

## Implications of Onshore Development on Global Software Engineering

Abdulrahman M. Qahtani<sup>1</sup> and Ahmed S. Ghiduk<sup>1,2,\*</sup>

<sup>1</sup>College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif, 21944, Saudi Arabia

<sup>2</sup>Department of Mathematics and Computer Science, Faculty of Science, Beni-Suef University, Beni-Suef, 62521, Egypt

\*Corresponding Author: Ahmed S. Ghiduk. Email: asaghiduk@tu.edu.sa

Received: 31 May 2022; Accepted: 12 July 2022

**Abstract:** Recently software industry has paid significant attention to customizing software products across distributed boundaries. Communicating the requirements of multiple clients across distributed borders is a crucial challenge for the software customization process. Local decision-making and local development at the client site are considered methods for reducing difficulties in communicating the requirements of multiple clients across distributed boundaries. This paper introduces a new model called the onshore development model (ODM) for accomplishing the customization requests in the distributed development process of software. This model presents a scenario for enhancing the onsite development of specific requirements to reduce delays and misunderstandings between the clients and the team involved. This model depends on moving the development process to the client's location. Three empirical studies were conducted to evaluate the proposed model to measure its productivity, time performance, and cost reduction. The proposed model has been compared with two other models: the basic model (BM), which allocates the decision-making process and the development process for teams at the vendor's location, and the local decision-making model (LDec), which assigns the decision-making process for team at the client's location. The results of the empirical studies showed significant outperforming of the proposed model over the basic model and local decision-making model in productivity, time performance, and cost reduction. The productivity of the proposed model improved by 39% and 10% more than the basic model and the local decision-making model, respectively. In addition, the time performance of the proposed model became faster by 49% and 20.8% than the basic model and the local decision-making model, respectively. Also, it reduced the total cost of the development process by 31% in terms of the salaries of all persons involved in requirements collecting, decision-making, and development.

**Keywords:** Distributed development; local development; software customization



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

The software industry has turned to the decentralizing style for performing the software engineering processes in a distributed manner around the world called Distributed Software Development (DSD). Consequently, software development companies decided to apply the global software development model (GSD) [1], which distributes both software development and maintenance on multiple sites worldwide. DSD and GSD have received much attention owing to the importance of increasing distributed development and outsourcing projects [2,3]. The main motive for this trend is the need to allocate development teams to different distributed locations to achieve higher quality products and services at lower cost [4] and look for competitive advantages in terms of quality, cost, and human resources [5]. The transition from co-located to distributed projects has been accompanied by significant changes to aspects of the software development field [6], including features and challenges integral to software development project practices [3].

Managing the communication of client requirements is a vital component of the development production process in the marketplace. While the local negotiation and communication of requirements are considerably hard, this process is challenging through distributed sites, especially for numerous clients. The hardness of this obstacle grows in DSD projects with multiple clients spread through organizational and distributed boundaries. Lately, there has been a significant change in forms of collocated development to GSD that demands extra communication through administrative boundaries [7].

The software development organizations adopted two manners to produce software [8]. The vendors can develop the software from scratch according to the clients' requirements or from the 'off-the-shelf' software. Many studies have referenced the value of customization in the software industry [9].

The current work explores the customization of software products through organizational and distributed sites to cope with the problem of communicating the requirements in DSD. The main contribution of this work is studying the research problem: does the development of the secondary requirements at the client site improve the customization process by reducing the cost and increasing productivity and performance? Customizing some needs at the client site contributes to a pivotal work in shortening the development task in the customization of software products. This work investigates the effect of local development on the productivity, time, and cost of the customization process in the distributed multiple client scope. This work will answer the research questions by emphasizing the following hypothesis. This hypothesis claims that locally implementing development activities on a few requirements could reduce the customization process's required time and cost and increase productivity.

To achieve the objective of this work, a new model for accomplishing the customization process locally in the distributed development process of software has been proposed. This model introduces a technique for enhancing the onsite development of specific requirements to reduce delays and misunderstandings between the clients and the teams involved. This model depends on moving the development process to the client's location. Three empirical studies were conducted to evaluate the proposed model depending on three criteria productivity, time performance, and cost reduction. The proposed model has been compared with two models: the basic model, which allocates the decision-making process and the development process to persons at the vendor's location, and the local decision-making model, which assigns the decision-making process to persons at the client's location. The empirical studies showed that the proposed model outperformed the basic and local decision-making models in productivity, time performance, and cost reduction. The proposed model improved

productivity by 39% and 10% more than the basic and local decision-making models. In addition, the proposed model enhanced the time performance to be faster by 49% and 20.8% than the basic and local decision-making models, respectively. Also, it reduced the total cost of the development process by 31% in terms of the salaries of all persons involved in requirements collecting, decision-making, and development.

The rest of this research is structured as follows. Section 2 introduces the related work. Section 3 presents the concepts of distributed software development. Section 4 introduces the details of the proposed model. Section 5 presents the empirical evaluation of the proposed model. Section 6 concludes the outcomes of this paper.

## 2 Related Work

Before the 1980s, while the DSD idea appeared [1], the industry depended on the classical in-house software development method [10]. Significant awareness has driven DSD from the 1990s to now [11]. Software projects are developed in various models. Software products can be created in a collocated form when the development team allocates only one site [12]. Also, software products can be developed in a geographically distributed form, where the stakeholders of the product physically exist in different locations [13].

Some DSD projects include several teams in only one city. In contrast, other projects contain groups distributed through different countries [14] to get the benefits of quality, cost, and persons' skills [5].

Shifting the software production from co-located to distributed development came through significant modifications to the software development activities and features which causes challenges to software development projects in practice [3,6].

Many researchers have studied the DSD idea from various perspectives according to the software development context, including development organizations [15,16], project management [17], and project allocation decisions [18]. Scholars investigated the interrelationship between vendors and clients and the effect of DSD on team members, including cultural differences and language [10]. Researchers reported many GSD concerns, including communication, collaboration, time zones, cultural and organizational boundaries, and managing requirements.

Jiménez et al. [3] reviewed the problems and challenges of DSD (e.g., communication, interrelationships between stakeholders, configuration and knowledge management, cultural differences, and quality measurement) and the possible solutions for these challenges. In addition, Sengupta et al. [1] surveyed challenges such as collaborative tools, knowledge acquisition and management, and testing in a distributed. They handled the difficulties in each field (e.g., inadequate communication, trust, system integration, and knowledge management).

Different DSD aspects were examined, such as multi-site development. Many DSD publications have studied the effect of multiple site projects on the development activities (e.g., the work presented by Damian et al. [19]). Damian et al. [19] recognized common issues such as poor communication, knowledge management, cultural difference, and time variance. In addition, Damian [7] studied global software engineering (GSE) problems and issues. He clarified that there are many issues in managing the requirements of distributed stakeholders in the DSD scope because of different cultural, time zone, and organizational borders. In addition, he explained that communication and coordination become harder in multiple site projects across administrative and cultural borders than in single site projects.

The agile development model has appeared in response to conventional models (e.g., the waterfall model) to decrease development time and production cost and implement changes quickly without affecting the total development time [20–22].

Husin et al. [23] and DeFranco et al. [24] studied the risk management of teamwork in DSD. The outcomes showed that communication is the key issue noted in the collaboration. L'Erario et al. [25] introduced a framework for communication in DSD.

Damato et al. [26] surveyed many tools that help in DSD. Jolak et al. [27] studied the influences of distance on distributed software design, and they found that distance still represents an issue for distributed software development.

Ghiduk et al. [28] investigated the efficiency of making decisions locally on the customization process in DSD according to productivity and cost reduction. The results showed the efficiency of local decision-making in increasing productivity and reducing cost.

### 3 Background

This section introduces the concepts required to understand the rest of this work.

#### 3.1 *Distributed Software Development*

Distributed software projects might depend on their stakeholders' geographical places [29]. DSD projects assign team persons to several sites over the development life cycle. The space among persons can be some meters in the same building or many kilometers across countries called global software development (GSD) [3]. Software development processes in GSD can be done in two different models (offshore and onshore) [14].

**Offshore:** Offshoring is a software development paradigm in which the company of the client and the company of the vendor exist in different countries [30]. This development pattern possesses some issues, which desire examination, for example, the time zone issue [31], distinct cultures [19], and problems of development through organizational and distributed boundaries [32].

**Onshore:** Onshore is the second paradigm of distributed development projects in which all clients and vendors exist in one country [14]. This paradigm faces fewer issues than the offshore paradigm, where culture, language, and time zones are the same among stakeholders of the onshore development paradigm. However, projects still suffer communication, collaboration, and management problems, particularly for multi-teams in diverse countries [1]. Onshore development has two techniques according to the place of the team. Firstly, the development activity carries out in the same country but in several cities or areas [1]; secondly, it carries out at one site [29]. It is hard to satisfy these requirements locally and communicate them over multi-sites [32]. This hardness scale becomes greater in DSD projects and projects in which many stakeholders are located in several organizations.

#### 3.2 *Basic Model of the Customization Process*

Many customers desire re-configuration for software to meet their requirements. Therefore, developers look forward to customizing the released software to satisfy the changes ordered by these customers [33].

The overall structure of the customization process in the DSD is as follows. The customization process contains three phases. Firstly, the vendor develops and deploys the software. Secondly, customers declare the required changes to be delivered to the vendor. Then, the vendor contacts

the customers to negotiate and decide (e.g., cancellation, rejection, and acceptance) on the demand changes. Thirdly, the vendor allocates customization teams to customize the software according to the required changes. The main issues of customization (e.g., decision-making, communicating, and negotiating with the client) occur in the third step. The vendor must handle these issues to reduce the customization duration. Fig. 1 shows the basic model (BM) of the customization process. This model allocates all customization processes such as decision-making and development to persons at the vendor's location, except collecting the requirements for a team at the client's location.

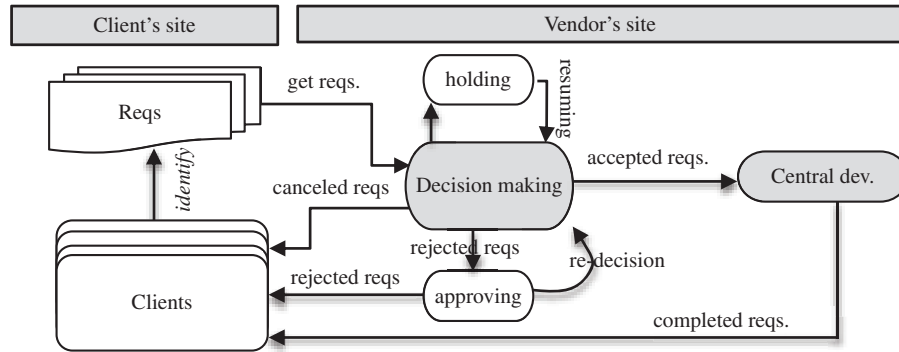


Figure 1: Basic model (BM) for customization process

### 3.3 The Local Decision-Making Model (LDec)

The local decision-making [33] model targets improving local decision-making in DSD to cope with the issues of communicating the requirements of clients over multi-sites [33]. The activities of the LDec model start with requirements analysis. Subsequently, the process of collecting the customization requirements is carried out at the customer's site, and the development activities containing design, implementation, testing, and evolution happen at the site of the offshore vendor. The decision-maker team moves to the client's location to reduce the communications issue, which is the main benefit of LDec. Fig. 2 shows the main activities of the customization process's local decision-making (LDec) model. This model allocates the identifying requirements and decision-making to persons at the client's location and the development process at the vendor's location.

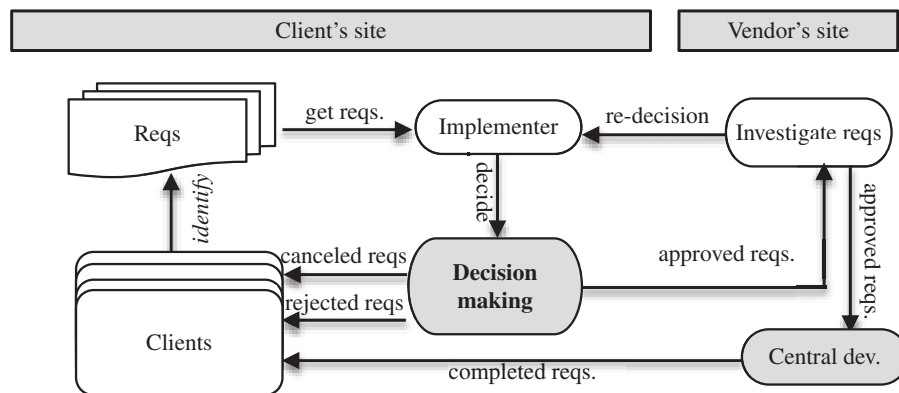
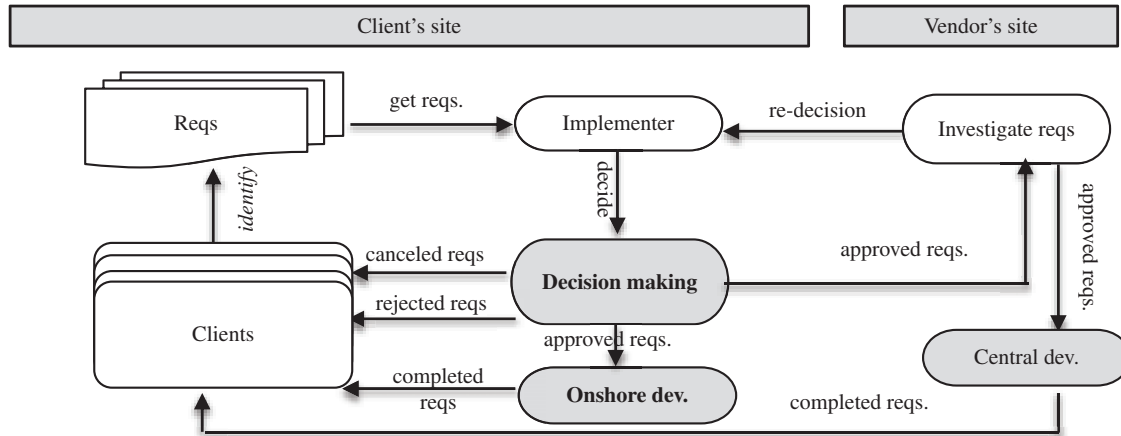


Figure 2: Local decision-making model (LDec) for customization process

#### 4 The Proposed Onshore Development Model (ODM)

This section presents the proposed onshore development model (ODM). The ODM model introduced in Fig. 3 divides the development process between onsite development carried out at the client's site and offsite development carried out at the vendor's site.



**Figure 3:** The onshore development model (ODM)

In ODM, the customization activities cycle begins with collecting clients' required requirements. The customization requirements might be bugs in the software that need to be debugged or new features that need to be added. These bugs are always issues in the software's released version, which requires examination to find the problem and debug it. At all times, the novel features demand analysis and development. Each of the two requirements needs mutual communication between the client and software vendor to understand and record the requirements. In ideal offshore projects, a small team representing the software vendor exists at the clients' site to contact them to understand and collect their needs [34]. The second stage is negotiating with the client to decide, which can be cancellation, rejection, or acceptance. The decision-making takes place at the client's site with an onsite development team for investing the benefits of direct communication and negotiation.

Subsequently, the requirements are allocated to the onsite team to be implemented and delivered to the client or passed to the offsite team at the vendor's site. The requirements given across a checkpoint at the vendor's site for the offsite customization team prove that the requirements are understood and scheduled as tasks for the developers. If the bugs are not realized or the needs for new features are ambiguous, these requests are returned to the client's site for extra investigation and re-decision. The last task in the customization activities is the development process, containing implementing, testing, verifying, and delivering the customized software. ODM aims to decrease the side effects of a distributed customization process, such as development time. Therefore, the ODM performs the development activities on a few requirements locally to reduce the required time for the customization process and, consequently, the period of releasing the customized product.

Tab. 1 shows the allocation of customization activities such as identifying the requirements, decision-making, and development at the client and vendor locations in the BM, LDec, and ODM models.

**Table 1:** The allocation of customization activities at the client and vendor locations

Model	Client's site	Vendor's site
BM	Identifying the requirements	Decision making Entire development
LDec	Identifying the requirements, and Decision making	Entire development
ODM	Identifying the requirements, Decision making, and Partial development	Partial development

## 5 Empirical Evaluation ODM

This section concentrates on the actions and procedures taken to answer the research questions and inspect the research hypotheses. Besides, it details the experimental studies that were conducted to estimate the efficiency of local development on the software customization activity in the DSD against the quality of processing requirements and reducing time. The experimental studies apply the principles suggested by Perry et al. [35].

This research uses the simulation method for the empirical evaluation of the ODM model using a chosen case study to answer the research questions. The used method imitates the customization model of the chosen case study to be a basic-state model in the experimental valuation for the ODM model to validate the suggested hypotheses. The details of the experimental evaluation and the obtained results are given below.

### 5.1 Problem Definition

As discussed, the customization activity, which depends on multi-client, faces many issues in communicating with offshore developers to understand and develop the demanded requirement [36].

### 5.2 Problem Review

The LDec model is introduced to overcome the issues in communication during the customization activity [33]. The objectives of LDec are reducing the time of the customization process [28,33], increasing the number of processed requests [28], and reducing the decision-making cost [28].

The main contribution of this paper is proposing a new model (ODM) for the local development of the customization process in DSD and estimating its effectiveness in terms of the ability to reduce the cost and time of the customization process and the entire duration of software development and increase the number of processed requests.

### 5.3 The Research Questions and Hypotheses

This work gauges the effects of local development of some requests in the customization activity by evaluating the quality of ODM to increase the number of processed requests and reduce the time cost of the customization process. The following research questions are defined to achieve these objectives:

**RQ1:** Does developing the secondary requirements at the client site reduce the customization time?

**RQ2:** Does the development of the secondary requirements at the client site increase the productivity of the customization process?

**RQ3:** Does developing the secondary requirements at the client site reduce the customization cost?

The study hypothesizes that the DSD localization conceptions of customizing software accelerate the customization operation. Therefore, the ODM model can decrease the required time and increase the number of implemented requests. The hypotheses can be defined as follows.

**H0:** The local implementation of development activities on a few requirements couldn't reduce the required time and cost for the customization process and increase the number of processed requests.

**H1:** The local implementation of development activities on a few requirements could significantly reduce the required time and cost for the customization process and increase the number of processed requests.

#### **5.4 Measurements**

To examine the hypotheses and solve the research questions of this work, we will assess the extent of the ODM model in reducing the time and cost of the customization process and the entire duration of software development and increasing the number of processed requests. Therefore, three experiments were carried out: the first experiment estimates the number of completed requirements, the second evaluates the time of the customization process, and the third estimates the cost in salaries of all persons involved in the activities of the customization process.

#### **5.5 Experimental Tool**

Simulation has been utilized to assess the performance of the ODM model. This study employs the simulation package SIMUL8 [37] to implement and execute a simulation for the ODM model. This package is offered and maintained by SIMUL8 Corporation and provides an integrated medium containing a simulation designer and many results analysis tools.

A case study has been carried out to explore the actual structure of the customization process and gather the required data to guide the simulation model [28]. The case study is an existing project utilized by a software development organization, which develops software and subsequently modifies them based on the requests of the customers [28,33]. This case study contains 18 customers spread out across 16 cities. In that case study, the decision-making is assigned to a team at the client location, and the development is assigned to a group at the vendor's area [28].

#### **5.6 The Study Design**

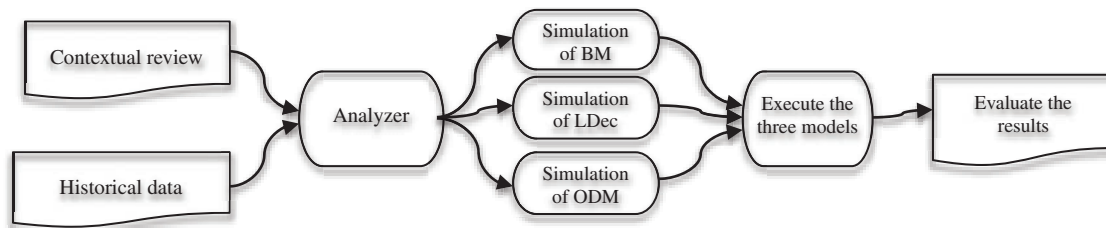
The variables of this study are as follows.

- (1) The object of analysis: The utilized case study contains 18 customers across 16 cities who worked on the same project and demanded 3000 requests within 1290 working hours. Data used in this work are accurate and gathered over an actual case study in a present software development organization. There is no authorization to refer to the organization's title, deploy or employ its data in the absence of a license.
- (2) Local development-based customization activity.
- (3) The production ratio: The mean of the wholly developed requests.
- (4) Time reducing ratio: the amount of the time of the customization process in the case of local development and offshore development.
- (5) Cost reducing ratio: the amount of the cost of the customization process in the case of local development and offshore development.



### 5.7 Procedures

This section presents the specifications and processes utilized in the experiments. The phases of the experimental study (see Fig. 4) begin by analyzing the real-world case study. Then using the data gathered from the case study and the literature review for the activities and knowledge sources of the customization product in that case study, the settings of the simulation patterns are defined. Data gathered through monitoring the case study were around 3000 requests collected in about 1300 working h. These requests/requirements came from 18 customers spread out across 16 towns who shared in the same customization project of a software development organization that develops software for those customers and customizes this software according to the new requirements of those clients.



**Figure 4:** The architecture of the procedure of the experimental study

Secondly, to perform the experiments, the simulation of the basic model (given in Fig. 1), the local decision-making model (presented in Fig. 2), and the ODM model (shown in Fig. 3) are built employing the same configurations extracted from the case study considering the specific modifications concerning the idea of the ODM model that allocates the local development at the location of the client. Fig. 5 shows a screenshot of the simulation system for the ODM model.

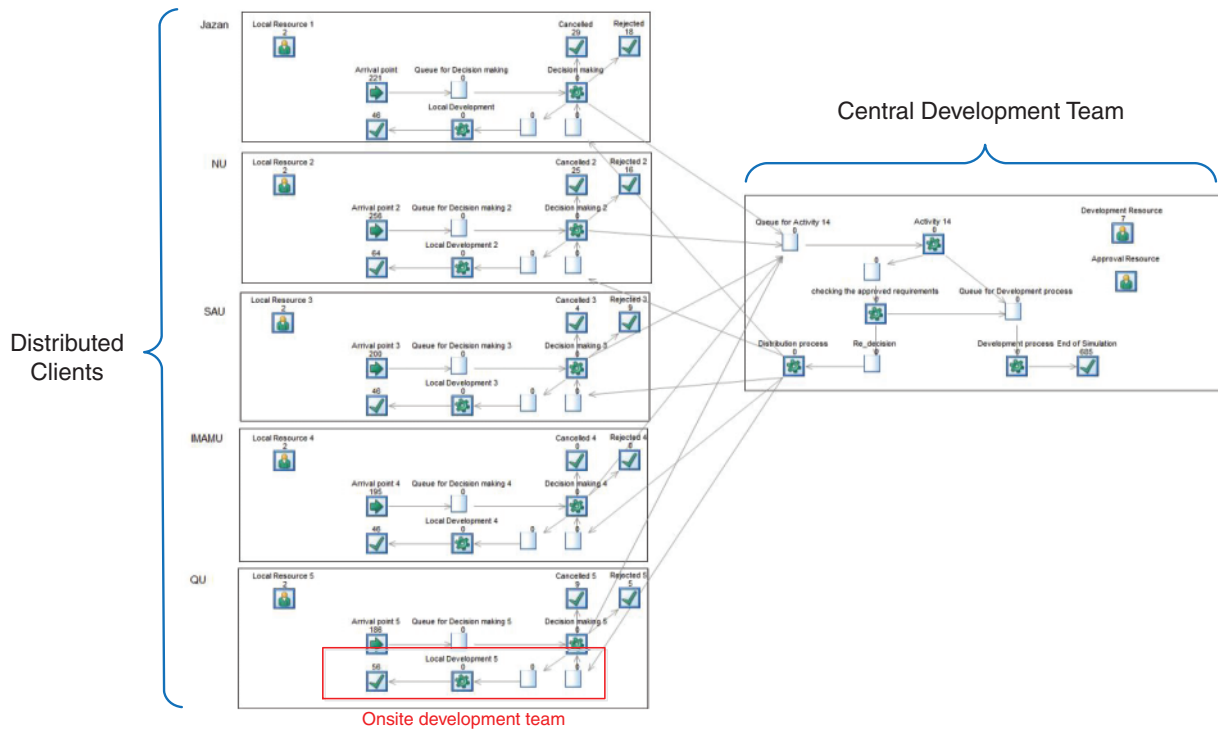
Thirdly, to assess the three models (BM, LDec, and ODM), the clients' requirements (as of arrival data) are inserted into the three models according to the arrival time. The arrival data created employing "Poisson distribution with  $\lambda = 0.365$ " to be similar to the arrival data in the case study.

Finally, the three models are executed, and their outcomes are statistically compared by "t-Test" to determine the significance of the difference between them in increasing the number of processed requirements and reducing the customization time and cost.

### 5.8 Results

The local development activities are allocated to a team at the place of the clients for handling particular forms of requirements (e.g., extracting a report according to the needs of the clients that neither modifies the software structure nor consumes a significant amount of time or critical requirements, which required to be treated locally to avert lateness in contacting through spread out borders).

These empirical studies target to assess the ODM model that employs the idea of localization in decision making and some development activities for reducing the issues of communicating customization requests for multi-client spread out across many cities. The given hypotheses are inspected and proven in these studies.



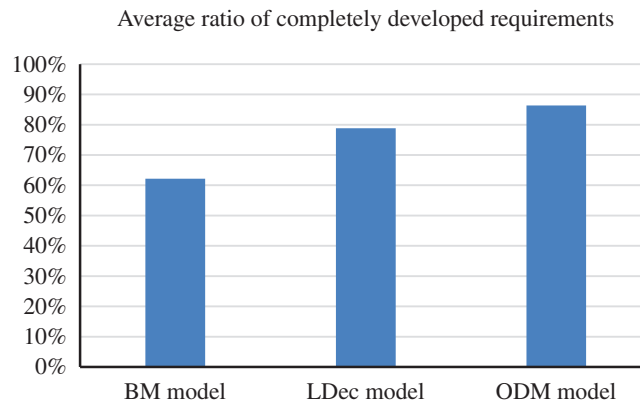
**Figure 5:** Simulation model for ODM

5.8.1 Results of the Empirical Study for Productivity

Productivity is a crucial concern for any software engineering activity. The primary objective of this experimental study is to estimate the productivity of the Basic model, the LDec model, and the ODM model in terms of the number of wholly developed requirements. In the simulation system, each model receives 600 requirements. These requirements reach each of the three models (BM, LDec, and ODM) in 30 batches containing 20 requirements. The run of the simulation process is repeated 100 times. According to the results of the simulation process given in Tab. 2, the BM model completed 373 requirements on average with a ratio of 62.2%, the LDec model met 473 requirements on average with a ratio of 78.8%, and the ODM model completed 518 requirements in average with percentage 86.3%. Fig. 6 compares the three models according to the ratio of the wholly developed requirements.

**Table 2:** Number of arrival and developed requirements

	BM model	LDec model	ODM model
Arrival requierments	600	600	600
Developed requierments	373	473	518
Ratio	62.2%	78.8%	86.3%



**Figure 6:** Ratio of completely developed requirements

The paired sample t-Test was conducted using the results of the BM model against the results of the LDec model and the ODM model to compare the efficiency of the three models (BM, LDec, and ODM). In addition, the paired sample t-Test was performed using the LDec model's results against the ODM model's results. The statistical results of the t-Test are given in [Tab. 3](#). The average of the developed requirements in 100 runs through 200 min of simulation time was 373 requirements under the BM model and 473 requirements under the LDec model. This average was 518 in the same configurations and environment under the ODM model. According to the results of the t-Test, the *P*-values are  $7.7E-34$ ,  $1.2E-32$ , and  $4.9E-22$ , and all values are less than 0.05 at a 95% level of confidence; which shows there is a significant difference between the productivity of the development process in the BM model and the LDec model, the BM model and the ODM model, and the LDec model and the ODM model. Generally, the results of the t-Test in this experimental study show a significant difference in the productivity between the three models, where the productivity of the ODM model is better than the productivity of both the BM model and LDec model by 39% and 10%, respectively. Also, the productivity of the LDec model is better than the productivity of the BM model by 27%. These results prove the rejection of the null hypothesis and the acceptance of the alternative hypothesis; there is a significant difference in the number of wholly developed requirements between the Basic simulation model, the LDec simulation model, and the ODM simulation model.

**Table 3:** t-Test paired two sample for means for productivity

	BM	LDec	BM	ODM	LDec	ODM
Mean	373	473	373	518	473	518
P (T <= t) two-tail		$7.7E-34$		$1.2E-32$		$4.9E-22$

### 5.8.2 Results of the Empirical Study for Time Performance

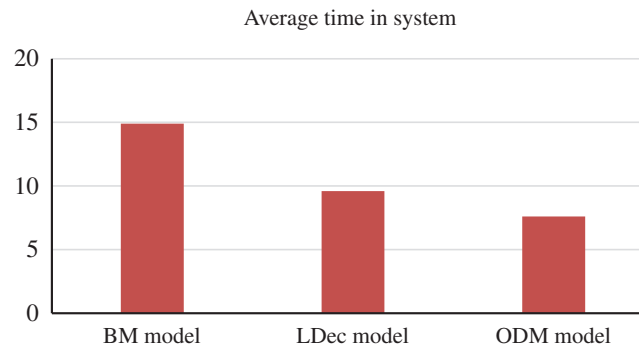
[Tab. 4](#) shows the average time taken by each model of the three models (BM, LDec, and ODM) and the *P* values generated by applying the t-Test paired two samples for means at a 95% confidence level. According to the results of the simulation process given in [Tab. 4](#), the BM model needs, on average,

14.9 min to ultimately develop 373 requirements, the LDec model requires on average 9.6 min to ultimately develop 473 requirements, and the ODM model needs in average 7.6 min to ultimately develop 518 requirements.

**Table 4:** t-Test paired two sample for means for time

	BM	LDec	BM	ODM	LDec	ODM
Mean	14.9	9.6	14.9	7.6	9.6	7.6
P (T <= t) two-tail	1.0E-36		5.9E-35		3.7E-44	

Fig. 7 compares the three models according to the average time of the wholly developed requirements. According to the results in Tab. 4, the ODM model reduced 48.9% of the development time taken by the BM model and 20.8% of the development time taken by the LDec model. In addition, the LDec model reduced 35.6% of the development time taken by the BM model.



**Figure 7:** Average time of each model

The paired sample t-Test was conducted using the results of the BM model against the results of the LDec model and the ODM model. In addition, the paired sample t-Test was performed using the results of the LDec model against the results of the ODM model. The statistical results of the t-Test are given in Tab. 4. According to the results of the t-Test, the  $P$ -values are 1.0E-36, 5.9E-35, and 3.7E-44, and all values are less than 0.05 at a 95% level of confidence; which shows there is a significant difference between the time performance of the development process in the BM model and the LDec model, the BM model and the ODM model, and the LDec model and the ODM model. Generally, the results of the t-Test in this experimental study show a significant difference in the time performance between the three models, where the ODM model is faster than both the BM model and LDec model by 49% and 20.8%, respectively. Also, the LDec model is faster than the BM model by 35.6%. These results prove the rejection of the null hypothesis and the acceptance of the alternative hypothesis; there is a significant difference in the time performance between the Basic simulation model, the LDec simulation model, and the ODM simulation model.

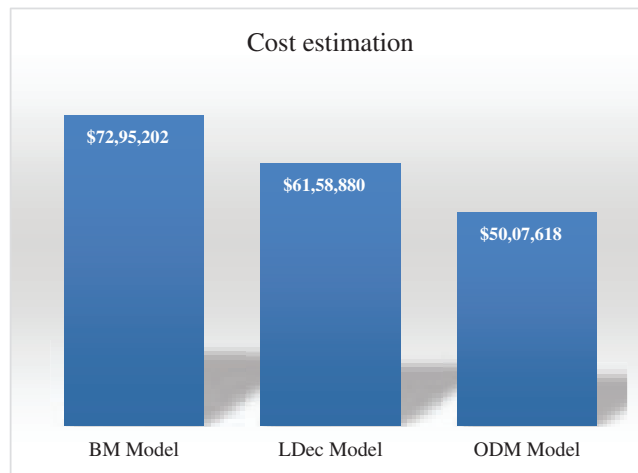
### 5.8.3 Results of the Empirical Study for the Cost Reduction

**Tab. 5** shows the total number of persons involved in the Basic model, LDec model, and ODM model and the total cost of each team in these models. The results were obtained from a study on 18 distributed clients contacting the offshore development team to customize a software system. The Basic model has 18 persons representing the development organization at the client's location, 17 offshore decision-makers, and 17 offshore developers. These persons contact the clients to get the requested requirements and pass them to the offshore decision-maker team to accept, reject, or change them. Generally, the delegates of the development company at clients' locations are supposed to be junior software engineers because those persons collect and analyze the requirement and deploy the changes that arrive from the offshore development team. Otherwise, a decision-maker must be a senior software engineer knowing clients' contracts to evaluate the changes. In the local decision-making model (LDec), the representative persons are excluded, the senior decision-makers change their place to clients' locations to shift decision-making activity to the areas of the clients, and the onshore developers (17 persons) continue in the exact location.

**Table 5:** Cost estimation

Person	Cost per person (\$/yr)	Basic model		LDec model		ODM model	
		no.	cost	no.	cost	no.	cost
Represintor	72905	18	1312290	0	0	18	1312290
Decision maker	175,968	17	2991456	18	3167424	0	0
Local developer	175,968	0	0	0	0	18	3167424
Main developer	175,968	17	2991456	17	2991456	3	527904
Total		52	7,295,202	35	6,158,880	39	5,007,618

Consequently, the number of persons involved in the LDec model has decreased from 52 in the basic model to only 35. In the local development model (ODM), there are 18 representatives, the decision-makers are excluded, and a number of the offshore developers (18 persons) are shifted to the client's locations with very few offshore developers (3 persons). Consequently, the number of persons involved in the ODM model has decreased from 52 in the basic model to only 39. According to these changes, the costs have been reduced from \$7,295,202 in the Basic model to \$6,158,880 in the LDec model and \$5,007,618 based on the salaries of junior software engineers and senior software engineers in the USA. Thus, the LDec model reduced 16% of the total cost of the Basic model, and the ODM reduced 31% of the total cost of the Basic model. **Fig. 8** shows the total price reduction after applying LDec and ODM models. The salaries were obtained according to the Glassdoor [38] website (last visit in May 2022). Glassdoor [38] is a website that contains actual salaries for a considerable number of jobs for employees of many large companies worldwide.



**Figure 8:** Cost estimation

## 6 Conclusion and Future Work

The current research aimed to examine local development's impacts on the DSD customization process. This research introduced a new software development model by moving the development process to the client's location. This work introduced three empirical evaluations for the efficiency of local development on the customization of software products in distributed development using three criteria productivity, time performance, and cost. The proposed model (ODM) has been compared with two models: the basic model, which allocates the decision-making process and the development process to persons at the vendor's location, and the local decision-making model, which assigns the decision-making process to persons at the client's location. The empirical studies showed that the proposed model outperformed the basic and local decision-making models in productivity, time performance, and cost reduction. The proposed model improved productivity by 39% and 10% more than the basic and local decision-making models, respectively. In addition, the proposed model enhanced the time performance to be faster by 49% and 20.8% than the basic and local decision-making models, respectively. Also, it reduced the total cost of the development process by 31% in terms of the salaries of all persons involved in requirements collecting, decision-making, and development. Future work may inspect the proposed model by utilizing different projects of diverse volumes with several customization requirements. In addition, the testing process in distributed development is crucial for future work. Besides, future work will compare the proposed model with other related models.

**Acknowledgement:** The authors deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project number (TURSP-2020/327), Taif University, Taif, Saudi Arabia.

**Funding Statement:** Researchers Supporting Project number (TURSP-2020/327), Taif University, Taif, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] B. Sengupta, S. Chandra and V. Sinha, "A research agenda for distributed software development," in *Proc. 28th Int. Conf. on Software Engineering (ICSE '06)*, New York, NY, USA, pp. 731–740, 2006.
- [2] S. Schneider, R. Torkar and T. Gorschek, "Solutions in global software engineering: A systematic literature review," *International Journal of Information Management*, vol. 33, no. 1, pp. 119–132, 2012.
- [3] M. Jiménez, M. Piattini and A. Vizcaíno, "Challenges and improvements in distributed software development: A systematic review," *Advances in Software Engineering*, vol. 2009, no. 4, pp. 1–14, 2009.
- [4] S. U. Khan, M. Niazi and R. Ahmad, "Critical success factors for offshore software development outsourcing vendors: A systematic literature review," in *Proc. 4th IEEE Int. Conf. on Global Software Engineering*, Limerick, Ireland, pp. 207–216, 2009.
- [5] C. Ranganathan and S. Balaji, "Critical capabilities for offshore outsourcing of information systems," *MIS Quarterly Executive*, vol. 6, pp. 147–164, 2007.
- [6] F. Q. B. Da Silva, A. C. C. França and R. Prikładnicki, "Challenges and solutions in distributed software development project management: A systematic literature review," in *Proc. 5th IEEE Int. Conf. on Global Software Engineering*, Princeton, NJ, USA, pp. 87–96, 2010.
- [7] D. Damian, "Stakeholders in global requirements engineering: Lessons learned from Practice," *IEEE Software*, vol. 24, no. 2, pp. 21–27, 2007.
- [8] L. Xu and S. Brinkkemper, "Concepts of product software," *European Journal of Information Systems*, vol. 16, no. 5, pp. 531–541, 2007.
- [9] T. Gregory and R. Baskerville, "Traveling of requirements in the development of packaged software: An investigation of work design and uncertainty," Ph.D. dissertation, Georgia State University, Georgia, USA, 2014.
- [10] R. Colomo-Palacios, C. Casado-Lumberos, P. Soto-Acosta, F. J. García-Peñalvo and E. Tovar, "Project managers in global software development teams: A study of the effects on productivity and performance," *Software Quality Journal*, vol. 22, no. 1, pp. 3–19, 2014.
- [11] Z. U. R. Kiani, D. Šmite and A. Riaz, "Measuring awareness in cross-team collaborations—distance matters," in *Proc. IEEE 8th Int. Conf. on Global Software Engineering*, Bari, Italy, pp. 71–79, 2013.
- [12] D. Damian and D. Moitra, "Guest editors' introduction: Global software development: How far have we come?," *IEEE Software*, vol. 23, no. 5, pp. 17–19, 2006.
- [13] R. Prikładnicki, J. L. N. Audy and R. Evaristo, "Global software development in practice: Lessons learned," *Software Process Improvement and Practice*, vol. 8, no. 4, pp. 267–281, 2003.
- [14] R. Prikładnicki, J. L. N. Audy, D. Damian and T. C. d. Oliveira, "Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring," in *Proc. Int. Conf. on Global Software Engineering (ICGSE '07)*, Munich, Germany, pp. 262–274, 2007.
- [15] E. Carmel and R. Agarwal, "The maturation of offshore sourcing of information technology work," in *Information Systems Outsourcing*, 2<sup>nd</sup> ed., vol. 1. Heidelberg, Berlin, Germany: Springer, pp. 631–650, 2006.
- [16] G. Höfner and V. S. Mani, "TAPER: A generic framework for establishing an offshore development center," in *Proc. Int. Conf. on Global Software Engineering (ICGSE '07)*, Munich, Germany, pp. 162–172, 2007.
- [17] M. Ribeiro, R. Czekster and T. Webber, "Improving productivity of local software development teams in a global software development environment," in *Proc. 2006 IEEE Int. Conf. on Global Software Engineering (ICGSE '06)*, Florianopolis, Brazil, pp. 253–254, 2006.
- [18] C. Ebert, "Optimizing supplier management in global software engineering," in *Proc. Int. Conf. on Global Software Engineering (ICGSE '07)*, Munich, Germany, pp. 177–185, 2007.
- [19] D. Damian and D. Zowghi, "Requirements engineering challenges in multi-site software development organizations," *Requirements Engineering*, vol. 8, no. 3, pp. 149–160, 2003.
- [20] R. Akbar, M. Harris and M. Naeem, "Agile framework for globally distributed development environment (The DAD Model)," in *Proc. 8th WSEAS Int. Conf. on Applied Informatics and Communications*, Rhodes, Greece, pp. 423–428, 2008.

- [21] R. Phalnikar, V. S. Deshpande and S. D. Joshi, "Applying agile principles for distributed software development," in *Proc. 2009 Int. Conf. on Advanced Computer Control*, Singapore, pp. 535–539, 2009.
- [22] S. Jalali and C. Wohlin, "Agile practices in global software engineering-A systematic map," in *Proc. 5th IEEE Int. Conf. on Global Software Engineering*, Princeton, NJ, USA, pp. 45–54, 2010.
- [23] W. S. W. Husin, Y. Yahya, N. F. M. Azmi, N. N. A. Sjari, S. Chuprat *et al.*, "Risk management framework for distributed software team: A case study of telecommunication company," *Procedia Computer Science*, vol. 161, no. 2, pp. 178–186, 2019.
- [24] J. F. DeFranco and P. A. Laplante, "Review and analysis of software development team communication research," *IEEE Transactions Professional Communication*, vol. 60, no. 2, pp. 165–182, 2017.
- [25] A. L'Erario, J. A. Gonçalves, J. A. Fabri, T. Pagotto and R. H. C. Palácios, "CFDSD: A communication framework for distributed software development," *Journal of the Brazilian Computer Society*, vol. 26, no. 7, pp. 1–21, 2020.
- [26] M. A. P. Damato, A. L'Erario and J. A. Fabri, "Awareness, collaboration and communication's tools for distributed software development: A systematic mapping," in *Proc. 2016 35th Int. Conf. of the Chilean Computer Science Society (SCCC '16)*, Valparaiso, Chile, pp. 1–8, 2016.
- [27] R. Jolak, A. Wortmann, M. Chaudron and B. Rumpe, "Does distance still matter? Revisiting collaborative distributed software design," *IEEE Software*, vol. 35, no. 6, pp. 40–47, 2018.
- [28] A. S. Ghiduk and A. M. Qahtani, "An empirical study of local-decision-making-based software customization in distributed development," *IET Software*, vol. 15, no. 2, pp. 174–187, 2021.
- [29] M. Robinson and R. Kalakota, "Offshoring strategy creation and execution," in *Offshore Outsourcing*, 2<sup>nd</sup> ed., vol. 1. Alpharetta, GA: Mivar Press, Inc, pp. 211–301, 2004.
- [30] D. Šmite, W. Wohlina, A. Aurumc, R. Jabangwe and E. Numminen, "Offshore insourcing in software development: Structuring the decision-making process," *Journal of Systems and Software*, vol. 86, no. 4, pp. 1054–1067, 2013.
- [31] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, vol. 18, no. 2, pp. 16–20, 2001.
- [32] D. Damian, "Requirements engineering in distributed projects," in *Proc. 2006 IEEE Int. Conf. on Global Software Engineering (ICGSE '06)*, Florianopolis, Brazil, pp. 69, 2006.
- [33] A. M. Qahtani, G. B. Wills and A. M. Gravell, "The impacts of local decision making on customisation process speed across distributed boundaries: A case study," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 1, pp. 63–68, 2015.
- [34] S. Gopalakrishnan, V. P. Kochikar and S. Yegneshwar, "The offshore model for software development: The Infosys experience," in *Proc. 1996 ACM SIGCPR/SIGMIS Conf. on Computer Personnel Research (SIGCPR '96)*, New York, NY, USA, pp. 392–393, 1996.
- [35] D. E. Perry, A. A. Porter and L. G. Votta, "Empirical studies of software engineering: A roadmap," in *Proc. of the Conf. on the Future of Software Engineering (ICSE '00)*, New York, NY, USA, pp. 345–355, 2000.
- [36] M. Zahedi, M. Shahin and M. A. Babar, "A systematic review of knowledge sharing challenges and practices in global software development," *International Journal of Information Management*, vol. 36, no. 6, pp. 995–1019, 2016.
- [37] Simul8\_Corporation, "SIMUL8," 1994. [Online]. Available: <https://www.simul8.com> [Accessed March 2022].
- [38] G. Inc, "Glassdoor," 2008. [Online]. Available: <https://www.glassdoor.com> Glassdoor Inc., [Accessed 24 May 2022].