Adopting DevOps practices: an enhanced unified theory of acceptance and use of technology framework

Ahmad Mahdi Salih1,2, Sharifah Mashita Syed-Mohamad3, Pantea Keikhosrokian1, Nur Hana Samsudin1
1School of Computer Sciences, University Sains Malaysia, Penang, Malaysia
2Mathematics Department, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq
3Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Terengganu, Malaysia

ABSTRACT
DevOps software development approach is widely used in the software engineering discipline. DevOps eliminates the development and operations department barriers. The paper aims to develop a conceptual model for adopting DevOps practices in software development organizations by extending the unified theory of acceptance and use of technology (UTAUT). The research also aims to determine the influencing factors of DevOps practices’ acceptance and adoption in software organizations, determine gaps in the software development literature, and introduce a clear picture of current technology acceptance and adoption research in the software industry. A comprehensive literature review clarifies how users accept and adopt new technologies and what leads to adopting DevOps practices in the software industry as the starting point for developing a conceptual framework for adopting DevOps in software organizations. The literature results have formulated the conceptual framework for adopting DevOps practices. The resulting model is expected to improve understanding of software organizations’ acceptance and adoption of DevOps practices. The research hypotheses must be tested to validate the model. Future work will include surveys and expert interviews for model enhancement and validation. This research fulfills the necessity to study how software organizations accept and adopt DevOps practices by enhancing UTAUT.

Journal homepage: http://ijece.iaescore.com
DevOps, the abbreviation of development-operations, is one of the new widely used movements in the software engineering discipline, specifically in the software development area, since 2009 [6]. DevOps eliminates the barriers between the development and operation departments fetched by the conventional software development structure [7]. It tends to enhance the efficiency and productivity of business activity and push it towards leaner and outcome-oriented [8]. However, adopting DevOps practices is still challenging in most developing countries. Although there are many information tools and practices regarding DevOps, it may still be unclear how to use such rich but scattered information in a structured and organized approach to adopt it [4], [9]. Moreover, few studies are concerned with understanding software development practitioners’ successful paths for adopting DevOps practices in developing countries [4], [10].

Insufficient knowledge of the factors that can help accept and adopt the DevOps approach in developing countries in software development organizations. The software development industry organizations have found it challenging to justify adopting the DevOps approach and its practices and are unaware of its benefits [5], [10]. There is a lack of coherent frameworks that can be used to study the behavioral intention to adopt and implement DevOps practices among information technology (IT) and software development practitioners and identify the crucial factors that need to be included in these frameworks. Those factors will influence the practitioners’ intention to adopt and use DevOps practices [4], [11]. It is significant to investigate how to develop a model that identifies the required factors influencing software development practitioners’ acceptance and adoption of DevOps practices in developing software using these innovative (DevOps) practices.

Software development organizations have followed particular software development models while developing software products. Some of those models are tailored to the organization as an alternative, and organizations can use pre-existing forms [7]. There are four widely used models: software development life cycle (SDLC), rapid application development model (RAD), prototype model, and component assembly models [12]. Mukred et al. [2] have described the standard SDLC as comprising six phases: requirements, specification, design, implementation, maintenance, and retirement.

SDLC is an iterative process repeated many times till the required functional software version is delivered. Developing software products is a dynamic operation characterized by change. Changing the requirements continuously throughout the SDLC may affect the schedule, budget, and product quality [7], [13]. Mukred et al. [2] illustrated that requirements changes could occur anytime during software development. Tseng et al. [14] found that poor client communication is one of the main reasons for delayed changes in requirements.

Departmental silos, a culture of traditional practices in the software development process, have not supported open communication and collaboration among software development teams. This silos culture in software development organizations is deemed one of the fundamental reasons for the failure of software projects [15], [16]. The DevOps software development approach can achieve an organizational culture that permits teams to communicate, collaborate, and share knowledge and resources [17]. Collaboration, sharing, and open communication can lead to high success in project rates, which is evidence of the emergence of software development methodologies like Agile and DevOps [18]. Those methodologies shift to the focus on soft human skills that encourage working towards common goals and having the same visions.

Dyck et al. [19], Jabbari et al. [20], and Sacolick [21] introduced the most comprehensive definitions of DevOps. The DevOps definition by Dyck et al. [19] is “DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions while guaranteeing their correctness and reliability”. Whereas Jabbari et al. [20] define DevOps as “DevOps is a development methodology aimed at bridging the gap between Development and Operations, emphasizing communication and collaboration, continuous integration, quality assurance and delivery with automated deployment utilizing a set of development practices”. Finally, Sacolick [21] presented this definition for DevOps “DevOps is about the culture, collaborative practices, and automation that aligns development and operations teams, so they have a single mindset on improving customer experiences, responding to faster business needs, and ensuring that innovation is balanced with security and operational requirements”.

Generally, DevOps can be put as the junction of the terms Dev (development) and Ops (operations) within the context of information technology. The software development department is responsible for programming and creating new software applications. The IT operations are the services and processes run by the IT department of the organization, whose operations comprise administrative processes and support for software and hardware [3].

The DevOps approach is a set of practices (capabilities and enablers) directed towards frequent releases and deployment, like continuous integration, testing, and deployment [22]. These capabilities also include cultural enablers to break the organizational silos between the development and operations departments, like shared incentives and goals, shared responsibilities, and shared respect and trust. DevOps include a set of technological enablers essential to achieve the abovementioned capabilities, like tools to
automate build, test, and deployment activities [23]. Table 1 provides a brief explanation of DevOps practices.

<table>
<thead>
<tr>
<th>Capabilities</th>
<th>Cultural enablers</th>
<th>Technological enablers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous planning</td>
<td>Shared goals, the definition of success, incentives</td>
<td>Build automation</td>
</tr>
<tr>
<td>Collaborative and continuous development</td>
<td>Shared ways of working, responsibility, collective ownership</td>
<td>Test automation</td>
</tr>
<tr>
<td>Continuous integration and testing</td>
<td>Shared values, respect, and trust</td>
<td>Deployment automation</td>
</tr>
<tr>
<td>Continuous release and deployment</td>
<td>Constant, effortless communication</td>
<td>Monitoring automation</td>
</tr>
<tr>
<td>Continuous infrastructure monitoring and optimization</td>
<td>Continuous experimentation and learning</td>
<td>Recovery automation</td>
</tr>
<tr>
<td>Continuous user behavior monitoring and feedback</td>
<td></td>
<td>Infrastructure automation</td>
</tr>
<tr>
<td>Service failure recovery without delay</td>
<td></td>
<td>Configuration management for code and infrastructure</td>
</tr>
</tbody>
</table>

It is essential to know what makes software development organizations look for new software development approaches like DevOps. This can be found out by summarizing the main problems that come from utilizing the traditional software development approach; those problems involve: the objective of the final products from the perspective of teams of development and operations is not the same; since the success of products is differently measured within development and operations departments, blame-shifting often happens because neither of the teams wants to be responsible for the project’s failure; and inhibitors of releasing the products early in the development life cycle make teams take more time to release the products, in other words, delaying the releasing date [24]. These problems can be resolved by exploiting DevOps practices represented by culture, measurement, automation, monitoring, and knowledge and experience sharing [17].

DevOps is a software engineering approach that arose from the IT industry. DevOps popularity has risen among IT practitioners in the last eight years [25], [26] and has drawn much attention from industry researchers such as [17], [27]–[32]. Organizations that have adopted the DevOps software development approach reported many benefits, like reduced marketing time, effective feature creation, and improved productivity and efficiency [18]. These potential benefits need to be explored to update the old and traditional software development approaches.

This paper is organized as follows: after briefly introducing the topic and reviewing the software development and DevOps concepts, the literature review on adopting DevOps sections is outlined. After that, the research method section is presented. The last section is dedicated to the conclusion and future work.

2. LITERATURE REVIEW

This section discusses the early results of our comprehensive literature review process as shown in Figure 1 to determine users’ acceptance and adoption of new technologies (theories of technology acceptance) and review related studies. After that, gaps in the literature have been determined as the commencing point to develop a conceptual model for the successful adoption of DevOps practices in software development organizations. Using the literature review results, we formulate a conceptual model to adopt DevOps practices in software development organizations.
2.1. Users accept new technologies

Technology acceptance theories are information systems (IS) theories used to model users’ acceptance and implementation of new technologies. The technology acceptance model (TAM) is one of the most widely used of those theories [33]. TAM explains computers’ use, acceptance, and information technologies (ITs) approval with users’ reactions to these technologies. It was developed from the theory of reasoned action (TRA) [34]. TAM’s first version comprises external variables which affect attitude to use indirectly and ultimately lead toward actual use of the system through influencing perceived usefulness (PU) and perceived ease of use (PEU) [35]. TAM “assumes that the effects of external variables (e.g., system design characteristics) on intention are mediated by the key beliefs (i.e., perceived ease of use and perceived usefulness)” [36]. Researchers have developed and improved TAM within the last three decades, which covers different disciplines and situations. The updated TAM (TAM2) version is developed to include the subjective norm variable as an additional intention predictor with compulsory system uses [37]. Venkatesh’s continuous research has developed the TAM3 version [4], [38]. The versions of TAM consider an effective measurement tool that investigates individual’s rejection or acceptance degree of new technologies in their organizations [39].

Another influential acceptance theory that can be used in technology adoption is unified theory of acceptance and use of technology (UTAUT). UTAUT model is based on the extension of TAM and other acceptance theories like TRA, motivation model (MM), theory of planned behavior (TPB), model of PC utilization (MPCU), combined TAM and TPB, diffusion of innovation theory (DOI) and social cognitive theory (SCT) [40]. UTAUT presumes the constructs: performance expectancy (PE), effort expectancy (EE), facilitating conditions (FC), and social influence (SI) have indirect and direct influences on behavioral intention (BI) and user behavior. The relations among these constructs are moderated by age, gender, voluntariness of use, and experience [40], [41].

Technology acceptance theories can be used to measure user perceptions like innovations, adoption, and acceptance in different fields and disciplines. The highest number of measurement studies that used technology acceptance theories are accounted for information and communications technology (ICT) field [42]–[52]. Adopting and implementing new technologies can be modeled using technology acceptance theories. Many studies have been measured using the acceptance and implementation of innovations for users in different fields and disciplines, particularly IT and ICT. Some of those studies extended acceptance theories’ constructs for enhancing the comprehension of the adoption and use of innovations and technologies and how these constructs can be used in different contexts, including the software development industry [39], [53]. Those studies have encouraged software industry researchers to adopt similar methods. These methods measure stakeholder perceptions and beliefs regarding adopting and utilizing software development innovations like DevOps. Moreover, adopting these methods can enhance the software organization’s productivity. The following section briefly reviews DevOps practices adoption in software development organizations.

2.2. Review related studies

The research suggests that software development organizations have benefited from DevOps adoption to reshape the software development industry [7], [17]. For example, Masombuka and Mnkandla [8] have developed a model identifying factors necessary to accept a DevOps collaboration culture in software organizations. The authors found the factors of open communication, trust, and respect to be the critical success factors that should be included in their model. They also illustrated that the factor of social influence plays an essential role in predicting behavioral intent in most domains, including software development. However, they found that role and responsibility factors were not statistically significant [8]. Christensen [54] investigate the advantages and limitations of adopting DevOps to boost the continuous delivery process. The authors found that implementing DevOps practices include increasing team productivity, improving software quality, faster marketing time, saving costs, and creating a new culture of a business-friendly approach [54]. Pérez-Sánchez et al. [49] and Mubarkoot [7] proposed approaches to assess how adopting DevOps practices can enhance organizational culture, leading to better performance and productivity. They also found challenges and difficulties in adopting DevOps practices. These challenges and difficulties summarize into: i) disintermediation risks of roles and responsibilities, ii) lack of understanding of DevOps practices, iii) change resistance and fear of failure in adopting DevOps, iv) lack of awareness because of misinterpretation, miscommunication, and v) insufficient knowledge and adequate training. We can notice that most of these challenges imply that adopting and implementing DevOps and its practices is complex and require much effort from those organizations to achieve [7], [49].

Many researchers have found and developed appropriate approaches (models, frameworks, and strategies) to adopt DevOps practices. However, these approaches have limitations and weaknesses. For example: Leite et al. [55] investigate the factors hindering DevOps adoption and propose strategies to address them. Still, they did not provide practical evidence of their proposed framework’s validity [55]. Ganeshan and Vigneshwaran [56] presented a theory about DevOps adoption, yet, the authors admitted that their
qualitative research contains some degree of research bias, and their study results cannot be valid for other scenarios (lack generalizability) [56]. Mansour and Qureshi [57] identified the problems related to adopting DevOps concerning change-resistant; their study had limited questionnaire surveys (30 respondents) which may cause inaccurate findings. Moreover, the researchers did not provide practical evidence for their proposed solution [57]. Rafi et al. [58] have developed a readiness model for DevOps adoption, yet, the study followed only quantitative methodology; therefore, the findings of the study can be more accurate if conducting mixed research methodology (quantitative and qualitative) [58].

We can observe a lack and limitations of empirical evaluations for the benefits of the DevOps approach. It can be said that it is insufficient to assess whether adopting DevOps practices quantitatively produces improvement in software development [4], [59]. Those results support the need for empirical studies that evaluate the DevOps approach’s effectiveness related to enhancing software development [26], [60]. Moreover, we can also notice a lack of perceptions and knowledge of the successful acceptance of DevOps and its practices and a lack of identifying factors necessary for successful adoption. In addition to the challenges and difficulties in accepting and adopting DevOps, there are no clear guidelines that software development organizations can follow to adopt DevOps and its practices as well as there is a need for a well-designed research methodology.

The studies utilizing UTAUT in adopting DevOps practices are relatively rare and involve limitations and weaknesses. Elberzhauger et al. [61] stated the main issues that companies should consider before adopting DevOps in their study. The authors describe practical experiences in introducing DevOps practices. However, in this study, UTAUT is not the main underpinning foundation of the study framework. The study is limited to a single company (Fujitsu EST) and only quantitative research methodology (lack of results generalizability) [61].

Čižmešija and Stapić [62] presented the research results concerning the use of GitHub (one of the DevOps practices) in software engineering courses. They found that the UTAUT model needs to be supported by additional variables to understand better the students’ intention to use social coding platforms. Nevertheless, measuring instruments need to be improved through interviews with students to identify what should be essential in academic settings and the adoption of GitHub. In addition, the samples of the study were very homogenous. Therefore, the new measuring instrument developed needs to be checked and tested in other environments (such as IT development organizations) [62].

Anderson [63], in his research, examines the extent to which the UTAUT model with its independent variables is statistically related to the BI (dependent variable) to adopt and implement continuous delivery for the managers of software development projects in their software development organizations. He found that the project managers who participated in the study realized that continuous delivery (one of the DevOps practices) adoption would be helpful in their job, assist them in accomplishing tasks more quickly, and increase their organizational productivity. However, the project managers’ scope limitation might decrease the results’ generalizability. The study population was small compared with the total number of technical project managers (the findings may not be generalizable). Furthermore, the study utilized UTAUT without improving its variables; the analysis depends only on quantitative research methodology [63].

Masombuka [64] in his study, has developed a framework to adopt and implement a DevOps collaboration culture by determining significant factors that need to be encompassed in the developed framework. The author found that there must be a comprehension of the factors necessary to accept a DevOps collaboration culture in a software development organization. The author claimed that DevOps managers could use the study result to adopt DevOps’s collaboration culture successfully. However, UTAUT is not the main underpinning foundation of the study model, and the research framework did not factorize the managers of DevOps and their personalities. Furthermore, the study was implemented in a developing country context (South Africa); hence it may not represent the true reflection of other developing countries. Therefore, the study results may not be generalizable to all developing countries [64], [65].

2.3. Gaps in the literature

The literature so far firmly suggests the essential benefits of adopting DevOps in SDLC and how technology acceptance theories can play a prominent role in measuring innovations and technology adoption in different disciplines. However, related studies highlighted a lack of adoption of DevOps practices in software organizations, raising the question, “Why, despite the proven benefits of DevOps practices in the software development lifecycle, is its adoption still minimal for software development organizations in developing countries?” Moreover, many DevOps practices adoption issues within the software development field remain ambiguous, particularly the social and organizational aspects. Therefore, successful DevOps practices adoption that leads to enhanced performance of software organizations need to be further explored.

A model based on the enhancement of UTAUT with additional constructs has been conceptualized to fill this knowledge gap. An expected result of this research will be to consolidate perceptions of DevOps practices: an enhanced unified theory of acceptance and use of … (Ahmad Mahdi Salih)
practices adoption in software organizations to assist stakeholders in gaining the benefits of implementing these practices. This study proposed a model that can be used as a guideline to support successful DevOps practices adoption. This model will be an improved conceptual model because it will be designed by integrating additional variables with the well-known technology acceptance theory, the UTAUT. The developed model will assist in measuring software development practitioners’ perceptions of DevOps practices adoption [4].

3. RESEARCH METHOD

This study will develop a new measuring instrument and consist of scales related to the UTAUT quantitative model. These instruments are enhanced with new variables. The enhanced UTAUT model investigates software development practitioners’ acceptance of the DevOps software development approach. The following section presents the conceptual model development and the model hypothesis.

3.1. The conceptual model development

As stated in the “INTRODUCTION” section, provide a statement of what is expected. Technology acceptance theories, like TAM, TAM2, and UTAUT, have modeled how users accept and implement innovations and technologies. Numerous studies have revealed the successful use of these theories in explaining individuals’ perceptions of adopting new technologies. Therefore, those theories will be adopted and enhanced in this research to measure the influence of DevOps practices adoption on software development organizations [3]. Furthermore, we will introduce concepts of DevOps adoption in software organizations by integrating UTAUT with additional factors to explain software development practitioners’ perceptions regarding DevOps practices adoption.

The UTAUT model, as shown in Figure 2, includes variables that impact technology adoption. Those variables are PE, SI, EE, and FC, which directly and indirectly can influence BI to adopt and use the new technologies. UTAUT is based on a combination of eight original competing acceptance theories and came up with one comprehensive model, which made it a robust underpinning to explore different areas of technology acceptance and adoption topics. Furthermore, the relations among its constructs are moderated by gender, experience, age, and degree of voluntariness to use [4], [5].

![Figure 2. UTAUT model](image)

The successful integration of additional constructs with UTAUT will assist us in understanding how the adoption of DevOps in software organizations can be achieved successfully. Literature has shown the successful applicability of UTAUT in the different disciplines of the IT fields [6]. However, there is a lack of exploring these concepts in the context of software organizations. One of the goals of this study is to enhance the perceptions of DevOps adoption in software organizations to assist stakeholders in benefiting from implementing DevOps practices [7], [8].
Based on the considerations mentioned above, we formulated the following questions: i) RQ1: What are the key factors influencing the intention to adopt DevOps practices in the software organizations context? and ii) RQ2: How will the enhanced UTAUT predict the intention to adopt DevOps practices in software organizations? The proposed model will explore those questions explained in the following section.

The conceptual proposed model of this research is shown in Figure 3. In addition to the original constructs of UTAUT: PE, SI, EE, and FC, the model includes (perceived DevOps practices (PDP), perceived feasibility (PF), perceived organizational usefulness (POU), and training) as new factors that may affect the intention to adopt DevOps practices. The PDP, PF, POU, and training will be vital factors in DevOps practices adoption and will be included in this research as researchers recommended in the software industry environment [8], [9].

The comprehensive literature review was the main component in conceptualizing the proposed model. Synthesis, criticism, and comparison techniques were used to design the proposed model. The variables’ measurement items are appropriately modified to be consistent with DevOps software organizations’ context [5]. Table 2 demonstrates the sources of the proposed model’s factors, and each factor (construct) has its scales (items) used to measure the concerning factor, as described in Table 3.

![Figure 3. The proposed model](image)

Table 2. Factors definition of the proposed model

<table>
<thead>
<tr>
<th>Factor/construct</th>
<th>Definition</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance expectancy (PE)</td>
<td>The degree to which using technology will benefit employees when performing certain activities.</td>
<td>[40]</td>
</tr>
<tr>
<td>Effort expectancy (EE)</td>
<td>The degree of ease associated with employees’ use of technology.</td>
<td>[40]</td>
</tr>
<tr>
<td>Social influence (SI)</td>
<td>The degree to which the individual believes that persons important to him/her think that he/she should use or refrain from adopting the new system/technology.</td>
<td>[40]</td>
</tr>
<tr>
<td>Facilitating conditions (FC)</td>
<td>The degree to which an individual is convinced that technical and organizational infrastructure are presently available to support the new system/technology use.</td>
<td>[40]</td>
</tr>
<tr>
<td>Perceived DevOps Practices (PDP)</td>
<td>PDP describes how practitioners comprehend DevOps practices’ significance in improving software development processes.</td>
<td>[10]</td>
</tr>
<tr>
<td>Perceived feasibility (PF)</td>
<td>PF defines software development practitioners’ perception of their knowledge, experiences, and abilities to adapt to DevOps practices.</td>
<td>[11], [12]</td>
</tr>
<tr>
<td>Perceived Organizational Usefulness (POU)</td>
<td>POU is defined as evaluating the effectiveness of adopting a system with its fundamental elements to an organization.</td>
<td>[13]</td>
</tr>
<tr>
<td>Training</td>
<td>Training provides management and employees with knowledge and information about the newly adopted systems.</td>
<td>[2]</td>
</tr>
</tbody>
</table>
3.2. Development of the model hypotheses

The conceptualization of the proposed model has been done to consolidate factors influencing the intention to adopt DevOps practices for software development organizations. The UTAUT original constructs are integrated with the new constructs to consolidate the developed model. It depends on factors (constructs) that their items will be explained in the following sections to derive the hypotheses of the research.

3.2.1. The constructs of UTAUT

Performance expectancy (PE) is defined as “the degree to which using technology will benefit employees when performing certain activities” [4], [5]. Regarding this research, PE refers to how software development practitioners believe that DevOps practices will assist them in attaining gains and increase
opportunities, productivity, and achievements in their software development processes. Based on these studies, the relationship posited by the UTAUT model assumes a positive influence of PE on employees’ BI to adopt new technology [6], [9], [14], [15], [66]; hence, this study proposes the following hypothesis to be tested:

**H1: Performance expectancy has a positive effect on the intention to adopt DevOps practices in software development organizations.**

Effort expectancy (EE) as defined by Venkatesh et al. [4] is “the degree of ease associated with employees’ use of technology”. In light of this research, EE indicates how easy it is for software development practitioners to adopt DevOps practices, precisely the difficulty or complexity of adopting DevOps practices, and the efforts devoted to this adoption [16], [17]. Based on the literature, EE can be considered a crucial predictor of behavioral intention, as reported, for instance by research [6], [18]–[22]. EE affects the attitude directly according to Abdou and Jasimuddin [50]. Hence, this research proposes the following hypothesis to be tested:

**H2: Effort expectancy has a positive effect on the intention to adopt DevOps practices in software development organizations.**

Social influence (SI) is stated by Venkatesh et al. [4] as “the degree to which the individual believes that persons important to think him/her that he/she should use or refrain from adopting the new system/technology”. Regarding this study, SI describes how software development practitioners perceive that important person influence their adoption of DevOps practices. These persons are mainly represented by stakeholders engaged in all stages of software development activities, such as policymakers, managers, heads of departments, and colleagues [14], [23], [24]. Some studies have revealed no significant effect of SI on user acceptance, such as by Mohamadali and Azizah [28]. However, many studies have reported that SI can be considered a significant predictor of behavioral intention, for example by Ifinedo [30], Yurdakul et al. [22], Dečman [31], Sharma et al. [32], Farooq et al. [33], Halliti and Sulaiman [34], Durak [35], and Zhang et al. [36].

Studies in the field of IS showed the significance of SI in adopting and using new technologies. In the earlier adoption phases, if the user has insufficient or no experience in new technology, his/her behavior may affect by others’ important opinions [34]. Based on the findings mentioned above, this research proposes that SI positively affects BI toward adopting DevOps practices, so the following hypothesis is to be tested:

**H3: Social influence has a positive effect on the intention to adopt DevOps practices in software development organizations.**

Facilitating conditions (FC) as defined by Venkatesh et al. [4] is “the degree to which an individual is convinced that technical and organizational infrastructure is presently available to support the new system/technology use”. In this research, FC is defined as the perception of software development practitioners of the support and resources available to utilize DevOps practices, including organizational and technological elements planned to eliminate the barriers to adopting DevOps practices [35], [36]. FC directly influences technology use but not behavioral intention to use technology [4]. However, many studies have shown a positive influence of FC construct on BI, for instance by research [15], [22], [37]–[43]. Worth mentioning, Nuq and Aubert [47] revealed that FC should be considered when examining BI towards technology use in developing countries’ cases due to fewer resources. Nuq and Aubert’s study has been supported by the results of the study of Alam et al. [46]. FC can have an expected positive influence on the practitioners of software development intention to adopt DevOps practices; hence, the following hypothesis is proposed to be tested:

**H4: Facilitating conditions have a positive effect on the intention to adopt DevOps practice in software development organizations.**

3.2.2. The non-UTAUT constructs

Four critical factors identified in the literature can significantly influence behavioral intention to adopt and use DevOps practices [2], [10], [12], [45], [46]. Those factors will be integrated with original UTAUT constructs as new factors that may impact DevOps adoption and be vital in DevOps practices adoption as recommended by researchers in software development organizations [2], [9], [47]. Those factors are explained as follows:

Perceived DevOps practices (PDP) describes how practitioners comprehend DevOps practices’ significance in improving software development processes. Many studies have shown how the DevOps approach’s practices improve the software development processes in the literature. Humble and Molesky [67] presented the advantages of introducing unit tests early and make pedagogical recommendations for the introduction, and use it in first year programming, Dijkstra [68] provided an insight in how automation can improve testability and scalability while simultaneously decreasing the operators’ manual work, Cois et al. [69] had shown that students equipped with the DevOps practices could learn and solve the programming problems by themselves, and that improve their software development experiences when they become industry practitioners, Lwakatare et al. [70] found that there are significant educational benefits of introducing a course in cloud computing combined with DevOps practices and capabilities, one of these...
benefits is learning to produce innovative services with higher quality at lower cost, Burrell [51] in his study, has shown the adoption of DevOps practices and the resulted potential impact on the agility of the organization stand to revolutionize how highly effective firms are being run [51], and Rafi et al. [58] have illustrated that integrating DevOps concepts and practices into the universities curricula can present the operations to students and improve practicing and strengthen the students’ knowledge of software development, hence, that will prepare them to become a successful DevOps practitioners.

Regarding the literature discussed above, this research hypothesizes that if the software development practitioners perceive the significance of the DevOps approach’s practices in improving software development processes, they will be more potentially to adopt the DevOps approach. So, the following hypothesis proposes to be tested:

\[ H5: \text{Perceived DevOps practices (PDP) have a positive effect on the intention to adopt the DevOps practices in software development organizations.} \]

Perceived feasibility (PF) is adapted from Bandura [11], as cited by Moghavvemi et al. [12]. Bandura argues that “taking action requires consideration of not just perceived desirability but also perceived feasibility” [11]. Perceived feasibility represents the perception of an individual capability (personal capabilities) to achieve a specific job or group of tasks [11], [12]. It concerns the ability of individuals and their judgment on their capabilities to use innovations [49], [50].

Perceived feasibility within this study defines by the practitioners of software development’s perception of their knowledge, experiences, and abilities to adapt to the DevOps approach [51]–[53]. A higher level of PF for the practitioners may lead them to have more intention to adopt DevOps practices [12], [51], [52]. Suppose the practitioners perceive that they have enough abilities and skills to adopt DevOps practices in their organizations. In that case, they will be more interested in utilizing it in their software development processes [53], [54].

Many studies have shown the significant effect of perceived feasibility on behavior intention in literature. Coduras et al. [59] illustrated in their study that credibility needs the behavior would be both desirable and feasible. Those antecedents affect the intentions of the behavior of new technology [59], Devonish et al. [71] and Dissanayake [72] in their researches showed that perceived feasibility and desirability perceptions had been argued to be instrumental in the promotion of positive entrepreneurial intentions, particularly with student populations [71], [72]. Moghavvemi et al. [73] proved in those studies that the effect of PF on behavior intention was significant [73]. In their study, Lin et al. [74] mentioned the effects of entrepreneurship education on perceptions of feasibility and desirability as a positive career of choice and increased students’ self-confidence [74].

Based on the supporting literature discussed above, a higher level of perceived feasibility for software development practitioners will lead them to have a higher intention to adopt DevOps practices. Hence, the following hypothesis proposes to be tested:

\[ H6: \text{Perceived feasibility has a positive effect on the intention to adopt DevOps practices in the software development organization.} \]

Perceived organizational usefulness (POU) is defined as evaluating the effectiveness of adopting a system with its fundamental elements to an organization. The organization’s developers do this evaluation. Organizational usefulness directly influences developers’ intentions to adopt the approach [8], [13]. This study defines organizational usefulness as the software development organization’s perceived usefulness from adopting the DevOps practices evaluated by the software development practitioners [73], [75]. In the literature, few types of research showed that organizational usefulness is not a significant predictor, such as the study conducted by [74]. However, many studies have shown that OU can be a significant predictor. In their research, Then and Amaria [76] illustrated that higher education institutions must evaluate the organization’s usefulness by adopting emerging IT technologies by their staff to develop the institution [76], [77]. Tregeagle [78] presented that OU was a vital factor in her study. Over time, OU can boost the guided practice system used in the agency’s functioning [78]. Globisch et al. [79] have found that the variables at the organizational level, in particular, “perceived organizational usefulness”, be significant to the organization members’ reactions to accepting and adopting electric vehicles system (EVs) and supporting the acquisition of EVs systems in other organizations [79]. Bamgbade et al. [80], in their research, have shown perceived organizational usefulness as one of the essential factors that can predict the intention to adopt sustainable construction technology, which influences the sustainability performance of construction firms [80]. Sok sophay and Duang-Ek-Anong [81] showed that OU directly influences the developers’ and practitioners’ intention to use the DevOps approach in a technology industry environment, mainly software development.

Based on the literature discussed above, if the perceived organizational usefulness is found to be at a high level by adopting DevOps practices, then this will encourage the software development practitioners to have a higher intention to adopt the DevOps software development approach. So, the following hypothesis proposes to be tested:
H7: Perceived organizational usefulness has a positive effect on the intention to adopt DevOps practices in software development organizations.

Training refers to providing knowledge and information for management and employees on the newly adopted systems. Training gives a better comprehension of how the jobs relate to functional organizational areas [2], [79]. Training is also a concern as successful system adoption requires training the stakeholders through in-service programs, workshops, and seminars on how to handle the system intended to adopt [2], [46], and thus, this research will include training as an independent variable to be examined.

Training practitioners ensure lessening risks and problems that may crop up to preclude the successful adoption and implementation of the system [80], [81]. Lack of proper and sufficient training may hamper the adoption of DevOps practices among software development practitioners [82]. Moreover, lack of training may increase the practitioners’ ambiguity and unfamiliarity with DevOps practices implementation, leading to failure of DevOps implementation [83]. DevOps practices training that breaks down silo mentality, encourages collaboration and improves communication should be the preparatory point of every software development organization [75], [84].

In literature, studies regarding the variable of training have shown that it positively affects the adoption and implementation of new systems and approaches, and this was demonstrated through the results reported by research [2], [39], [80], [82], [85], [86]. Therefore, proper training sessions will be required to comprehend the DevOps concepts and practices properly. Software organizations must support their practitioners with training sessions to help their organizations adopt the DevOps approach successfully [52], [87]. Hence, the following hypothesis needs to be tested:

H8: Training has a positive effect on the intention to adopt DevOps practices in software development organizations.

DevOps adoption and perceived productivity. It is believed that when software development practitioners collaborate in the DevOps environment, their organization’s productivity will significantly be improved [88]. Therefore, implementing DevOps practices effectively and efficiently within a software development organization will potentially bring about the production’s success [87], [89].

This study will consider the identified factors of the proposed model as the behavioral intention’s determinants and examine the relationship between the BI to adopt the DevOps practices and the perceived productivity of software development organizations [82]. Based on the consistency of UTAUT performance, it can be recommended to examine practitioners’ intention towards adopting the DevOps practices and their organizations’ productivity and performance regarding software development processes [46], [58], [70], [90].

In their qualitative study, Silva et al. [91] analyzed if there were productivity could be gained after transitioning to DevOps practices. The authors found that the team’s overall time regarding the operational work has been reduced from 50% to 26%. Furthermore, they also found a reduction in defects and a growth in development capacity. This can be considered a positive trend of shifting capacity from defect solving to developing new features, which increase delivery quality, customer satisfaction, and a product with low bugs [91]. Ali et al. [92] presented a systematic reused-based software development and management process with a hybrid DevOps process to reduce the cost and effort essential for increasing productivity. The study results also revealed that the proposed process got a 35.2% average gain in developed function points [92]. In his study, Mubarkoot [7] assessed the factors that impact DevOps practices adoption in the public sector. After evaluating the factors, he found that DevOps practices enhance organizational productivity and performance.

Regarding this study, we define behavioral intention (BI) to adopt DevOps practices as the level to which software development practitioners intend to utilize DevOps practices to perform their software development tasks. For specific approaches and technologies, BI is frequently measured compared to its actual usage [63], [91]. The UTAUT theory supports the influence of BI on system use (the variable that represents the perceived productivity of this study). The proposed model can be considered quite robust and extended to comprise valuable constructs to provide insight into technology adoption (DevOps approach in the current study) [16], [92]. Based on the discussion above, this study proposes the following hypothesis to be tested:

H9: Adoption of DevOps practices will positively affect the perceived productivity of software development organizations.

3.2.3. The moderators of the proposed model

In the improved UTAUT model of this study, some moderators can change the effect of factors on behavioral intention to adopt DevOps practices [70], [93]. This study will not incorporate the moderators of age and voluntariness of use because they may have little or no effect on behavioral intention in studies associated with adopting DevOps practices [94]–[96]. The other two moderators, namely gender and years of experience as a software developer, and their hypotheses will be determined for examining the moderating effects of the variables PE, EE, SI, and FC, respectively.
Gender: in the literature, the existing models regarding the adoption of technology have shown that demographic factors are essential components in the intention to adopt and implement technology [97]–[99]. Previous studies have shown that gender is essential to adopt and using new technologies [4], [100]. Venkatesh et al. [4] showed that the explanatory power in the technology adoption model increased to 52% when gender was a moderator. Consequently, gender can moderate the effect of acceptance and technology use. Furthermore, gender has a psychological effect on the acceptance process. Biljon and Kotzé [101] has shown that women are less than men concerned with the “benefit of technology”. Therefore, gender is an essential variable in studies exploring technology adoption behavior. Based on the original UTAUT model [4], supported by many studies from the literature on the effect of gender as a moderator on the variables PE, EE, and SI [99], [102]–[104], the following hypotheses propose to be tested:

**H10**: Gender is significantly moderating the relationship between PE and intention to adopt DevOps practices in software development organizations.

**H11**: Gender is significantly moderating the relationship between EE and intention to adopt DevOps practices in software development organizations.

**H12**: Gender is significantly moderating the relationship between SI and intention to adopt DevOps practices in software development organizations.

Years of experience: experience is essential since it plays a significant role in accepting the effort made at the beginning of gaining a new behavior [4], [105], [106]. Previous studies have found that a user’s experience with technologies affects the relationship among variables regarding new technology acceptance [86], [107], [108]. The literature synthesis has shown the effect of experience as a moderator on technology acceptance and use. In research [38], [109]–[112] found that variables related to behavioral intention toward using new technologies were significantly affected by the moderator’s experience. The years of experience moderator can vary from study to study. In this study, the years of experience moderator will represent the years of experience of the practitioners in software development in the context of software development organizations. Based on Venkatesh et al. [4], the following hypotheses propose to be tested:

**H13**: Years of experience significantly moderate the relationship between EE and the intention to adopt DevOps practices in software development organizations.

**H14**: Years of experience significantly moderate the relationship between SI and intention to adopt DevOps practices in software development organizations.

**H15**: Years of experience significantly moderate the relationship between FC and the intention to adopt DevOps practices in software development organizations.

Fifteen hypotheses have been developed in this research. They represent the relationships among the constructs of the developed model to address the study’s research questions. These hypotheses are illustrated in Table 4. Figure 4 shows the study’s conceptual model and the hypotheses that reflect the relationships among constructs of the conceptual model.

### Table 4. The research hypotheses

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>Performance expectancy has a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H2</td>
<td>Effort expectancy has a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H3</td>
<td>Social influence has a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H4</td>
<td>Facilitating conditions have a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H5</td>
<td>Perceived DevOps practices have a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H6</td>
<td>Perceived feasibility has a positive effect on the intention to adopt DevOps practices in the software development organization.</td>
</tr>
<tr>
<td>H7</td>
<td>Perceived organizational usefulness has a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H8</td>
<td>Training has a positive effect on the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H9</td>
<td>Adoption of DevOps practices will have a significant relationship with the perceived productivity of software development organizations.</td>
</tr>
<tr>
<td>H10</td>
<td>Gender is significantly moderating the relationship between PE and intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H11</td>
<td>Gender is significantly moderating the relationship between EE and intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H12</td>
<td>Gender is significantly moderating the relationship between SI and intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H13</td>
<td>Years of experience significantly moderate the relationship between EE and the intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H14</td>
<td>Years of experience significantly moderate the relationship between SI and intention to adopt DevOps practices in software development organizations.</td>
</tr>
<tr>
<td>H15</td>
<td>Years of experience significantly moderate the relationship between FC and intention to adopt DevOps practices in software development organizations.</td>
</tr>
</tbody>
</table>
Adopting DevOps practices: an enhanced unified theory of acceptance and use of ... (Ahmad Mahdi Salih)

4. CONCLUSION AND FUTURE WORK

This research has developed a conceptual DevOps practices acceptance framework based on the UTAUT model. The developed model identifies the key factors that influence DevOps practices adoption in software development organizations. In addition to the original UTAUT constructs, performance expectancy, effort expectancy, social influence, and facilitating conditions, the proposed model was enhanced by adding four variables; perceived DevOps practices, perceived feasibility, perceived organizational usefulness, and training. The proposed model and the research hypotheses will be checked for validity and reliability by conducting an extensive questionnaire survey that targets software development practitioners and experts. This will assist in shaping the final developed model to understand better the adoption of DevOps practices in software development organizations. The resulting developed model of this research is expected to enhance our understanding of software development practitioners’ acceptance and adoption of the DevOps approach. Future works will include surveys and expert interviews to enhance model validation and generalizability.

REFERENCES


Int J Elect & Comp Eng
ISSN: 2088-8708


Adopting DevOps practices: an enhanced unified theory of acceptance and use of ... (Ahmad Mahdi Salish)
BIOGRAPHIES OF AUTHORS

Ahmad Mahdi Salih is a lecturer at Mathematics Department, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq. He received a bachelor’s degree in computer engineering from Northern Technical University, Iraq, and a master’s degree in IT from the Faculty of Computing, Universiti Utara Malaysia. Currently, he is a Ph.D. student at the School of Computer Sciences, Universiti Sains Malaysia. Ahmad is specialized in computer engineering and software engineering. His research interests include software engineering, software requirements, and software development approaches. He can be contacted at email: ahmad_mahdi@student.usm.my.

Sharifah Mashita Syed-Mohamad is a senior lecturer at the Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu. She was awarded a Ph.D. in Service System Engineering (SSE), Universiti Sains Malaysia, Malaysia. Dr. Sharifa specialized in software reliability, testing, metrics, and measurement. Her research interests are software reliability, testing, verification and validation, software metrics and measurement, iterative and incremental development, empirical software engineering, requirements engineering, and software maintenance and evolution. She can be contacted at email: s.mashita@umt.edu.my.

Pantea Keikhosrokiani is a senior lecturer at the School of Computer Sciences, Universiti Sains Malaysia. She was awarded a Ph.D. in Service System Engineering (SSE), Universiti Sains Malaysia, Malaysia. Dr. Pantea specialized in information systems and behavioral analytics, service system engineering, technopreneurship, business intelligence. Her research interests are information systems and analytics, business intelligence, behavior-change support systems, big data analytics for information systems, behavioural and medical analytics, health, medical and business informatics, database management systems, text analytics and opinion mining, and digital technopreneurship. She can be contacted at email: pantea@usm.my.

Nur Hana Samsudin is a senior lecturer at the School of Computer Sciences, Universiti Sains Malaysia. She was awarded a Ph.D. in Computer Science from the University of Birmingham, UK. Dr. Nur Hana specializes in natural language processing, speech processing, and computational linguistics. Her research interest is best categorized in speech synthesis and speech processing domain. It also belonged to the natural language processing, human-computer interaction, and multilingual speech processing categories. She is also interested in voice banking, speech recognition, and speech understanding research. She can be contacted by email: nurhana.samsudin@usm.my.