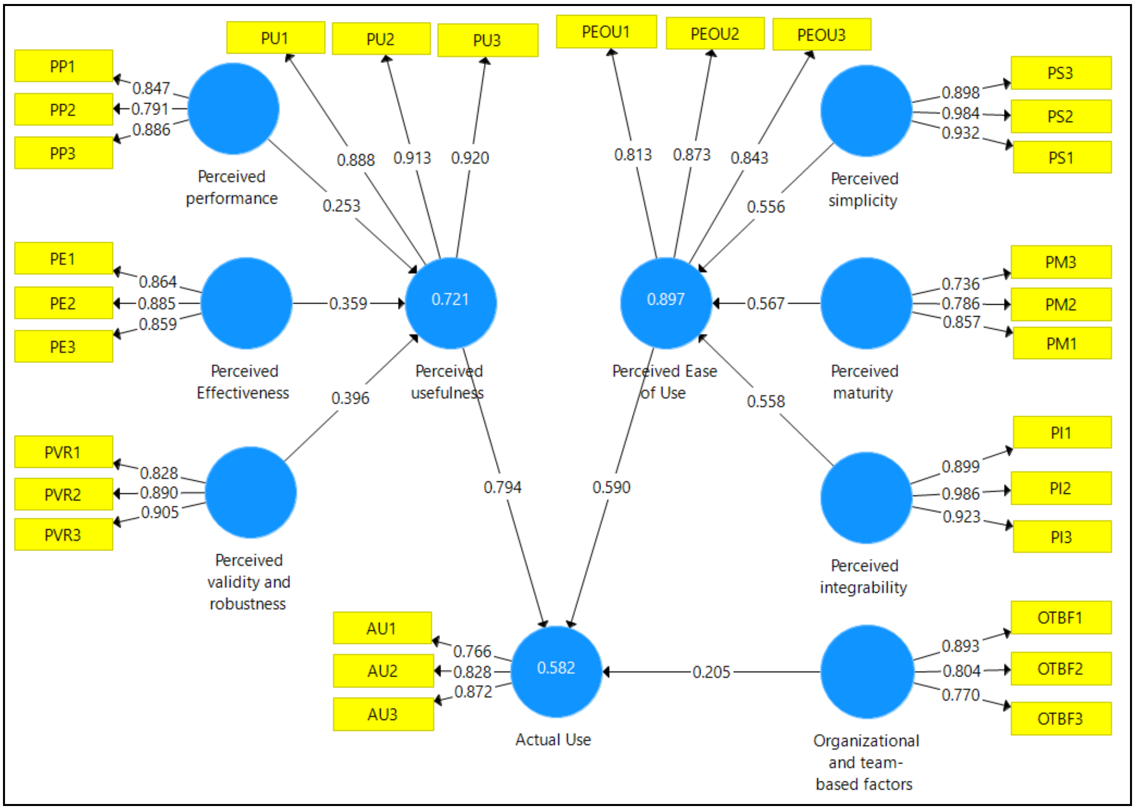

FIGURE 8. PLS algorithm results.

TABLE 5. Hypotheses testing results.

Hypothesis	Path	Path Coefficient	*p*-values	Remarks
H_1	PP → PU	0.253	0.046	Supported
H_2	PE → PU	0.359	0.001	Supported
H_3	PVR → PU	0.396	0.002	Supported
H_4	PS → PEOU	0.556	0.000	Supported
H_5	PM → PEOU	0.567	0.000	Supported
H_6	PI → PEOU	0.558	0.000	Supported
H_7	PEOU → AU	0.590	0.000	Supported
H_8	PU → AU	0.794	0.000	Supported
H_9	OTBF → AU	0.205	0.014	Supported

the perceived maturity enhances the perceived ease of use of SBSET. Hypothesis 6 ($B = 0.558$, $p < 0.001$) shows the path between perceived integrability and perceived ease of use; representing that the perceived integrability leverages the perceived ease of use of SBSET.

Hypothesis 7 ($B = 0.590$, $p < 0.001$) demonstrates the path between perceived ease of use and the actual use of SBSET; revealing that the perceived ease of use positively influences the actual use of SBSET. Hypothesis 8 ($B = 0.794$, $p < 0.001$) describes the path between perceived usefulness and the actual use of SBSET; indicating that the perceived usefulness significantly affects the actual use of SBSET. Hypothesis 9 ($B = 0.205$, $p < 0.05$) demonstrates the path between the organizational and team-based factors and the actual use of SBSET; revealing that the organizational and team-based factors positively influence the actual use of SBSET.

With respect to the (R^2), the results in Figure 8 indicate that perceived performance, perceived effectiveness, and perceived validity and robustness explain 72.1% of the variance in perceived usefulness. It is also revealed that perceived simplicity, perceived maturity, and perceived integrability explain 89.7% of the variance in perceived ease of use. More interesting, perceived usefulness, perceived ease of use, and organizational and team-based factors explain 58.2% of the variance in the actual use of SBSET. In compliance with the recommended values of (R^2) [76], the achieved (R^2) values in this study are regarded to be extremely acceptable.

VI. CHALLENGES AND OPPORTUNITIES FOR SBSET USE

This section analyses the responses to the open-ended questions about problems preventing the use of SBSET and proposals to increase the level of SBSET acceptance.

A. PROBLEMS PREVENTING THE USE OF SBSET

One of the good examples that was given by the respondents regarding the problems hindering the SBSET use is "There are multiple problems. The relative complexity of SBSE together with a relative lack of support tool mean that the cost of adoption is quite high. Specialist support is needed to ensure that the problem is formulated in a way that is meaningful for SBSE and relevant to the company. Initial adoption of SBSE in the industry is less like adopting a tool and more like conducting a research project.". The rest of the answers can be generalized into four different groups. First, the lack of user-friendly tools and the absence of at least one killer application. Second, the internal problems of SBSE such as difficulties in parameters tuning and the scalability to real-world problems. Third, the lack of empirical research can mislead the easier use of SBSET. Fourth, the absence of awareness and lack of human-competitive results. The absence of mature and user-friendly tools is the main problem triggered out by the SBSE practitioners. The respondents also pointed out that most of the existing SBSE tools are just approaches proposals. The previous problems are considered the reasonable ones that may hinder the adoption of SBSET in industry.

In addition, some studies have attempted to study the phenomenon of human competitiveness in SBSE [77]. The approach proposed by Barr *et al.* [78] has won an award regarding human competitiveness.

B. PROPOSALS FOR IMPROVING SBSET INDUSTRIAL ACCEPTANCE

Correspondingly with the formulated problems, the respondents have suggested different proposals for improving the industrial acceptance of SBSET which can be generalized into three different groups. First, the development of mature tools. Second, the development of more generic approaches and solutions. Third, the development of SBSET practical examples and demonstrating them to the industrial practitioners.

There are other possible improvements related to the translation of SBSET from the academia into the industry, in which, it needs the application of methods from the economic domain. This includes: 1) the return on investment of SBSET in the real world, 2) the development of SBSET cost-benefit analysis, 3) the use of interns, placements, and contract work, 4) orientation and integration into the process of software development companies, and 5) investigating the potential of SBSE solutions reuse in other projects.

VII. CONCLUSION

SBSE is a discipline that concentrates on the applicability of search-based optimization techniques to solve SE problems. While SBSET has received considerable attention from a number of software development companies, others are reluctant to accept it. In order to provide a common platform for evaluating the SBSET acceptance, this paper took the advantages of TAM to achieve this target. The TAM has been extended by the factors from the literature, which researchers assume to be important for the usefulness and ease of use of SBSET (i.e., perceived performance, effectiveness, validity and robustness, simplicity, maturity, and integrability). It was also assumed that organizational and team-based factors had an influence on the acceptance of SBSET.

The results of this research study indicated that perceived ease of use along with its determinants (perceived simplicity, perceived maturity, and perceived integrability), perceived usefulness along with its predictors (perceived performance, perceived effectiveness, and perceived validity and robustness), and organizational and team-based factors positively affect the actual use of SBSET. In that, the SBSET was perceived to have good performance, effectiveness, validity, robustness, simplicity, maturity, and process integrability, and the practitioners are highly motivated toward the use of SBSET in their organizations.

Together with the investigation of industrial acceptance of SBSET using TAM, the study also analyzed the obstacles of SBSET acceptance and provided several proposals for its improvement in the software industry. The development of mature tools and more generic approaches (e.g., the introduction of parameter-free meta-heuristics) are the most critical directions that need to be considered. The acceptance of SBSET by industry would also require an investigation of the return on investment, cost-benefit analysis, and the effective ways of integrating the SBSE into the business processes of software development companies.

Although the current study yielded significant results, it also posits some limitations that need to be considered in future attempts. First, the sample size is relatively small. Second, by taking the previous limitation into consideration, the entire collected data were analyzed as a whole (i.e., academics and software engineers together). Thus, it is highly imperative if future trials would take these limitations into consideration and attempt to separately analyze the developed model in terms of academics and software engineers.

APPENDIX
CONSTRUCTS AND CORRESPONDING ITEMS

Perceived Performance (PP)

PP1. SBSE techniques have good performance.

PP2. SBSE techniques allow to achieve valuable results in a reasonable time frame.

PP3. SBSE techniques allow to achieve comprehensive resolution.

Perceived Effectiveness (PE)

PE1. SBSE techniques provide effective solutions.

PE2. SBSE techniques allow to automate problem-solving.

PE3. SBSE techniques allow to manage conflicting objectives.

Perceived Validity and Robustness (PVR)

PVR1. SBSE techniques have been extensively empirically validated.

PVR2. The robustness of SBSET is proven.

PVR3. SBSE techniques produce trustworthy problem solutions.

Perceived Simplicity (PS)
PS1. It is difficult to reformulate a SE problem as a search problem.
PS2. It is difficult to define software metrics to formulate a fitness function.
PS3. It is hard to solve the highly constrained problems.
Perceived Maturity (PM)
PM1. SBSE techniques are not mature enough.
PM2. SBSE techniques do not have mature and reliable tools.
PM3. SBSE techniques lack of real-world examples.
Perceived Integrability (PI)
PI1. SBSE techniques can be easily integrated into the business process of a software company.
PI2. SBSE techniques can be used at any stage of SE development process.
PI3. SBSE techniques can be used repeatedly throughout the development process.
Perceived Usefulness (PU)
PU1. The use of SBSE techniques is beneficial for me.
PU2. SBSE techniques allow solving complex problems.
PU3. SBSE techniques increase the productivity and enhance the quality of software development.
Perceived Ease of Use (PEOU)
PEOU1. I believe that SBSE techniques are easy to use.
PEOU2. No specific knowledge is needed to apply SBSE techniques.
PEOU3. I can use SBSE techniques without special training and expert help.
Organizational and Team-Based Factors (OTBF)
OTBF1. The profile of my company corresponds to SBSE techniques.
OTBF2. My organization provides training and tools for SBSE techniques.
OTBF3. Lack of corresponding to SBSE competencies can neglect the success of SBSE techniques application.
Actual Use (AU)
AU1. I use SBSE techniques frequently.
AU2. I use SBSE techniques in daily practice.
AU3. My commitment to SBSE techniques is high.

REFERENCES

[1] M. Harman and B. F. Jones, "Search-based software engineering," *Inf. Softw. Technol.*, Jul. 43, pp. 833–839, Dec. 2001.

[2] Y. Zhang, M. Harman, G. Ochoa, G. Ruhe, and S. Brinkkemper, "An empirical study of meta-and hyper-heuristic search for multi-objective release planning," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 1, p. 3, 2018.

[3] M. Paixao, M. Harman, Y. Zhang, and Y. Yu, "An empirical study of cohesion and coupling: Balancing optimization and disruption," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 394–414, Jun. 2018.

[4] K. Mao, M. Harman, and Y. Jia, "Sapienz: Multi-objective automated testing for Android applications," in *Proc. 25th Int. Symp. Softw. Test. Anal.*, 2016, pp. 94–105.

[5] B. Marculescu, R. Feldt, R. Torkar, and S. Poulding, "Transferring interactive search-based software testing to industry," *J. Syst. Softw.*, vol. 142, pp. 156–170, Aug. 2018.

[6] A. Arcuri, "An experience report on applying software testing academic results in industry: We need usable automated test generation," *Empirical Softw. Eng.*, vol. 23, no. 4, pp. 1959–1981, 2018.

[7] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quart.*, vol. 13, no. 3, pp. 319–340, 1989.

[8] M. Harman, S. A. Mansouri, and Y. Zhang, "Search based software engineering: A comprehensive analysis and review of trends techniques and applications," Dept. Comput. Sci., King's College London, London, U.K., Tech. Rep. TR-09-03, 23, 2009, pp. 1–78.

[9] R. E. Lopez-Herrejon, L. Linsbauer, and A. Egyed, "A systematic mapping study of search-based software engineering for software product lines," *Inf. Softw. Technol.*, vol. 61, pp. 33–51, May 2015.

[10] M. Harman, Y. Jia, J. Krinke, W. B. Langdon, J. Petke, and Y. Zhang, "Search based software engineering for software product line engineering: A survey and directions for future work," in *Proc. 18th Int. Softw. Product Line Conf.*, vol. 1, 2014, pp. 5–18.

[11] J. del Sagrado, I. M. del Águila, and F. J. Orellana, "Multi-objective ant colony optimization for requirements selection," *Empirical Softw. Eng.*, vol. 20, no. 3, pp. 577–610, 2015.

[12] A. M. Pitangueira, R. S. P. Maciel, and M. Barros, "Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature," *J. Syst. Softw.*, vol. 103, pp. 267–280, May 2015.

[13] A. M. Pitangueira, P. Tonella, A. Susi, R. S. Maciel, and M. Barros, "Risk-aware multi-stakeholder next release planning using multi-objective optimization," in *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Cham, Switzerland: Springer, 2016.

[14] I. M. Del Águila and J. Del Sagrado, "Three steps multiobjective decision process for software release planning," *Complexity*, vol. 21, no. S1, pp. 250–262, 2016.

[15] B. Amal, M. Kessentini, S. Bechikh, J. Dea, and L. Ben Said, "On the use of machine learning and search-based software engineering for ill-defined fitness function: A case study on software refactoring," in *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Cham, Switzerland: Springer, 2014.

[16] T. Mantere and J. T. Alander, "Evolutionary software engineering, a review," *Appl. Soft Comput.*, Jul. 5, pp. 315–331, Mar. 2005.

[17] M. Harman, P. McMinn, J. T. de Souza, and S. Yoo, "Search based software engineering: Techniques, taxonomy, tutorial," *Lecture Notes in Computer Science* (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2011.

[18] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Inf. Softw. Technol.*, vol. 51, pp. 957–976, Jun. 2009.

[19] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege, "A systematic review of the application and empirical investigation of search-based test case generation," *IEEE Trans. Softw. Eng.*, vol. 36, no. 6, pp. 742–762, Nov./Dec. 2010.

[20] A. Ramirez, J. R. Romero, and C. Simons, "A systematic review of interaction in search-based software engineering," *IEEE Trans. Softw. Eng.*, to be published.

[21] M. Harman, Y. Jia, and Y. Zhang, "Achievements, open problems and challenges for search based software testing," in *Proc. IEEE 8th Int. Conf. Softw. Test., Verification Validation (ICST)*, Apr. 2015, pp. 1–12.

[22] R. Malhotra, "Special issue on search-based techniques and their hybridizations in software engineering," *Comput. Lang. Syst. Struct.*, vol. 47, no. P2, pp. 151–152, 2017.

[23] A. Arcuri and G. Fraser, "Parameter tuning or default values? An empirical investigation in search-based software engineering," *Empirical Softw. Eng.*, vol. 18, pp. 594–623, Jun. 2013.

[24] M. Harman and A. Mansouri, "Search based software engineering: Introduction to the special issue of the IEEE transactions on software engineering," *IEEE Trans. Softw. Eng.*, vol. 36, no. 6, pp. 737–741, Nov./Dec. 2010.

[25] B. Marculescu, R. Feldt, R. Torkar, and S. Poulding, "An initial industrial evaluation of interactive search-based testing for embedded software," *Appl. Soft Comput.*, vol. 29, pp. 26–39, Apr. 2015.

[26] N. Fenton, "Software metrics: A rigorous and practical approach," in *Computer*. Long Beach, CA, USA: CRC Press, 1998.

[27] C. L. Simons, "Whither (away) software engineers in SBSE?" in *Proc. 1st Int. Workshop Combining Modelling Search-Based Softw. Eng. (CMSBSE)*, 2013, pp. 49–50.

[28] C. L. Simons and I. C. Parmee, "Elegant object-oriented software design via interactive, evolutionary computation," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1797–1805, Nov. 2012.

[29] Y. Zhang, M. Harman, and S. L. Lim, "Empirical evaluation of search based requirements interaction management," *Inf. Softw. Technol.*, vol. 55, pp. 126–152, Jan. 2013.
[30] D. Meignan, S. Knust, J.-M. Frayret, G. Pesant, and N. Gaud, "A review and taxonomy of interactive optimization methods in operations research," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 3, p. 17, 2015.
[31] C. Simons, "The truth is out there: Reflections on search in software engineering," in *Proc. 1st Int. Summer School Search-Based Softw. Eng.*, 2016, pp. 1–15.
[32] V. Appana, *Evaluating Industrial Relevance in Search Based Software Engineering Research: A Systematic Mapping Study and Survey*. Karlskrona, Sweden: Faculty of Computing, Blekinge Institute of Technology, 2017.
[33] S. Freudenberg and H. Sharp, "The top 10 burning research questions from practitioners," *IEEE Softw.*, vol. 27, no. 5, pp. 8–9, Sep./Oct. 2010.
[34] G. Fraser and A. Arcuri, "EvoSuite: Automatic test suite generation for object-oriented software," in *Proc. 19th ACM SIGSOFT Symp. 13th Eur. Conf. Found. Softw. Eng.*, 2011, pp. 416–419.
[35] M. Staron, "Adopting model-driven software development in industry—A case study at two companies," in *Proc. ACM/IEEE 9th Int. Conf. Model Driven Eng. Lang. Syst. (MoDELS/UML)*, 2006, pp. 57–72.
[36] P. Mohagheghi, W. Gilani, A. Stefanescu, and M. A. Fernandez, "An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases," *Empirical Softw. Eng.*, vol. 18, pp. 89–116, Feb. 2013.
[37] P. Gregory, L. Barroca, H. Sharp, A. Deshpande, and K. Taylor, "The challenges that challenge: Engaging with agile practitioners' concerns," *Inf. Softw. Technol.*, vol. 77, pp. 92–104, Sep. 2016.
[38] T. Dybå, N. B. Moe, and E. M. Mikkelsen, "An empirical investigation on factors affecting software developer acceptance and utilization of electronic process guides," in *Proc. 10th Int. Symp. Softw. Metrics*, Sep. 2004, pp. 220–231.
[39] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User acceptance of computer technology: A comparison of two theoretical models," *Manage. Sci.*, vol. 35, pp. 982–1003, Aug. 1989.
[40] I. Ajzen and M. Fishbein, *Understanding Attitudes and Predicting Social Behavior*. Prentice-Hall, 1980.
[41] I. Ajzen, "From intentions to actions: A theory of planned behavior," in *Action Control*, Berlin, Germany: Springer, 1985, pp. 11–39.
[42] M. Turner, B. Kitchenham, P. Brereton, S. Charters, and D. Budgen, "Does the technology acceptance model predict actual use? A systematic literature review," *Inf. Softw. Technol.*, vol. 52, pp. 463–479, May 2010.
[43] F. D. Davis, "A technology acceptance model for empirically testing new end-user information systems: Theory and results," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep., 1986.
[44] P. Y. K. Chau, "An empirical investigation on factors affecting the acceptance of CASE by systems developers," *Inf. Manage.*, vol. 30, no. 6, pp. 269–280, 1996.
[45] B. C. Hardgrave and R. A. Johnson, "Toward an information systems development acceptance model: The case of object-oriented systems development," *IEEE Trans. Eng. Manag.*, vol. 50, no. 3, pp. 322–336, Aug. 2003.
[46] L. G. Wallace and S. D. Sheetz, "The adoption of software measures: A technology acceptance model (TAM) perspective," *Inf. Manag.*, vol. 51, no. 2, pp. 249–259, 2014.
[47] V. Mezhuyev, M. Al-Emran, M. Fatehah, and N. C. Hong, "Factors affecting the Metamodelling Acceptance: A Case Study from Software Development Companies in Malaysia," *IEEE Access*, vol. 6, pp. 49476–49485, 2018.
[48] M. Umarji and C. Seaman, "Predicting acceptance of software process improvement," *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–6, 2005.
[49] T. Mangan, "Perceived Performance-Tuning a system for what really matters," TMurgent Technol., Canton, MA, USA, Tech. Rep., 2003.
[50] J. J. Jiang, G. Klein, H.-G. Hwang, J. Huang, and S.-Y. Hung, "An exploration of the relationship between software development process maturity and project performance," *Inf. Manage.*, vol. 41, no. 3, pp. 279–288, 2004.
[51] V. Venkatesh, J. Y. L. Thong, and X. Xu, "Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology," *MIS Quart.*, vol. 36, no. 1, pp. 157–178, 2012.
[52] W. J. Lloyd, M. B. Rosson, and J. D. Arthur, "Effectiveness of elicitation techniques in distributed requirements engineering," in *Proc. IEEE Joint Int. Conf. Requirements Eng.*, Sep. 2002, pp. 311–318.
[53] A. Rai, "External information source and channel effectiveness and the diffusion of CASE innovations: An empirical study," *Eur. J. Inf. Syst.*, vol. 4, no. 2, pp. 93–102, 1995.
[54] N. J. Harris, "The perceived validity and reported use of management selection methods in New Zealand organisations," Massey Univ., Wellington, New Zealand, Tech. Rep., 1991.
[55] J. Maeda, *The Laws of Simplicity*. MIT Press, 2006.
[56] D. Lee, J. Moon, and Y. J. Kim, "The effect of simplicity and perceived control on perceived ease of use," in *Proc. AMCIS*, 2007, pp. 1–17.
[57] I. Cho and Y. Kim, "Critical factors for assimilation of object-oriented programming languages," *J. Manag. Inf. Syst.*, vol. 18, no. 3, pp. 125–156, 2001.
[58] S. Walderhaug, M. Mikalsen, I. Benc, and S. Erlend, "Factors affecting developers' use of MDSD in the Healthcare domain: Evaluation from the MPOWER project," in *Proc. Workshop Eur. Conf. Model-Driven Archit.*, 2008.
[59] M. Mari and N. Eila, "The impact of maintainability on component-based software systems," in *Proc. 29th Euromicro Conf.*, Sep. 2003, pp. 25–32.
[60] D. B. Walz, J. J. Elam, and B. Curtis, "Inside a software design team: Knowledge acquisition, sharing, and integration," *Commun. ACM*, vol. 36, no. 10, pp. 63–77, 1993.
[61] I. Thomas and B. A. Nejmeh, "Definitions of tool integration for environments," *IEEE Softw.*, vol. 9, no. 2, pp. 29–35, Mar. 1992.
[62] S. Kim, "Factors affecting the use of social software: TAM perspectives," *Electron. Library*, vol. 30, no. 5, pp. 690–706, 2012.
[63] M. G. Morris and A. Dillon, "The influence of user perceptions on software utilization: Application and evaluation of a theoretical model of technology acceptance," *IEEE Trans. Softw. Eng.*, vol. 14, no. 4, pp. 58–65, Jul. 1997.
[64] S. A. Salloum and M. Al-Emran, "Factors affecting the adoption of E-payment systems by University students: Extending the TAM with trust," *Int. J. Electron. Bus.*, vol. 14, no. 4, pp. 371–390, 2018.
[65] R. A. S. Al-Maroof and M. Al-Emran, "Students acceptance of Google classroom: An exploratory study using PLS-SEM approach," *Int. J. Emerg. Technol. Learn.*, vol. 13, no. 6, pp. 112–123, 2018.
[66] I. Arpaci, "Antecedents and consequences of cloud computing adoption in education to achieve knowledge management," *Comput. Hum. Behav.*, vol. 70, pp. 382–390, May 2017.
[67] J. Hutchinson, J. Whittle, M. Rouncefield, and S. Kristoffersen, "Empirical assessment of MDE in industry," in *Proc. 33rd Int. Conf. Softw. Eng. (ICSE)*, 2011, pp. 471–480.
[68] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, "A taxonomy of tool-related issues affecting the adoption of model-driven engineering," *Softw. Syst. Model.*, vol. 16, no. 2, pp. 313–331, 2017.
[69] C. M. Ringle, S. Wende, and J.-M. Becker, "SmartPLS 3. Bönningstedt: SmartPLS," Tech. Rep., 2015.
[70] M. Al-Emran, V. Mezhuyev, and A. Kamaludin, "PLS-SEM in information systems research: A comprehensive methodological reference," in *Proc. 4th Int. Conf. Adv. Intell. Syst. Inform. (AISI)*, 2018, pp. 644–653.
[71] J. Hair, C. L. Hollingsworth, A. B. Randolph, and A. Y. L. Chong, "An updated and expanded assessment of PLS-SEM in information systems research," *Ind. Manag. Data Syst.*, vol. 117, no. 3, pp. 442–458, 2017.
[72] N. Urbach and F. Ahlemann, "Structural equation modeling in information systems research using partial least squares," *J. Inf. Technol. Theory Appl.*, vol. 11, no. 2, pp. 5–40, 2010.
[73] J. F. Hair, Jr., G. T. M. Hult, C. Ringle, and M. Sarstedt, *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*. Thousand Oaks, CA, USA: Sage Publications, 2016.
[74] J. F. Hair, Jr., G. T. M. Hult, C. Ringle, and M. Sarstedt, *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*. Thousand Oaks, CA, USA: Sage Publications, 2014.
[75] J. Henseler, C. M. Ringle, and M. Sarstedt, "A new criterion for assessing discriminant validity in variance-based structural equation modeling," *J. Acad. Marketing Sci.*, vol. 43, no. 1, pp. 115–135, 2015.
[76] W. W. Chin, "The partial least squares approach to structural equation modeling," *Mod. methods Bus. Res.*, vol. 295, no. 2, pp. 295–336, 1998.
[77] J. T. de Souza, C. L. Maia, F. G. de Freitas, and D. P. Coutinho, "The human competitiveness of search based software engineering," in *Proc. 2nd Int. Symp. Search Based Softw. Eng.*, Sep. 2010, pp. 143–152.
[78] E. T. Barr, M. Harman, Y. Jia, A. Marginean, and J. Petke, "Automated software transplantation," in *Proc. Int. Symp. Softw. Test. Anal.*, 2015, pp. 257–269.

VITALIY MEZHUYEV received the specialist degree in informatics from Berdyansk State Pedagogical University (BSPU), Ukraine, in 1997, the Ph.D. degree in educational technology from Kiev National Pedagogical University, in 2002, and the Sc.D. degree in information technology from Odessa National Technical University, Ukraine, in 2012. From 2004 to 2014, he was a Head of the Department of Informatics and Software Engineering, BSPU. He is currently a Professor with the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, and the Head of the Software Engineering Research Group. During his career, he participated in the multiple international scientific and industrial projects, devoted to the formal modeling, design, and development of advanced software systems as a network-centric real-time operating system, IDEs for the automation of development of parallel real-time applications, tools for specification, verification, and validation of software products, and visual environment for metamaterials modeling. His current research interests include formal methods, metamodeling, safety modeling and verification of hybrid software systems, and the design of cyber-physical systems.

MOSTAFA AL-EMRAN received the B.Sc. degree (Hons.) in computer science from Al Buraimi University College, the M.Sc. degree (Hons.) in informatics from The British University in Dubai, (with the first-class Academic Excellence Award), and the Ph.D. degree in computer science from Universiti Malaysia Pahang. He has published over 40 research papers in highly reputed journals such as the *Computers & Education*, the *International Journal of Information Management*, the *Computers in Human Behavior*, the IEEE ACCESS, the *Journal of Educational Computing Research*, and the *International Journal of Engineering Education*, among many others. Most of his publications were indexed under the ISI Web of Science and Scopus. His current research interests include mobile learning, knowledge management, technology acceptance, and data mining. He is a certified recognized Reviewer by several leading journals in Elsevier.

MOHD ARFIAN ISMAIL received the Ph.D. degree in computer science from the Universiti Teknologi Malaysia. He is currently a Senior Lecturer with the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang. His expertise includes soft computing methods, computational systems biology, machine learning, fuzzy systems, and artificial neural networks.

LUIGI BENEDICENTI received the Laurea degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from the University of Genoa, Italy. He is currently a Full Professor and a Dean with the Faculty of Computer Science, University of New Brunswick. His current research areas include software agents, software process, and new media technology.

DURKAHPUVANESVARI A/P CHANDRAN is currently pursuing the bachelor's degree in computer science (software engineering) with the Faculty of Computer Systems and Software Engineering, Universiti Malaysia Pahang. Her research interest includes application of technology acceptance model in empirical studies.