



PRIORITIZED RISKS AND TREATMENT STRATEGIES IN GLOBAL SOFTWARE DEVELOPMENT (GSD)

M. S. M. Hasni^{1,*}, A. I. G. Masbah², T. H. El hasady³

¹Derna University, Derna, Libya, mohamed.hasni@omu.edu.ly

²College of Technical Science, Derna, Libya, aimn_gramay@yahoo.com

³College of Technical Science, Derna, Libya, tarek.h.elhasady@hotmail.com

ABSTRACT

The study conducts a review for the literature in order to compile the risks that are directly related to the GSD strategy. The case study adopts a Delphi methodology that allows the researcher to achieve consensus on the most relevant and critical risks associated with the GSD project. Twenty software development experts from all around the world participated in the study, with a minimum of fifteen expert in each Delphi round. The four rounds of the Delphi method used in this study are designed to gain consensus on the most crucial risks of the GSD strategy, as well as perform a risk assessment for all the risks compiled from the literature and verified by the participating experts. The results show that there are ten main risks that gained consensus by software development experts and need to be addressed as a priority in GSD projects, where all of them have medium to high probability of occurrence and impact on the software project success using a GSD strategy.

Keywords:

Global Software
Development (GSD).
Risk Management.
Delphi Technique.

*Corresponding Author Email: mohamed.hasni@omu.edu.ly

1 INTRODUCTION

The term Geographic Software Development (GSD) is defined as software projects that distributed among several teams or individuals that could be located in different geographic locations. Although sources have conveyed this definition through different terms [1,2], this study will use the term Geographic Software Development (GSD) as it includes a more generic meaning than other terms, such as [3]:

- Distributed Software Development (DSD): which refers to corporate effort that compiles the efforts of several teams located around the world for software development, working on parts of the software separately.
- Dispersed Software Development (DSD): which refers to individuals who are working together on a software but located in different sites.
- Global Software Development (GSD): which refers to different teams or individuals that located in different parts of the world.

There are several studies that have focused on risk management in GSD projects [2,4,5]; however, there are few studies that tried to establish standardized risks for GSD circumstances. Thus, this research is focused to compile the most important risks that are associated with GSD projects, as well as connecting them to the best risk treatment strategies, and testing their validity and importance through experts and project managers who have well-established practical experience in the domain.

There are many risks that are associated with Geographic Software Development (GSD) that are different in nature than developing the software in-house, exceeding its difficulty and the nature of its risk management process [6,7]. The main reason behind the difficulty of such projects are the many challenges that accompany them, which result into forming risks on the project development's success, since the word risk originally mean "to dare", taken from its Italian roots [2]. Moreover, the efforts in research and development in the subject of risk management in the Geographic Software Development (GSD), also known as Distributed Software Development (DSD), have been mainly focused into two directions:

- Defining the risks that are typically associated with the process [4,8].
- Defining the risk management process in order to provide a standardized system in identifying, assessing, mitigating, controlling and monitoring the risks, taking into consideration the operational, strategic and tactical levels [2,9].

Further studies are reviewed in the coming sections in order to understand the results found through these two directions. However, it is evident that the literature did not work towards the standardization of risks and their associated mitigation strategies as a guide for project managers in GSD projects. Therefore, the literature research directions are studied for risks and mitigation strategies compilation, which is needed in the case study.

The risks that are emerging in software development, similar to other industries, are mainly associated with the time and cost of the project and subsequently impose risks on the same factors [1]. Nevertheless, the Geographic Software Development (GSD) adds a new source of risk, which is the location or the geographic dimension. Thus, the introduction of new location or several locations in the software development process have its unique risks such as [4,10,11]:

- Communication
- Coordination
- Trust
- Commitment
- Logistics

Furthermore, there are software that are used in order to import the project's schedule, identify its issues according to set parameters and perform a risk analysis and adjustments to minimize risk and help develop the mitigation plan. However, such software has initially two main issues [12]:

- They mainly used in projects of a localized nature, as it does not take into consideration the type of risks that were mentioned earlier, which are mainly emerging from distributing the project over several locations.
- They do not provide guidance and access to mitigation strategies that have proven its effectiveness in reducing the risk impacts, rather than calculating risk probabilities and adjusting the project schedule.

The significance of this research is emerging from the lack of a tool that provide a standardized risk management guidance for GSD project managers that could save efforts and increase the possibility of reducing the risk impacts in the software project lifecycle. Moreover, this tool contributes into expertise sharing between different specialists in the domain, which could be an important step to unify the strategies used in GSD project management. Furthermore, the current tools used in GSD project managers are reviewed in the literature in order to highlight their advantages and disadvantages in the management process. Additionally, integration possibilities are studied in order to help the current tools' users to benefit from the research results and save any time and effort that could be spent in the implementation process.

1.1 Global Software Development (GSD)

In setting up a GSD project, the initial goal is to manage the project in a global context while administrating operations among the different sites. The GSD project managers focus their efforts on defining the team structure and allocating the different tasks according to the skill pool available for them. Moreover, implementing a strong communication strategy is essential in order to ensure a smooth operation between the different sites. In order to ensure that all skills are covered within the team, project managers propose a training program for the team members. During operations, management strategies, procedures, coordination tools, meeting policies, incentive plans and role descriptions are implemented as basic features for the GSD project [13,14].

1.2 Risk Management in Software Development

The main aim of risk management in software development is to minimize the impact of unforeseen events that could jeopardize the success of the project through increasing costs, time. Moreover, it is possible to turn risks into opportunities that the project could benefit from through an extensive assessment and knowledge of the possible treatment strategies that can be used [15]. There are three main goals that the risk management strategy shall ensure in software development project: completion of the development according to the time schedule, completion of the development according to the set budget, completion of the development with the required functionality. The ratio of failed software projects increases the urgency to implement risk management strategy, techniques, and tools, as 25% of the software projects fail and more than 40% face risks that could lead to failure [16,17].

Through a literature survey of the studies that addressed risk management in software development in the past thirty years, Neves & da Silva (2016) showed that all of the software

projects adopt project management systematics in risk management, while 85% depend on lessons learned in order to identify, assess and control risks. Moreover, the authors confirmed the large number of projects that are either faced with high impact risks or cancelled due to a very high risk factor. Of the 535 publications that addressed risk management in projects, 61% were addressing software in risk development, indicating the significance of the subject for the domain [18,19].

Pimchangthong & Boonjing [20] identified four main practices for risk management, which are followed by information technology projects, and used for project management and risk management worldwide; risk identification, risk analysis, risk response and risk monitoring and control. These practices are set to monitor two main performance factors within the project; process and product. The process performance is measured through the completion of the project within the planned budget and schedule. Nonetheless, the product performance is measured through the reliability of the developed system, its usability, its flexibility, meeting user's requirements, meeting the functional requirements, user satisfaction and overall high quality [20,21].

Risk Identification

An informal approach can also be used through a discussion between the project stakeholders in order to list the possible sources of risk in the project through their past experiences. The periodic approach is performed through deploying certain tools to check for risk potentials in the different aspects of the project. The formal approach is carried out as risk management specialists are employed in order to produce a report of the current risks that are faced by the project, as well as potential risks that may arise in the future [15]. Furthermore, in studies that addressed risk in software projects adopting the extreme models, the authors confirmed that the most efficient method in order to identify risks in software projects is by either surveying the literature in order to find out the risks from the past experiences and lessons learned, or through identifying a full set of the software development operations and performing brainstorming sessions to identify the potential risks that are associated with each process or operation [22]. Moreover, there are recommendations to implement brainstorming and surveying methodologies, such as risk breakdown structure, event and defect trees, as well as using external consultants by the Delphi technique, which is the methodology followed in this research [23].

Risk Analysis and Assessment

de Wet & Visser [16] presented a model for risk assessment through a risk analysis using several tools, including performance models, cost models, network analysis, decision analysis, and quality factor analysis. Based on the performed analyses, the risks can be prioritized according to their positive or negative impacts on the project. Therefore, risks are ordered according to treatment and importance priority based on risk leverage and risk exposure, as well as applying a compound risk reduction to each one of them.

The most important criteria to identify during the risk analysis is the occurrence probability of the risk and the expected impact of the risk on the time and cost factors of the project. If the risk has a chance of more than 70% to occur, the probability is marked as high, while a score between 30% to 70% is given to risks with medium occurrence probability, and low occurrence risks are assigned to scores below 30%. Moreover, the impact of the risk is evaluated according to the impact on the project's budget. If the risk is expected to cause the failure of the project through missing the launching date of the software or increasing the project cost by more than 50%, the risk is considered high and catastrophic. If the impact of the risk would cause issues in the project in term of recoverable time schedule impact and cost increase between 10% to 50%, then the risk impact is considered medium and critical. If the risk would cause minor issues to the project with time impact not affecting the launch date and cost impact of less than 10% of the project budget, then the risk is considered low and marginal [24]. Based on that, risk score is calculated on the risk analysis matrix, shown in Figure 1, and through the following equation:

$$\text{Risk score} = \text{probability} \times \text{impact}$$

		Impact		
		Low	Medium	High
Probability	High	I1P3	I2P3	I3P3
	Medium	I1P2	I2P2	I3P2
	Low	I1P1	I2P1	I3P1

Figure 1. Risk analysis matrix [24]

Figure 2 shows the four quadrants and their representation. The assignment of the risks to the quadrant can be accomplished through the assessment of field experts [25], which is a methodology adopted in this research.

Q4	10	Q1
High Probability Low Impact	Probabilit	High Probability High Impact
0		10
Low Probability Low Impact		Low Probability High
Q3	0	Q2

Figure 2. Risk assessment representation on four quadrants [25]

Risk Treatment and Response

There are basic treatment and response strategies that are used in risk management, which are applied to software development projects. Moreover, the choice of the treatment strategy mainly depends on the type of the risk, its probability, its impacts, its occurrence closeness, its frequency and the possibility of the treatment strategy implementation. Once the treatment strategy is selected, along with the specific measures to be taken, the risk is assigned to one of the team members in order to follow its implementation [26]. The treatment strategies that are standardized in risk management practices are as the following [27,28]:

- Risk avoidance: a strategy that eliminates the risk by eliminating the related activity that is associated with it. The selection of this strategy is attributed to very high impacts or losses that are expected to be imposed on the project. It is also possible to change the method that is used for the activity execution, if the analysis shows that it is the source of the risk. Since adopting an avoidance response would possibly affect the scope or the course of the project, it is crucial that proper and sufficient communications are established between the project stakeholders in order to reach a common understanding of the risk and the necessity of the measure.
- Risk transfer: in case the risk had a low probability of occurrence and a high impact on the project, this strategy is preferred by the project managers. In this case, the risk is transferred to a third party that can manage the risk due to its specialty in it. Furthermore, the risk can be transferred to another phase of the project, where it can be handled with less impact.

- Risk mitigation: this measure includes reducing the risk to an acceptable level, which can be achieved by different approaches such as escalating the matter to a higher management level for a strategic decision, performing a sensitivity analysis, or finding an engineering solution for the issue causing the risk.
- Risk acceptance: this measure occurs when the impact of the risk is accounted for by allocating the required contingency or accepting the losses due to the low impact expected to be imposed on the project.

Risk Control and Monitoring

In the risk control and monitoring phase the actions and treatment strategies that were set for the existing risks are followed through. While the effectiveness of the taken measures is evaluated, new risks are identified and added to the tracking logs for analysis, assessment and treatment [29]. Different types of data are collected about the different risks in order to update their status on a periodical manner. The data include financial, schedule, technical, managerial and supply chain information that could assist the project manager to take further decisions regarding the risk status [30,31].

1.3 Survey of GSD Risks, Challenges, and Issues

As risks are usually caused by challenges to the projects, it is significant to understand them in order to be able to understand the risk root causes. Furthermore, challenges and issues can be used to conclude indirect risks in GSD projects, which can be included in the risk lists of the current study. Aranda, Vizcaino, & Piattini [32] identified four main factors that cause the challenges and issues in GSD projects, which are:

- Timetable challenges.
- Cultural challenges.
- Knowledge management challenges.

Jimenez et al. [33] have identified nine main challenges that faces software projects adopting a GSD strategy as the following:

- Communication challenges.
- Configuration management challenges.
- Knowledge management challenges.
- Quality management challenges.
- Project management challenges.
- Process support challenges.
- Coordination challenges.
- Collaboration challenges.

A critical review for the global software development strategy surveyed the literature for the main issues that face projects that adopt the strategy. As shown in Table 1, while the GSD strategy provides several advantages on all of the studied elements, new risks are imposed on

them [34]. The presented risks are attributed according to their discipline for classification purposes beneficial for the current study.

Table 1. Implications of the GSD strategy on the different software development project elements [34]

Elements	Advantages	Risks	Risk Discipline
Cost	Lowering labor costs	Increase of running costs	Financial
Skill	Access to a larger skill pool	Inadequate skill level	Human Resources
Operations	Follow-the-sun development	Minimized instantaneous communication	Communication
Quality	Added expertise by other teams	Incompatible quality standards	Quality
Culture	Widens perspectives and enhance opportunities	Issues due to cultural differences	Coordination and management
Language	Enhanced communication with stakeholders through language diversity within the team	Establishing a language barrier	Communication

In a study that focused on the communication issues that face software development projects with GSD strategy, the authors provided two main strategies to resolve the related risks; introduction of several communication tools through scrum and usage of version one software [1]. On GSD challenges types, Tihinen [35] classified the risks into five main categories:

- Communication.
- Coordination.
- Control.
- Cohesion.
- Culture.

Conchuir, et al. [36] identified three distances and three project elements that are resulting and affected by using the GSD strategy. Moreover, the authors developed a matrix showing the opportunities and challenges that are imposed by each of the distances on the software development project element. The first distance is identified as the temporal distance, which is the time distance that separates the different locations within the project. The socio-cultural distance is the differences between the team members in different locations in language, working habits, professional conduct and business manners, which could cause misunderstandings and miscoordination between the team members. The geographic distance is the most physical distance, which is pursued due to certain advantages; however, this distance is main cause for the temporal and socio-cultural distances. Table 2 shows the positive and negative impacts of the three distances on three main project elements: communication, coordination, and control.

Table 2. Positive and negative impacts of GSD distances on project elements [36]

	Temporal	Geographical	Socio-Cultural
Communication	Better communication record but less chance for synchronized communication	Appropriate teams can be placed near potential clients and far skilled workforce can be accessed, but face to face meetings are not always possible	Exchange of expertise and knowledge is facilitated, but there is possibilities for cultural misunderstandings
Coordination	As coordination requires additional resources to be achieved, any coordination will be an added cost.	Coordination can be planned in a more flexible manner, but with reduced informal contact, this could lead to less awareness of critical tasks	More powerful expertise sharing and learning; however, inconsistency in work practices and reduced cooperation caused by misunderstandings have higher potential to occur
Control	Work can be performed throughout the day hours, but management issues can cause delays	Communications can be audited; however, conveying motivation and project vision is difficult	Differences in proactivity and perception of authority from one culture to another, as well as adhering by local regulations

Prikladnicki, et al. [37] identified four main issues that face software projects with GSD strategy: strategic, cultural, knowledge and technical. There are risks associated with software development that emerge from the core and subsidiary operations of the project. It was found that low user involvement and unrealistic time schedule and budget estimations are the most common reasons behind risks in software development projects. Moreover, ambiguous or misunderstood project scope and objectives, understaffing, and lack of senior management commitment and technical knowledge are important factors in causing risks. Therefore, such issues can be resolved by adopting risk management techniques that could alleviate their impacts, as well as eliminate any possible risks that are associated with them [38].

Lamersdorf & Munch [39] provided a study on the main goals from adopting a global software development strategy, which are affected by a set of factors the accompany the adoption of such a strategy. As shown in Figure A, fifteen factors are the direct and indirect cause for four main risks impacting the goals of GSD projects. Furthermore, the main risks that are found in this study are as the following [39]:

- Cost overhead.
- Lack of trust.
- Productivity.
- Communication, coordination, and control.

Widiyatmoko [40] identified seven main risks associated with adopting the GSD strategy in software engineering. The first risk is having insufficient direct communication between the

different parts of the team, which mainly affects the problem-solving time and efficiency. Moreover, as tasks within software development projects may depend on each other, delays imposed by certain part of the team can affect the progress of the work for other teams. The overlapping of working hours is also one of the issues that face GSD projects, which its lack can results into inadequate collaboration and subsequent delays. Knowledge differences between the different teams imposes a high risk of distorted information, which can result into quality and rectification processes. Furthermore, the failure of the project management team to provide face-to-face communication tools can add necessary but extra travelling costs. As the GSD teams can be located in different countries, a lack of common understanding can be caused by different expertise and knowledge among the team. The last identified GSD risk within the study is the weakness in control over the project as conventional monitoring tools need to be modified to account for the geographical distance [41].

2 MATERIALS AND METHODS

2.1 Delphi Technique

The Delphi method or technique is defined as a process to gain consensus through a collective judgement about an event or a phenomenon by a group of experts. The results shall be empowered by a structured knowledge, experience and creativity from a group of experts [42]. Moreover, this method has been well-utilized as a stand-alone or a combined method for studies that are related to risk management in different industries [43]. The process of the Delphi methodology may vary from a study to another; however, there studies that standardized the process for usage facilitation and understanding. Hsu et al. [44] explained the process as the following:

- Round one: a part of the study, where usually open-ended questions are used in order to develop the knowledge gathered from the theoretical review. This round is often referred as the data collection round, which can be performed through single or multiple questionnaires until the researcher is certain that the provided items represent the topic of research.
- Round two: this round includes testing the results of the first round with the experts' panel through sharing the finalized items with them. This round can be performed through asking the experts to indicate the correctness of the items, to rank-order them, or to provide comments on them.
- Round three: the results of the second round are shared with the experts, similar to the first round, where the opinions of the experts are taken into consideration to reach a certain level of consensus.
- Round four: the final items and their ratings that achieved the consensus are shared with the experts for final comments.

During the development and the carrying out of the Delphi method, the experts' quality is very critical to choose the experts to be from top management who are willing to use the

results of the study, professional employees who have field experience or selecting experts based on their achievement in the field of the research [44]. The majority of the studies look for conducting the Delphi study through one to three rounds, and with a number of experts that is ranging from three to more than hundred. Nonetheless, Skulmoski et al. [45] shows the strength of the Delphi study results can be increased by performing a wide literature review along with a pilot study to ensure that the items initially included are the best representing items. The first round's questionnaire is mainly developed through two processes, which are the literature review and the brainstorming based on the researcher's experience. Thereafter, the pilot study is performed on a trial sample in order to ensure that the questions are well-understood and to eliminate any confusions. Subsequently, the different rounds are performed similar to the process reviewed in [44].

2.2 Research Design

The application of the Delphi methodology is adopted according to the different sources reviewed in the previous section. While the main Delphi structure is maintained, the main aim of the design is to simulate brainstorming between the participating experts as recommended by Tavares, et al. [22] and Didraga [23] as an essential activity in risk management, as well as measuring consensus between experts in software development provided by the method. The rounds are designed to verify the compiled risks that are reviewed in the literature, test the relevance of the risks to global software development projects, assess the extents of the risks, and provide the best treatment strategy for each risk. The study includes four main rounds that aim to achieve the objectives through four rounds, as the following:

- Data collection round (Round 1): the compiled risks are provided to the participating experts for verification. The experts are able to retain, delete and modify risk descriptions according to their experience in the field. The opinions of the experts are collected through a narrative form, in addition to action selection. The researcher judges the feedback of the experts in order to determine whether an item should stay or be removed from the risk list. Moreover, items may be modified to describe the risks in a better way.
- Relevance round (Round 2): the qualified risk list from the first round is provided for experts for voting based on the relevance of each item to GSD projects. The risks are categorized and judged by the participating experts on a 5-point Likert relevance scale from very relevant to very irrelevant. No elimination is used in any of the subsequent round in order to calculate the consensus for each item and apply a risk assessment on them based on the experts' expertise.
- Impact assessment round (Round 3): the experts are asked in this round to indicate the probability of occurrence and the severity of the risks, similar to the risk assessment models reviewed in Lopez and Salmeron (2012) and Vahidnia, et al. (2017). Based on the results further discussions are provided for the two models. The risk probability of occurrence is evaluated on four percentage scale options; 0% to 25%, 25% to 50%,

50% to 75% and 75% to 100%. The severity of the risk impact is judged on a three-option scale; low, medium and high.

- Risk treatment round (Round 4): this round aims to indicate the most suitable risk treatment strategy for each risk item based on the expert's field experience. Four main risk treatment strategies were identified in the literature: reduction, transference, acceptance and avoidance. The reduction of the risk corresponds to other detailed mitigation strategies that are specific for the risk and the project's environment.

The four rounds that form the case study in this research are an efficient tool that uses the brainstorming technique to collect experts' opinions, while performing a risk assessment session that is based on consensus. Therefore, it is expected at the end of this research to provide a full list of the risks with their assessments, as well as a final list of the most items that should be prioritized in GSD projects.

2.3 Participating Experts

In order to gather the opinions of GSD experts, more than a thousand software development experts were invited to the study through LinkedIn. The final participating experts are shortlisted based on their experience in software development, experience in global software development, number of GSD projects participated in and the level of commitment to the study. A total of twenty experts were gathered, as shown in Table 3 along with their profiles. At the beginning of each round, an email is sent to the experts notifying them of the aim of the round, the link to the survey and providing them with a deadline to complete the task. Once the opinions of fifteen experts were collected past the deadline date, the data collection was closed, and a new round was initiated. The study has been conducted between the months of June and October 2018, where experts were given one to two weeks for each round.

Table 3. Participating experts' profiles and round participations

Expert No.	Country	Years in software development	GSD experience (Years)	No. of GSD Projects	Rounds Participation			
					R1	R2	R3	R4
1	US	14	14	>15	•	•	•	•
2	UK	12	4	2	•		•	
3	EG	7	5	6	•			•
4	TR	5	3	2	•	•		
5	TR	5	5	4	•	•	•	•
6	NZ	4	3	2	•	•	•	•
7	US	12	7	>15	•	•	•	•
8	IN	16	16	12	•	•	•	•
9	IN	3	3	2		•	•	
10	UK	6	5	2		•		•
11	TR	10	5	4			•	•
12	US	5	5	4	•	•	•	•
13	US	6	3	4	•			•
14	AU	4	2	1		•		
15	US	20	10	>15	•	•	•	•
16	US	13	10	6	•	•	•	•
17	DE	6	2	3	•	•	•	•
18	UK	10	4	6	•	•	•	•
19	US	7	3	4	•		•	
20	NL	3	3	2		•	•	•

2.4 List of Initial GSD Risks

Based on a literature research, fifty-nine risks have been collected and distributed into five main categories, as follows below. The categories had been divided and named according to project management categorization. One category has been added, the fourth category in line with the literature endorsement of the importance of communication, coordination and collaboration in the GSD project. The other categories contain risks that are viewed by the researcher as important risks to be considered from a software project management point-of-view.

- Category A: Financial risks (5 risks),
- Category B: Operational and planning risks (13 risks),
- Category C: Management and human resources risks (17 risks),
- Category D: Communication, coordination and collaboration risks (15 risks),
- Category E: Technical risks (9 risks).

The risks and their categories are shown in Table A1 in the appendix. The categorization ensures the coverage of several aspects that are considered influential on the success and failure of the software project. The financial aspect of the project is essential as it impacts the continuity of the project. Operations, planning, management, and human resources are project

management aspects that affects running the software project in a smooth manner. The communication, coordination and collaboration risks have their own dedicated category due to its significant impacts on global software development projects. Finally, technical risks are gathered in a separate category to reflect the issues that can face GSD projects during the development from a technical point of view. The risk identification codes are unique codes according to the grounded theory method, which allows easy referencing and tracing risk items throughout the study.

3 RESULTS AND DISCUSSION

3.1 Results

the compiled fifty-nine risks were presented before the software development experts participating in the study. By the majority vote of the experts, sixteen risk items were deleted from the initial list of GSD risks, which led into reducing the risks qualifying for the second Delphi round into forty-three risk items. In the second round of the Delphi study, the risks were judged based on their relevance to GSD projects by the participating experts. The cluster mode method (CM) is used to evaluate the consensus on each item in this round, which requires a single rating out of the five Likert scale ratings to achieve a minimum of 50% in order to gain consensus.

Nineteen risk items have gained consensus using cluster mode; however, no elimination was made from the GSD risk list as further consensus is required, in addition for the value added of the risk assessment process that is carried out in the third round. In the third round of the Delphi study, each of the risk items have been assessed according to risk management practices based on their probability of occurrence and impact by the participating experts. The final consensus evaluation is conditioned for items that have achieved consensus based on the cluster mode method in the second round, as well as achieving consensus in both criteria in the third round through the same method.

The results of third round shows that ten GSD risks have gained consensus in the relevance, probability of occurrence and impact criteria. The full list is moved for a fourth round, where the forty-three risk items are voted for the best fit treatment strategy. The participating software development experts were asked to choose one of four main risk treatment strategies for each risk based on their expertise. A summary of the results of the four Delphi rounds is presented in Table A2 in the Appendix. The ten GSD risks that achieved consensus through the Delphi methodology are shown in Table 4 along with their best fit treatment strategies, as per the participating software experts. The final prioritized GSD risks includes one financial risk (A1), two operational and planning risks (B6 and B11), three management and human resource risks (C3, C9 and C14), two communication, collaboration and coordination risks (D1 and D11), and two technical risks (E2 and E3).

Table 4. GSD risks that achieved consensus through Delphi methodology

Risk ID	Risk Item	Category	Probability of Occurrence	Impact	Risk Score	Treatment
A1	Increase of running costs due to unforeseen factors in foreign countries	Financial	50% to 75%	Medium	I2P2	Accept/mitigate
B6	Lack of awareness of critical tasks	Operational and Planning	75% to 100%	High	I3P3	Avoid
B11	Poor telecommunication infrastructure in foreign countries		50% to 75%	High	I3P2	Transfer
C3	Inability to acquire the necessary skills for the project		75% to 100%	Medium	I2P3	Transfer
C9	Miscommunicated project goals and objectives	Management and Human Resource	50% to 75%	High	I3P2	Avoid
C14	Reduction of productivity due to lack of team awareness		50% to 75%	High	I3P2	Reduce
D1	Communication inefficiencies	Communication, collaboration and coordination	75% to 100%	High	I3P3	Reduce
D11	Lack of trust		75% to 100%	High	I3P3	Avoid
E2	Quality issues emerging from different quality standards and perceptions	Technical	50% to 75%	Medium	I2P2	Avoid
E3	Software compatibility issues		50% to 75%	High	I3P3	Accept/mitigate

3.2 Discussion

The probability of risk occurrence for the shortlisted risk items range between a high medium and high (50% to 100%) with medium and high impacts on GSD projects. Therefore, the risk scores are illustrated in accordance with the model provided by Vahidnia, et al. [24] in Figure 3. The risk scores for the shortlisted items range between I2P2 (second degree impact and second-degree probability of occurrence) and I3P3 (third degree impact and third-degree probability of occurrence).

		Impact		
		Low	Medium	High
Probability	High	I1P3	I2P3	I3P3
	Medium	I1P2	I2P2	I3P2
	Low	I1P1	I2P1	I3P1

Figure 3. Position of key GSD risks within the risk analysis matrix as presented by Vahidnia, et al. [24]

Moreover, a risk assessment model recommended by Lopez and Salmeron [25], which represents risks on four quadrants based on their probability of occurrence and impact on the project success. Therefore, the key GSD risks are represented in Figure 4 based on that model. It can be observed that all of the key GSD risks that are concluded in this research lay in the first quadrant (Q1), where high probability and high impacts are illustrated. Nonetheless, risk items B6, D1 and D11 have the highest probabilities of occurrence and highest impacts. Moreover, risk items B11, C9, C14, E2 and E3 have a medium high probability of occurrence and high impact. Risk item A1 has both medium probability of occurrence and impact on GSD project, while risk item C3 has a higher probability of occurrence with a medium impact.

Although global software development is meant to benefit from the follow-the-sun development and the larger pool of talents that are provided by several regions, as shown by Agerfalk, et al. [46] and Khan and Subhan [47], there are risks that directly related to the possibility of not finding such talent or the hindrance of the software development caused by time overlapping and holiday differences. However, there risks are found to be treatable through setting the proper processes and procedures to be followed by the project team, as well as hiring and empowering key personnel that can coordinate productivity with other development regions.

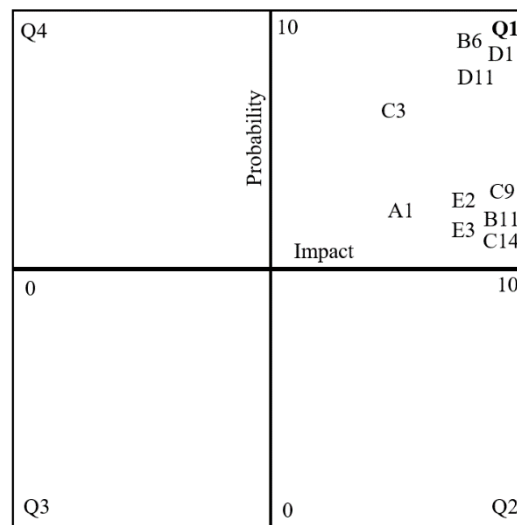


Figure 4. Risk assessment representation on quadrants for the key GSD risks as per the model of Lopez and Salmeron [25]

There are several benefits that can be accomplished by performing a sound risk management practice in GSD projects in order to allow the strategy to fulfil its goals as presented by Prikladnicki, et al. [2]. Creating trust between the developers and the clients is a key goal of the GSD strategy; however, clients can have lack of trust in development teams that are located in different geographic regions. Therefore, it is one of the essential tasks following the risk assessment in GSD projects to assure clients that quality, budget and time schedule are not compromised through adopting such a strategy. On the contrary, clients can be shown that adopting such a strategy would eventually save their time and money.

As the four risk management practices are identified by Pimchangthong and Boonjing [20]; identification, analysis and assessment, treatment and response, and monitoring and control. Three of these practices were performed in this research, where GSD risks were identified through the literature. In the first and second rounds, the compiled risks are analysed for comprehensivity and relevance. In the third round, the risks were assessed based on recommended models by Vahidnia, et al [24] and Lopez and Salmeron [25]. Finally, risk treatment strategies were selected and solutions were suggested through the fourth round. Risk monitoring and control can be achieved through creating a log, where all identified risks are entered, analyzed, assessed and assigned to treatment strategies and solution. Thereafter, the items can be reviewed on a periodic basis to ensure that they attended with the required measures.

4 CONCLUSIONS

The main problem addressed by this research is the new risks emerging primarily from the geographic distance that is created between the different segments of the development team,

which imposed new risks that need to be addressed. Therefore, the main aim of this study was to develop standardized top common risks in Geographic Software Development (GSD), associate the proven risk treatment strategies, and develop a priority within each list according to the impact of the risk and the effectiveness of the risk treatment strategy. Through an extensive literature review of more than forty studies on GSD risks, fifty-nine risk items were identified and categorized under five main categories:

- Financial risks.
- Operational and planning risks.
- Management and human resource risks.
- Communication, coordination, and collaboration risks.
- Technical risks.

The risks were categorized based on their source or impact on the software project that is developed in a globalized manner. The research adopted a grounded theory methodology to assign a unique identification number for each risk item. Furthermore, the Delphi methodology is utilized in order to gain consensus on the most important and relevant risk items for the GSD project. Subsequently, a panel of twenty software experts is created to perform the assessment of the Delphi rounds. The design of the Delphi rounds consisted on four main rounds to achieve consensus and to assess the risks and suggest treatment strategies for them.

The results of the study showed consensus on ten risk items, which are presented in Table 4. It is recommended based on the research findings to understand that global software development risks are unique for projects that uses the strategy; therefore, it is recommended to perform a separate risk management process for risks associated with the GSD strategy. Hence, risks that are concluded in this study are recommended to be prioritized in all projects adopting the GSD strategy. Furthermore, a special care should be given to the communication, collaboration, and coordination risks as they mainly emerge from the geographic, temporal, and socio-cultural distances created by the GSD strategy. It is also recommended to create partnerships within the software development domain, where trust, coordination and cooperation can be achieved over time. Moreover, this allows the teams to align their knowledge, expectations, and quality perceptions. In future research, this study can be repeated in a periodical basis using the Delphi methodology, as new challenges may raise in software projects adopting GSD strategy and for comparison purposes.

REFERENCES

- [1] A. Al Zaidi and R. Qureshi, "Global Software Development Geographical Distance Communication Challenges," *The International Arab Journal of Information Technology*, vol. 14, no. 2, pp. 215-222, 2017.

- [2] R. Prikladnicki, M. H. Yamaguti and D. C. Antunes, "Risk Management in Distributed Software Development: A Process Integration Proposal," in *Working Conference on Virtual Enterprises'04*, 517-526, Toulouse, France, 2004.
- [3] K. Mannaro, Adopting Agile Methodologies in Distributed Software Development (PhD Thesis), Cagliari, Italy: University of Cagliari, 2008.
- [4] E. Wallmüller, "Risk Management for IT and Software Projects," in *Business Continuity*, Berlin, Germany, Springer-Verlag Berlin Heidelberg, 2002, pp. 165-178.
- [5] C. B. Sahin, O. B. Dinler and L. Abualigah, "Analysis of Risk Factors in the Scope of Distributed Software Team Structure," *European Journal of Science and Technology*, no. 28, pp. 417-424, 2021.
- [6] D. Smite and J. Borzovs, "Managing Uncertainty in Globally Distributed Software Development Projects," *Latvijas Universitātes Raksti*, vol. 733, pp. 9-23, 2008.
- [7] M. Esteki, T. J. Gandomani and H. K. Farsani, "A Risk Management Framework for Distributed Scrum using PRINCE2 Methodology," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1299-1310, 2020.
- [8] Q. Khan and S. Ghayyur, "Software Risks and Mitigation in Global Software Development," *Journal of Theoretical and Applied Information Technology*, pp. 58-69, 2010.
- [9] J. Wan, D. Wan and H. Zhang, "Case Study on Business Risk Management for Software Outsourcing Service Provider with ISM," *Technology and Investment*, vol. 1, pp. 257-266, 2010.
- [10] S. R. Jan, F. Dad, N. Amin, A. Hameed and A. S. A. Shah, "Issues in Global Software Developmet (Communication, Coordination and Trust) - A Critical Review," *IJSRSET*, vol. 2, no. 2, pp. 660-663, 2016.
- [11] B. G. Tavares, C. E. S. da Silva and A. D. de Souza, "Practices to Improve Risk Management in Agile Projects," *International Journal of Software Engineering and Knowledge Engineering*, vol. 29, no. 3, pp. 381-399, 2019.
- [12] Oracle, "A Standardized Approach to Risk Improves Project Outcomes and Profitability (White Paper)," Oracle Corporation, Redwood Shores, CA, USA, 2010.
- [13] M. P. Baloch, S. Qadri, S. Hussain, S. Ahmad, A. Siddique and F. Azam, "Comparative Study of Risk Management in Centralized and Distributed Software Development Environment," *Science International*, vol. 26, no. 4, pp. 1523-1528, 2014.
- [14] H. A. Bilal, S. Amjad and M. Ilyas, "A Comparative Study of Global Software Development Tools Supporting Project Management Activities," *International Journal of Education and Management Engineering*, vol. 6, pp. 32-39, 2017.
- [15] N. Sehwat, N. Munsir and M. Jain, "Risk Management in Software Projects," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 10, pp. 845-849, 2014.

-
-
- [16] B. de Wet and J. K. Visser, "An Evaluation of Software Project Risk Management in South Africa," *South African Journal of Industrial Engineering*, vol. 24, no. 1, pp. 14-28, 2013.
 - [17] G. Arcidiacono, "Comparative research about high failure rate of IT projects and opportunities to improve," *PM World Journal*, vol. 6, no. 2, pp. 1-10, 2017.
 - [18] H. Abdul-Rahman, F. A. Mohd-Rahim and W. Chen, "Reducing failures in software development projects: effectiveness of risk mitigation strategies," *Journal of Risk Research*, vol. 15, no. 4, pp. 417-433, 2012.
 - [19] S. M. Neves and C. E. S. da Silva, "Risk management applied to software development projects in incubated technology-based companies: literature review, classification, and analysis," *Gestão & Produção*, vol. 23, no. 4, pp. 798-814, 2016.
 - [20] D. Pimchangthong and V. Boonjing, "Effects of Risk Management Practice on the Success of IT Project," *Procedia Engineering: 7th International Conference on Engineering, Project, and Production Management*, vol. 182, pp. 579-586, 2017.
 - [21] T. W. Kwan and H. K. N. Leung, "A Risk Management Methodology for Project Risk Dependencies," *IEEE Transactions on Software Engineering*, vol. 37, no. 5, pp. 635-648, 2011.
 - [22] B. G. Tavares, C. E. S. da Silva and A. D. de Souza, "Risk management analysis in Scrum software projects," *International Transactions in Operational Research*, pp. 1-22, 2017.
 - [23] O. Didraga, "The Role and the Effects of Risk Management in IT Projects Success," *Informatica Economică*, vol. 17, no. 1, pp. 86-98, 2013.
 - [24] S. Vahidnia, O. O. Tanriover and I. N. Askerzade, "An Early Phase Software Project Risk Assessment Support Method for Emergent Software Organizations," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 5, pp. 105-118, 2017.
 - [25] C. Lopez and J. L. Salmeron, "Risks Response Strategies for Supporting Practitioners Decision-Making in Software Projects," *Procedia Technology: Conference on ENTERprise Information Systems / HCIST 2012*, vol. 5, pp. 437-444, 2012.
 - [26] V. Bhoola, S. B. Hiremath and D. Mallik, "An Assessment of Risk Response Strategies Practiced in Software Projects," *Australasian Journal of Information Systems*, vol. 18, no. 3, pp. 161-191, 2014.
 - [27] V. A. Banuls, C. Lopez, M. Turoff and F. Tejedor, "Predicting the Impact of Multiple Risks on Project Performance: A Scenario-Based Approach," *Project Management Journal*, vol. 48, no. 5, pp. 95-115, 2017.
 - [28] F. S. Rahayu, T. Indrawan and S. Kamarudin, "Risk mitigation strategies in implementing scrum framework for internet-based IT companies in Indonesia," *Indonesian Journal of Information Systems*, vol. 3, no. 1, pp. 50-63, 2020.

- [29] S. K. B. Shiri, S. S. K. Teja and N. Ganesan, "Advance Tools and Techniques for Software Risk Management," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 3, pp. 456-459, 2016.
- [30] M. Khraiwesh, "Project Monitoring and Control Measures in CMMI," *International Journal of Computer Science & Information Technology*, vol. 5, no. 5, pp. 39-56, 2013.
- [31] S. Y. Chadli and A. Idri, "Identifying and mitigating risks of software project management in global software development," in *Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement*, Gothenburg, Sweden, 2017.
- [32] G. N. Aranda, A. Vizcaino and M. Piattini, "Analyzing and Evaluating the Main Factors that Challenge Global Software Development," *The Open Software Engineering Journal*, vol. 4, pp. 14-25, 2010.
- [33] M. Jimenez, A. Vizcaino and M. Piattini, "Improving Distributed Software Development in Small and Medium Enterprises," *The Open Software Engineering Journal*, vol. 4, pp. 26-37, 2010.
- [34] S. Haq, M. Raza, A. Zia and M. N. A. Khan, "Issues in Global Software Development: A Critical Review," *Journal of Software Engineering & Applications*, vol. 4, pp. 590-595, 2011.
- [35] M. Tihinen, Measurement-based management of global software development projects (PhD Thesis), Linnanmaa: VTT & University of Oulu, 2014.
- [36] E. O. Conchuir, P. J. Agerfalk, H. H. Olsson and B. Fitzgerald, "Global Software Development: Where are the benefits?," *Communication of the ACM*, vol. 52, no. 8, pp. 127-131, 2009.
- [37] R. Prikladnicki, J. L. N. Andy and R. Evaristo, "Global Software Development in Practice: Lessons Learned," *Software Process Improvement and Practice*, vol. 8, pp. 267-281, 2003.
- [38] A. Elzamly and B. Hussin, "Modelling and Evaluating Software Project Risks with Quantitative Analysis Techniques in Planning Software Development," *Journal of Computing and Information Technology*, vol. 2, pp. 123-139, 2015.
- [39] A. Lamersdorf and J. Munch, "Studying the Impact of Global Software Development Characteristics on Project Goals: A Causal Model," *The Open Software Engineering Journal (TOSEJ)*, vol. 4, pp. 2-13, 2010.
- [40] C. B. Widiyatmoko, Methodological Support for Task Coordination in Global Software Engineering Projects at Product Software Companies (Master Thesis), Utrecht: Utrecht University, 2017.
- [41] N. Prenner, C. Unger-Windeler and K. Schneider, "Goals and challenges in hybrid software development approaches," *Journal of Software: Evolution and Process*, vol. 33, no. 11, 2021.

- [42] M. C. Garrido, M. C. A. Ruotolo, R. F. M. L and H. A. Naked, "Risk identification techniques knowledge and application in the Brazilian construction," *Journal of Civil Engineering and Construction Technology*, vol. 2, no. 11, pp. 242-252, 2011.
- [43] R. K. Kansal and M. Sharma, "Risk Assessment Methods and Application in the Construction Projects," *International Journal of Modern Engineering Research*, vol. 2, no. 3, pp. 1081-1085, 2012.
- [44] C. C. Hsu and B. A. Sandford, "The Delphi Technique: Making Sense of Consensus," *Practical Assessment, Research & Evaluation*, vol. 12, no. 10, pp. 1-8, 2007.
- [45] G. J. Skulmoski, F. T. Hartman and J. Krahm, "The Delphi Method for Graduate Research," *Journal of Information Technology Education*, vol. 6, pp. 1-21, 2007.
- [46] P. J. Agerfalk, B. Fitzgerald, H. H. Olsson and E. O. Conchuir, "Benefits of Global Software Development: The Known and Unknown," in *Making Globally Distributed Software a Success Story*, Berlin, Springer-Verlag Berlin Heidelberg, 2008, pp. 1-9.
- [47] A. S. Khan and Z. Subhan, "Distributed Software Development Process, Initiatives and Key Factors: A Systematic Literature Review," *International Journal of Multidisciplinary Sciences and Engineering*, vol. 5, no. 12, pp. 7-21, 2014.

APPENDICES

Appendix A

Table A1. Compiled GSD Risks for the Delphi Technique

Risk Category	Risk ID	Risk
Financial risks	A1	Increase of running costs due to unforeseen factors in foreign countries
	A2	Added cost for coordination
	A3	Added travel costs
	A4	Increase of communication costs
	A5	Cost estimation difficulties due to distributed project locations
Operational and Planning risks	B1	Timetable differences affecting the synchronization of the work
	B2	Lack of coordination affecting task completion and quality
	B3	Lack of control due to different corporate practices between countries
	B4	Misused processes and lengthy problem-solving process
	B5	Lack of adequate documentation
	B6	Lack of awareness of critical tasks
	B7	Adhering to local regulations of foreign countries
	B8	Process dependency risk
	B9	Data transfer privacy and security
	B10	Political and country risks
	B11	Poor telecommunication infrastructure in foreign countries
	B12	Difference in holidays between countries
	B13	Uncoordinated task allocation
Management and Human resource risks	C1	Cultural differences affecting work environment
	C2	Differences in knowledge and expertise
	C3	Inability to acquire the necessary skills for the project
	C4	Language barrier
	C5	Lack of communication between teams in different locations
	C6	Decreased sense of competitiveness
	C7	Misunderstanding of individual and team responsibility
	C8	Difficulty in conveying motivation and project
	C9	Miscommunicated project goals and objectives
	C10	Issues due to differences in offshore contracts
	C11	Lack of group awareness
	C12	Multiple work standards
	C13	lack of understanding of management and authority
	C14	Reduction of productivity due to lack of team awareness
	C15	Lack of team management resulting from different practices
	C16	Lack of people management and conflict resolution in one or more location
	C17	Difficulty in creating team spirit
Communication, coordination and collaboration risks	D1	Communication inefficiencies
	D2	Lack of coordination due to distance
	D3	Collaboration issues due to asynchronized interaction
	D4	Weakened team cohesion due to distances (temporal, geographic and socio-cultural)
	D5	Lack of mutual understanding due to cultural difference
	D6	Difficulty in building social relations between the team members
	D7	Social isolation forming between team members
	D8	Difficulty in synchronized and face to face communication
	D9	Possibility of misunderstanding due to cultural differences
	D10	Limited common knowledge
	D11	Lack of trust
	D12	Lack of transparency
	D13	Difference in quality standards perception
	D14	Communication issues between offshore teams and client
	D15	Lack of trust between client and offshore teams
Technical risks	E1	Lack of time overlap affecting software configuration
	E2	Quality issues emerging from different quality standards and perceptions
	E3	Software compatibility issues
	E4	Software architecture mismatch
	E5	Distorted information
	E6	Technology incompatibility
	E7	Lack of knowledge about development models and approaches
	E8	Difficulties in applying international software development practices
	E9	Asymmetry in processes, policies and standards

Table A2. Delphi rounds summary for GSD risks (15 software development experts)

Risk Category	Risk ID	Round 1	Round 2			Round 3				Round Consensus	Consensus	Round 4
		Action	Relevance 5-point rating		Probability of Occurrence (P)		Impact (I)		Prioritized Treatment Strategy			
			Highest Cluster Rating	Percent	Round Consensus	Highest Cluster	Percent	Highest Cluster				Percent
Financial risks	A1	Retain	4	53.3	✓	50% to 75%	66.7	Medium	60.0	P&I	✓	Accept
	A2	Retain	4	40.0	✗	50% to 75%	40.0	Medium	40.0	✗	✗	Reduce
	A3	Retain	3	46.7	✗	50% to 75%	46.7	Low	53.3	I	✗	Reduce
	A4	Retain	5	53.3	✓	25% to 50%	46.7	Medium	53.3	I	✗	Accept
	A5	Delete										
Operational and Planning risks	B1	Retain	5	60.0	✓	0% to 25%	40.0	Low	40.0	✗	✗	Reduce
	B2	Retain	5	53.3	✓	50% to 75%	46.7	High	40.0	✗	✗	Reduce
	B3	Retain	4 & 5	33.3	✗	50% to 75%	40.0	Medium	46.7	✗	✗	Transfer
	B4	Delete										
	B5	Delete										
	B6	Retain	5	53.3	✓	75% to 100%	60.0	High	66.7	P&I	✓	Avoid
	B7	Retain	3	40.0	✗	50% to 75%	53.3	Medium	46.7	P	✗	Transfer
	B8	Delete										
	B9	Delete										
	B10	Delete										
	B11	Retain	5	66.7	✓	50% to 75%	53.3	High	53.3	P&I	✓	Transfer
	B12	Delete										
	B13	Delete										
Management and Human resource risks	C1	Retain	4	33.3	✗	25% to 50%	60.0	Medium	46.7	P	✗	Reduce
	C2	Retain	5	40.0	✗	25% to 50%	46.7	High	46.7	✗	✗	Accept
	C3	Retain	5	66.7	✓	75% to 100%	53.3	Medium	53.3	P&I	✓	Transfer
	C4	Delete										
	C5	Retain	5	60.0	✓	75% to 100%	66.7	Medium	46.7	P	✗	Avoid
	C6	Retain	4	46.7	✗	25% to 50%	46.7	Medium	60.0	I	✗	Accept
	C7	Delete										
	C8	Retain	4	46.7	✗	25% to 50%	53.3	Medium	46.7	P	✗	Accept
	C9	Retain	5	66.7	✓	50% to 75%	60.0	High	53.3	P&I	✓	Avoid
	C10	Retain	4 & 5	33.3	✗	50% to 75%	40.0	Medium	46.7	✗	✗	Avoid
	C11	Retain	5	53.3	✓	50% to 75%	46.7	Low	40.0	✗	✗	Accept
	C12	Delete										
	C13	Delete										
	C14	Retain	5	60.0	✓	50% to 75%	53.3	High	66.7	P&I	✓	Reduce
	C15	Retain	5	46.7	✗	25% to 50%	46.7	Medium	53.3	I	✗	Reduce
	C16	Retain	4 & 5	33.3	✗	50% to 75%	40.0	Low	46.7	✗	✗	Reduce
	C17	Delete										
Communication, coordination and collaboration risks	D1	Retain	5	66.7	✓	75% to 100%	60.0	High	66.7	P&I	✓	Reduce
	D2	Retain	3	33.3	✗	0% to 25%	40.0	Medium	60.0	I	✗	Reduce
	D3	Retain	3 & 4	33.3	✗	0% to 25%	53.3	Medium	40.0	P	✗	Transfer
	D4	Retain	4 & 5	33.3	✗	50% to 75%	60.0	Medium	46.7	P	✗	Reduce
	D5	Retain	4	33.3	✗	50% to 75%	46.7	Medium	66.7	I	✗	Reduce
	D6	Retain	4	46.7	✗	25% to 50%	40.0	Medium	66.7	I	✗	Accept
	D7	Retain	4 & 5	33.3	✗	25% to 50%	40.0	Medium	46.7	✗	✗	Accept
	D8	Retain	5	40.0	✗	50% to 75%	46.7	Medium	53.3	I	✗	Accept
	D9	Delete										
	D10	Retain	3 & 5	33.3	✗	50% to 75%	53.3	Medium	40.0	P	✗	Accept
	D11	Retain	5	66.7	✓	75% to 100%	60.0	High	66.7	P&I	✓	Avoid
	D12	Retain	5	53.3	✓	50% to 75%	46.7	High	53.3	I	✗	Avoid
	D13	Retain	3 & 5	33.3	✗	50% to 75%	60.0	Medium	46.7	P	✗	Transfer
	D14	Retain	5	53.3	✓	25% to 50%	46.7	Medium	60.0	I	✗	Reduce
	D15	Retain	5	53.3	✓	50% to 75%	40.0	Medium	53.3	I	✗	Avoid
Technical risks	E1	Retain	4	46.7	✗	25% to 50%	40.0	Medium	53.3	I	✗	Reduce
	E2	Retain	4	53.3	✓	50% to 75%	53.3	Medium	60.0	P&I	✓	Avoid
	E3	Retain	4	53.3	✓	50% to 75%	53.3	High	66.7	P&I	✓	Accept
	E4	Retain	4 & 5	33.3	✗	50% to 75%	60.0	Medium	46.7	P	✗	Transfer
	E5	Delete										
	E6	Delete										
	E7	Retain	4	53.3	✓	50% to 75%	53.3	Medium	46.7	P	✗	Avoid
	E8	Retain	4	40.0	✗	50% to 75%	40.0	Medium	53.3	I	✗	Avoid
	E9	Retain	4 & 5	33.3	✗	50% to 75%	46.7	Medium	60.0	I	✗	Reduce

المخاطر ذات الأولوية واستراتيجيات العلاج في تطوير البرمجيات العالمية

محمد الحصني^{1*}، أيمن مصباح²، طارق الحصادي³

¹ جامعة درنة، ليبيا، mohamed.hasni@omu.edu.ly

² كلية العلوم التقنية، درنة، ليبيا، aimn_gramay@yahoo.com

³ كلية العلوم التقنية، درنة، ليبيا، tarek.h.elhasady@hotmail.com

الملخص

تجري الدراسة مراجعة للأدبيات من أجل تجميع المخاطر التي ترتبط ارتباطاً مباشراً باستراتيجية تطوير البرمجيات العالمية. تتبنى دراسة الحالة منهجية دلفي التي تسمح للباحث بالتوصل إلى إجماع حول أكثر المخاطر ذات الصلة والدرجة المرتبطة بمشروع تطوير البرمجيات العالمية. شارك في الدراسة عشرون خبيراً في تطوير البرمجيات من جميع أنحاء العالم، مع ما لا يقل عن خمسة عشر خبيراً في كل جولة من جولات دلفي. تم تصميم الجولات الأربع لطريقة دلفي المستخدمة في هذه الدراسة للحصول على إجماع حول المخاطر الأكثر أهمية لاستراتيجية تطوير البرمجيات العالمية، بالإضافة إلى إجراء تقييم للمخاطر لجميع المخاطر التي تم تجميعها من المؤلفات والتحقق منها من قبل الخبراء المشاركين. تظهر النتائج أن هناك عشرة مخاطر رئيسية حظيت بالإجماع من قبل خبراء تطوير البرمجيات وتحتاج إلى معالجتها كأولوية في مشاريع تطوير البرمجيات العالمية، حيث يكون لكل منهم احتمالية متوسطة إلى عالية لحدوثها وتأثيرها على نجاح مشروع البرمجيات باستخدام استراتيجية تطوير البرمجيات العالمية.

الكلمات الدالة:

تطوير البرمجيات العالمية.

إدارة المخاطر.

تقنية دلفي.

* البريد الإلكتروني للباحث المراسل: mohamed.hasni@omu.edu.ly