BMIJ
ISSN: 2148-2586

bmij (2021) 9 (2):456-471
doi: https://doi.org/10.15295/bmij.v9i2.1782

Research Article

Relationship between requirements analyzability, interpersonal trust, and software project performance: Moderating role of customization

Yazılım proje performansı, ihtiyaçların analiz edilebilirliği ve kişiler arası güven ilişkisi: Yazılımın kişiselleştirme derecesinin düzenleyici rolü

Dilek Erdoğan[1]

Tuğba Gürçaylılar Yenidoğan[2]

[1] Asst. Prof., Gaziantep University, Gaziantep, Turkey,

dilekerdogan@gantep.edu.tr

ORCID: 0000-0002-2079-777X

[2] Prof. Dr., Akdeniz University, Antalya, Turkey, gurcaylilar@akdeniz.edu.tr

ORCID: 0000-0002-4836-6904

Corresponding Author:

Dilek Erdoğan,

Gaziantep University, Gazintep, Turkey, dilekerdogan@gantep.edu.tr

Submitted: 14/02/2021

Revised: 5/04/2021

Accepted: 14/04/2021

Online Published: 25/06/2021

Citation: Erdoğan, D., & Gürçaylılar Yenidoğan, T., Relationship between requirements analyzability, interpersonal trust, and software project performance: Moderating role of customization, bmij (2021) 9 (2): 456-471, doi: https://doi.org/10.15295/bmij.v9i2.1782

Abstract

Based on the information processing perspective, this study empirically investigated the effects of requirements analyzability, an indicator of technological complexity and interpersonal trust as the soft factor of buyer-supplier relationships on software project development performance. Beyond that, the possible moderator role of the degree of software customization was examined on the analyzability-interpersonal trust and analyzability-performance links. Data were collected from 138 companies through a questionnaire-based survey of IT purchasing managers in Turkey. Partial Least Squares Structural Equation Modeling was used to test the research hypotheses. This study indicated that requirements analyzability and interpersonal trust have a significant effect on software performance. Overall, the study provides evidence that interpersonal trust mediates the relationship between requirements analyzability and software project performance. The research findings support the moderation role of customization in the relationship between requirements analyzability and software project performance. However, partial support is provided for the moderating effect of customization on the relationship between requirements analyzability and interpersonal trust.

Keywords: Software Development Project, Information Processing Theory, Task Complexity

Jel Codes: M15, L86, D89

Öz

Bilgi işleme perspektifinden hareketle, bu çalışmada teknolojik karmaşıklığın bir göstergesi olan ihtiyaçların analiz edilebilirliği ve alıcı-tedarikçi ilişkilerinin yumuşak faktörü olarak kişiler arası güvenin yazılım proje performansı üzerindeki etkileri ampirik olarak araştırılmıştır. Çalışmada ayrıca yazılımın kişiselleştirme derecesinin analiz edilebilirlik-kişiler arası güven ve analiz edilebilirlik-performans bağlantılarında olası bir moderatör rolü olup olmadığı sınanmıştır. Araştırmanın verileri, anket tekniği kullanılarak, Türkiye'de faaliyet gösteren 138 alıcı işletmenin bilgi teknolojileri yöneticilerinden toplanmıştır. Araştırmanın hipotezlerini test etmek için Kısmi En Küçük Kareler Yapısal Eşitlik Modellemesi kullanılmıştır. Araştırmanın sonuçları ihtiyaçların analiz edilebilirliği ve kişiler arası güvenin yazılım proje performansı üzerinde istatistiksel olarak anlamlı bir etkiye sahip olduğunu göstermiştir. Genel olarak, bu çalışma kişiler arası güvenin ihtiyaçların analiz edilebilirliği ve yazılım proje performansı arasındaki ilişkide aracı rolü olduğuna dair ampirik kanıtlar sağlamaktadır. Araştırmanın bulguları, ihtiyaçların analiz edilebilirliği ve yazılım proje performansı arasındaki ilişkide yazılımın kişiselleştirme derecesinin düzenleyici rolü olduğunu desteklemektedir. Bununla birlikte, ihtiyaçların analiz edilebilirliği ve kişiler arası güven ilişkisinde kişiselleştirme derecesinin düzenleyici etkisi kısmen desteklenmektedir.

Anahtar Kelimeler: Yazılım Geliştirme Projesi, Bilgi İşleme Teorisi, Teknolojik Karmaşıklık

JEL Kodları: M15, L86, D89

Introduction

IT infrastructure contains all the business software to meet the information processing needs of end-users. The software is composed of tightly coupled, independently managed elements, namely system of systems. Software development refers to an uncertain process with highly complex and unpredictable activities towards achieving the task. Despite all the difficulties and challenges faced in the development process, the software is one of the most commonly outsourced components of IT services (Wang, 2002). Business organizations usually apply to software outsourcing in two ways: they buy commercial off-the-shelf (COTS) software solutions or build a project-based relationship with software providers. A COTS software package offers standardized plug-in solutions that fulfil the basic needs of business operating systems. COTS-based configuration prevents buyers from making -small or large but not least non-systemic, complete changes during future activities and downstream tasks. As well the structural rigidity against unspecified changes, this kind of software outsourcing may display inconsistency with business operations. As an alternative, project-based bespoke software can seem like a more usable or even cheaper choice to meet user requirements. However, a range of information processing and adaptation problems would inevitably arise as no ex-ante analysis to the software and services can be relied upon entirely where project team members perform ambiguous and uncertain tasks. The longer the problems in coordinating software design changes remain unsolved, the greater the impact on project cost, schedule, and quality (Susarla et al., 2010; Wang, 2002). Indeed, failure rates of bespoke software solutions are cited as high as 80 per cent or sometimes higher in practice (Standish Group International, 2014). It is stated that only one in every eight IT projects can be completed successfully to satisfy users' needs within the budget or schedule (McManus & Wood-Harper, 2007). All in all, the market researchers emphasize that business organizations are experienced numerous serious failures of software projects. As more and more organizations invest substantial resources in software development, significant causes of project failure become more crucial and inspire more research attention on software project management.

Factors impacting software project success are frequently associated with incomplete and inconsistent formal specifications (Liu et al., 2011). Incomplete software requirements specification arises from the characteristics of requirements, including requirements diversity, requirements instability, and requirements analyzability (Nidumolu, 1996). These are strongly linked to the task environment. Uncertain and complex requirements entail making substantial adjustments in the software procurement process. Non-adjusting requirements lead to poor performance in two different ways: governance risks vs information processing capabilities. According to a risk-based approach to information processing (Chen et al., 2004), project failure is mainly due to the governance risks (i.e., hold-ups in performing desirable efforts and dynamic tasks) that restrain from involving observable and verifiable behaviour performance measurement.

On the other hand, a heterogeneous and ambiguous description of user needs (i.e. requirements uncertainty) increases misaligned incentives on effort and task performance. In turn, increasing opportunistic propensity undermines software performance (Chen & Paulraj, 2004; Gefen et al., 2008). Contrarily to requirements uncertainty, a task description in clear and understandable terms diminishes the performance measurement risk, leaving a narrower space for opportunism (Barthelemy & Quelin, 2006; Qi & Chau, 2015). Notably, a more precise and unambiguous formal specification prevents conflicts arising from requirements instability (Chen et al., 2004; Lioliou et al., 2014) and functions as a safeguard mechanism against opportunism.

At this point, it is worthy of note that incompleteness can be high even under low requirements uncertainty. Even if an initial set of user requirements is remarkably well-documented, errors and exceptions remain a matter when compiling the workflow. Deviations in software for detailed requirements appear in a set of confusing transformations. Change is the only constant in performing the task of developing and executing the software. Change requests for the project task, including designing, implementing, budgeting, and scheduling, impose some additional requirements and new activity links that a project must fulfil. Accordingly, user needs can never be reduced to a standardized protocol, yet potential problems can be overcome by predetermined analytic procedures, namely requirements analyzability (Jaffee, 2001; Perrow, 1967). In the presence of a known procedure for performing the task, analyzability prevents being stuck in the maladaptive information processing patterns (Bensaou & Anderson, 1999). A proper analysis of requirements enables adjustments and maintenance enhancements to be made. Then, compatible changes within and between information systems lead to better project performance. Contrarily to analyzability, non-analyzability results in poor performance (Bensaou & Venkatraman, 1996), but this time is due to a poor-fit between the technology capability and the requirements of the business process (Premkumar et al., 2005), not the performance measurement risk. Management by routines for gathering and processing information to link user needs (Galbraith, 1974) allows the atmosphere to be completed successfully without formal rigidities to monitor performance. When a project is highly analyzable, project parties respect each other's expertise, supporting the development of empathy between boundary-spanning personnel. After the fact, they can constitute a team communicating better. Good communication facilitates problem-solving and enhances the effectiveness of business process analysis. At last, behavioural trust (Nooteboom, 2014) is built-in developers' competencies for software support and maintenance. When trust deepens, buyers believe that software providers have sufficient competence to carry out the dynamic responsibilities and perform the best.

As the literature above reveals, analyzability and interpersonal trust play an essential role in project success. However, very little attention has been paid to the direct effect of requirements analyzability on software performance. Addressing this gap, we here examine the effect of analyzability on project performance, especially considering that interpersonal trust is playing an essential role as a mediating element in the relationship between requirements analyzability and software project development performance. This research contributes to the project management literature in two different ways. First, the study offers theoretical support that requirements analyzability is crucial for project performance. In previous studies, analyzability has been considered to be a sub-dimension of requirements uncertainty that undermines software project performance (Nidumolu, 1996; Moynihan, 2000). However, analyzability functions alone to adjust the project scope and create transaction value instead of risk remedies. Non-analyzability increases technical complexity at best, not internal uncertainty. This study adds more insights to the information processing view by highlighting the importance of requirements analyzability on project performance. Secondly, this research confirms the mediating role of interpersonal trust in the relationship between requirements analyzability and project performance. A limited number of previous studies have already investigated the effect of interpersonal trust on software project performance (Qi & Chau, 2013). Despite all of these, project management deals with either the calculative trust or considering it in managing requirements uncertainty at the inter-organizational level (Dyer & Singh, 1998; Zaheer et al., 1998). Differently, we adopt the capability perspective of project management and benefit from the competence-based interpersonal trust that mediates the relationship between requirements analyzability and project development performance. This kind of endeavour is the first in the project-based software development literature to the best of our knowledge. By doing so, we hope to contribute to the literature.

Literature review and research hypotheses

Several criteria are used to evaluate project success in software development, such as scope, time, cost, quality, satisfaction, etc. (Deephouse et al., 1995; Wang et al., 2006). This study focuses on the two areas of performance to determine the success or failure of a software project: product performance and process performance (Nidomolu, 1995; Wallace et al., 2004). On the one hand, product performance is described as the extent of success of a system developed. Performance assessment of a product is conducted to evaluate how the final delivered software works. On the other hand, process performance regards the quality of the software project development process itself. The performance of a product development process denotes to what extent a software project is delivered on schedule and within the allocated budget. Software development activities are of paramount importance for product quality (Arpaci, 2017; Henderson & Lee, 1992). This means that a software project is considered successful if a supplier delivers the final product by agreed-upon commitments on quality and budget within the delivery timetable.

This study clarifies the relationship between project performance and requirements specification by providing a well-crystallized understanding of how requirements analyzability influences project performance. In this study, we follow up an extensive literature claiming the performance is diminishing effect of inadequacy and incompleteness of requirements specification (Schmidt et al., 2001), a legally binding document but proceeding differently. As stated in the introductory section, we take here an account of the information processing perspective. Hence, we distinguish requirements uncertainty from the technological complexity, unlike the previous literature on software project management (see Figure 1). More importantly, we attribute performance failure to the two different conditions of inadequacies in requirements specification: misaligned incentives and incongruent transformations.

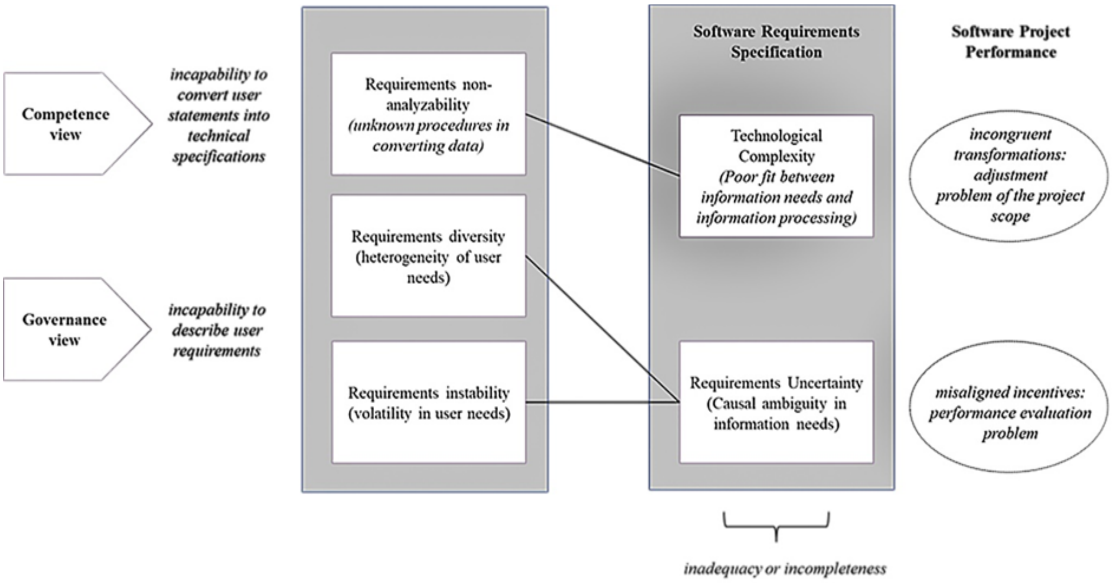
Competence view
incapability to convert user statements into technical specifications
Governance view
incapability to describe user requirements
Requirements non-analyzability (unknown procedures in converting data)
Requirements diversity (heterogeneity of user needs)
Requirements instability (volatility in user needs)
Software Requirements Specification
Technological Complexity (Poor fit between information needs and information processing)
Requirements Uncertainty (Causal ambiguity in information needs)
Software Project Performance
incongruent transformations: adjustment problem of the project scope
misaligned incentives: performance evaluation problem
inadequacy or incompleteness

Figure 1: A Conceptual Approach to Requirements Specification for Software Project Failure

Source: (developed by authors).

Requirements analyzability and software project performance

A research agenda of software development emphasizes the importance of determining the users' needs for project success. Therefore, requirements uncertainty has become one of the most visited conceptual phenomena (Nidumolu, 1996; Moynihan, 2000). The coexistence of requirements diversity with the instability of system requirements constitutes internal uncertainty that induces vulnerability to incomplete information. Although requirements analyzability was previously defined as a source of requirements uncertainty (Nidumolu, 1995, 1996), there exists an increasing number of studies (Chen et al., 2004; Jiang, 2009; Liu et al., 2011; Zowghi & Nurmuliani, 2002) that do not deliberately incorporate analyzability in the measurement model of requirements uncertainty. Consistently, this study regards requirements analyzability as an indicator of technological complexity, focusing further on the project performance.

Requirements analyzability is a high priority in software projects due to the sequential spread of failure and software development (Turner, 1992). Analyzability denotes the extent to which mechanical steps and analytic procedures for converting user needs into requirements specification can be designated in advance (Nidumolu, 1996). A higher analyzability enables adaptive task adjustments to correct errors and system performance deviations that cause damage to software products. As analyzability decreases, it will be tough to produce a maintainable software product within the allocated time and budget. Analyzability is about a technological competence that affects software performance throughout its life cycle. Uncertainty in software features poses a significant challenge for software professionals. Undoubtedly, software developers will be more productive when a set of specific requirements can be specified before the coding (Cusumano et al., 2003; Srinavasan, 2009). However, it is much more important to restore the performance where change is inevitable. At this point, software analyzability facilitates adaptive responses to changes and then improves performance. Based on these arguments, we derive the following hypothesis:

H_1: *As requirements analyzability increases, software project performance increases.*

Interpersonal trust and software project performance

According to social exchange theory (Anderson, 1995), trust functions as a social norm between buyer and seller. The two themes of trust are stated in a buyer-supplier dyad relationship (Oza et al., 2006). The first is an institutional- or system-based trust that characterizes attitudes towards organizations. The second is an interpersonal trust which characterizes a relationship between two individuals. Zaheer et al. (1998) revealed that inter-organizational trust plays a more dominant role than interpersonal trust in buyer-supplier relationships. This is an expected contribution when a more transaction-oriented pattern of exchange relations is followed by which trust involves buyer expectations on the delivery of high-quality services.

On the contrary, Qi and Chau (2013) engaged specifically with a team-based collaboration and emphasized the importance of interpersonal trust in project performance. As software development requires close personal ties between users and developers, interpersonal trust exerts a more profound effect on project success. Similarly, Kern and Willcocks (2002) found that interpersonal trust between key persons in IT outsourcing affects the process and outcome of inter-organizational relationships. Furthermore, Wade and Hulland (2004) demonstrated that relationship-specific entities (including interpersonal trust) encourage cooperative behaviour that creates competitive advantages for both parties. Finally, Paul and McDaniel (2004) provided further support for a robust causal effect between interpersonal trust and collaborative relationship performance.

In software projects, interpersonal trust performs at the level of individual boundary spanners (Zucker, 1987) and is defined as the faith in customer contact people to work in the buyer's best interest (Lau & Rowlinson, 2009, p. 543). One or several people occupying boundary-spanning positions interact the most in software development (Miranda & Kavan, 2005). On the buyer side, boundary-spanning people consist of IT managers, IT team leaders, or IT professionals, while on the supplier side, they are usually software developers. Trust depends on confidence in suppliers' credibility, integrity, and benevolence during the software development life cycle (Cao & Lumineau, 2015). Users (buyer) will likely "feel a sense of betrayal" (Zaheer et al., 1998, p.148) if the vendor's performance is below the user's expectations. A boundary-spanning role helps surface the conflict without disruptive consequence at a higher level of interpersonal trust (Zaheer et al., 1998). Overall, interpersonal trust facilitates conflict resolution and encourages a collaborative orientation to adapt to changing environmental conditions (Carson et al., 2006; Zaheer et al., 1998). There is high trust between the parties in a software development relationship. System requirements can be systematically adjusted to correct faults without costly pre-documentation investments, leading to better performance (Oza et al., 2006). Based on these explanations, we derive the following hypothesis:

H_2: *As interpersonal trust increases between boundary spanners, software project performance increases.*

Requirements analyzability and interpersonal trust

The requirements analysis is a social process where there are close working relationships between users and software developers. It requires intense cooperation between end-users and software developers to determine the project requirements wholly and accurately. At this stage, if interpersonal communication and information sharing increase and hence a strong collaboration establishes between the parties, a positive team atmosphere breeds trust between individuals (Reyes et al., 2008). However, users and software developers will have different viewpoints if requirements and tasks are low in analyzability. This situation provokes task-based conflicts due to unclear requirements in the implementation phase of the project. At the same time, low analyzability in requirements results in performance fluctuations during the project. All of which may cause damages to the project climate and mistrust between end-users and software developers (Liu et al., 2011). Based on these explanations, we derive the following hypothesis:

H_3: *As requirements analyzability increases, interpersonal trust increases between boundary spanners.*

The research model includes potential mediation effects. For example, interpersonal trust may serve to mediate the impact of requirements analyzability on software performance. We suggest below hypothesis:

H_4: *Interpersonal trust mediates the relationship between requirements analyzability and project performance.*

Moderating effect of software customization

Customization refers to software adaptation for a particular user (organization or business unit) and differs from off-the-shelf software packages designed for the mass market (Dittrich et al., 2009). The presence of tacit knowledge in customized software projects increases project costs. Furthermore, it is very troublesome to reveal a clear and consistent design for a customized software because customization often relies upon undocumented assumptions that necessitate the development of relationship-specific knowledge with the technical experience of software developers. Therefore, a correct and precise analysis of the requirements is a more critical issue in software projects with a high degree of customization.

A higher degree of software customization increases task uncertainty. In this case, requirements analyzability becomes a more difficult task due to the likelihood of misaligned incentives under internal uncertainty (Parthasarathy & Daneva, 2016). Furthermore, when requirements are inevitably subject to process adjustments, especially in more customized projects, the supplier's technical competence in designating an accurate analysis breeds interpersonal trust, different from calculative controls of a violation. As a result, requirements analyzability and interpersonal trust in more customized projects may exert a more decisive influence over project performance. Accordingly, we derive the following hypotheses:

***H$_{5a}$:** As the customization degree of software increases, requirements analyzability will have a stronger effect on project performance.*

***H$_{5b}$:** As the customization degree of software increases, requirements analysis will have a stronger effect on interpersonal trust.*

Based on the explanations, the research model of the study is presented as follows:

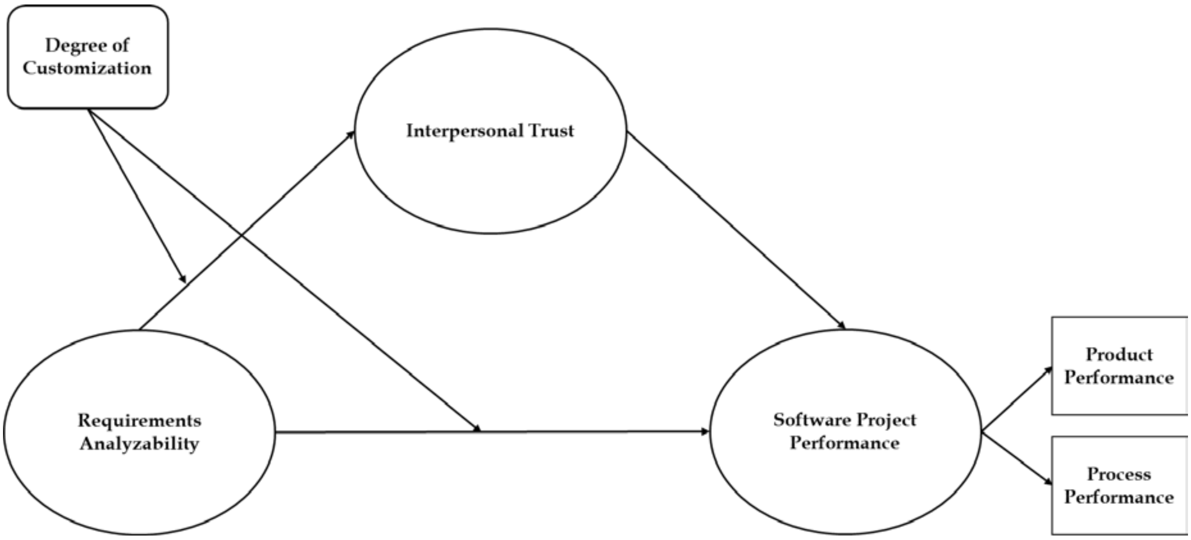

Figure 2: The Research Model

Research methodology

Data collection and sample

The data of this research were collected using an internet-based survey between the dates of 15.05.2019-24.07.2019. The critical informant methodology was used to ensure that IT managers with enough participation knowledge are included in the survey. Mid and senior IT executives were chosen as critical informants as they have completed IT projects. The surveys were sent to a selective sample of IT experts, IT managers, IT directors, CIOs, IT project and team leaders, and IT Chiefs who work in different companies in Turkey through LinkedIn, a professional communication network. On the cover page of the questionnaire, the participants were asked for answering all the questions by considering a particular software project they have actively involved in. The survey tool was piloted with 10 IT professionals from customers and service providers. According to the feedback of participants, the questionnaire was carefully modified to have a more understandable version. Since the data of this study had been collected before January 1, 2020, an ethics committee report was not received.

Table 1: Profile of Software Projects (N=138)

Type of Software Project	F	%
New Development	79	57,3
Maintenance	33	23,9
Reengineering	26	18,8
Total	138	100

Price Structure of Contract	F	%
Fixed-price	82	59,4
Time-material	56	40,6
Total	138	100,0

Contract Time	F	%
Definite time	60	43,5
Indefinite time	78	56,5
Total	138	100,0

Nationality of Supplier	F	%
Domestic	113	81,9
Foreign	25	18,1
Total	138	100

	Mean	Std. dev.
Relation Duration	4,57	3,772

Of the 600 questionnaires, 138 usable responses were obtained, and analyses were carried out on 138 questionnaire data. Twenty-seven participants work as Chief Information Officer (CIO), 34 participants as IT Director, 64 participants as IT Manager, 7 participants as IT Project Leader, 6 participants as IT Chief. The sector profile of client organizations is mainly distributed across manufacturing (17.4%), wholesale and retail (16.7%), information and communication (14.5%), banking and insurance (10,9%), and tourism (5.1%). The profiles for software projects within the scope of the research are given in Table 1.

Measures

In this study, survey scales were adopted from prior studies, and some items were adjusted to fit the context of software projects. All perceptual items were measured with seven-point Likert scales ranging from "strongly disagree" to "strongly agree". Citing references for the constructs, all the measurement items are listed in Table 2. The requirements analyzability refers to the process of reducing the user's needs to a set of requirements lists, mechanical steps, or objective procedures (Nidumolu, 1996). Requirement analyzability is the first order reflective construct. This construct was measured with four items adapted from Nidumolu's (1996) study. Interpersonal trust (on the buyer side) indicates to what extent the employee(s) of the buyer perceives the supplier's employee(s) as honest and benevolent (Ashnai et al., 2013). Interpersonal trust as the first-order reflective construct was measured with six items adapted from Ashnai et al. (2013) and Kumar et al. (1995).

Project performance is a second-order formative construct derived by two first-order reflective constructs, including product performance and process performance. While software product performance refers to the success of the developed system, process performance refers to the success of the development process (Nidumolu, 1996). Product- and process-performance were measured with items adapted from Wallace (2004). The customization degree of software is used as a moderator variable. Customization is software development dedicated to the specific needs of the user. An increased degree of customization means making the software features more specific to the buyer's needs. The single item scale of customization degree was measured on a 7-point Likert scale ranging from low to high.

Table 2: Construct Measurements (N=138)

Construct	Measurements	References
Requirements Analyzability	RA1-There was a clearly known way to convert user needs to requirements specifications. RA2-Available knowledge was of great help in converting user needs to requirement specifications. RA3-Established procedures and practices could be relied upon to generate requirements specifications. RA4-An understandable sequence of steps could be followed for converting user needs to requirements specifications.	Nidumolu, 1996
Interpersonal Trust	IPT1-Whenever the people we deal with at this supplier give us advice on our business operations, we know that they are sharing their best judgment. IPT2- We can count on the people we deal with at this supplier to be sincere. IPT3-Though circumstances change, we believe the people we deal with at this supplier will be ready and willing to offer assistance and support. IPT4-When making important decisions, the people we deal with at this supplier are concerned about our welfare. IPT5-When we share our problems with the people we deal with at this supplier, we know that they will respond with understanding. IPT6-When it comes to things that are important to us, we can depend on the support of the people that we deal with at this supplier.	Ashnai et al. (2013) and Kumar et al. (1995).
Product Performance	PP1-The application developed is reliable. PP2-The application is easy to maintain. PP3-The users perceive that the system meets intended functional requirements. PP4-The system meets user expectations with respect to response time. PP5-The overall quality of the developed application is high.	Wallace, 2004
Process performance	PRC1- The system was completed within budget. PRC2-The system was completed within schedule.	Wallace, 2004

Data analyses and results

The Structural Equation Modeling (SEM) technique through Partial Least Squares (PLS-SEM) of the SmartPLS 3 software (Ringle et al., 2015) was conducted to test the measurement model and the research hypotheses. PLS-SEM analysis is widely used due to the robustness of the test statistics to nonnormality and small sample sizes. Besides, PLS enables employing reflective and formative constructs in the same model (Ringle et al., 2012). Although PLS does not require a strict assumption of the normal distribution, the skewness and kurtosis values of all items for this study vary between (-1.5) and (+1.5), indicating a normal distribution of data (Hair et al., 2014). As a result, PLS-SEM was purposefully used in this research employing reflective and formative variables together with relatively small sample size. The measurement model was first assessed in the data analysis phase, and then the structural model was tested. Following this, mediation analysis was conducted for a probable mediating effect. Then, the moderator effects were tested through the SPSS process macro.

Assessment of measurement model

CFA (confirmatory factor analysis) was run to evaluate the validity and reliability of the measurements. The results are presented in Table 3. Cronbach's alpha values are above 0.70, confirming the reliability of the scales. Composite reliability for each construct is above the threshold value of 0.70. (Bagozzi & Yi, 1988). The convergent validity was evaluated by examining whether the factor loadings were above 0.5 and whether the average variance (AVE) extracted for each construct was found at least 0.5 (Fornell & Larcker, 1981). The items with low factor loadings were excluded from the constructs (PP2, PP3, IPT3). With the exclusion of these items, the AVE for each construct is greater than 0.50, and all items' loadings are higher than 0.70 (Chin, 1998).

Table 3: Results of CFA

Construct	Item	Factor Loading	STDDEV	T Stat.	p	CR	AVE	(α)
Requirements Analyzability	RA1	0,858	0,031	27,902	0,00	0,880	0,649	0,818
	RA2	0,737	0,053	13,953	0,00			
	RA3	0,768	0,052	14,657	0,00			
	RA4	0,852	0,032	26,771	0,00			
Interpersonal Trust	IPT1	0,799	0,040	19,827	0,00	0,946	0,777	0,927
	IPT3	0,849	0,041	20,886	0,00			
	IPT4	0,920	0,017	52,887	0,00			
	IPT5	0,940	0,013	72,250	0,00			
	IPT6	0,893	0,024	36,712	0,00			
Product Performance	PP1	0,843	0,033	25,732	0,00	0,901	0,752	0,835
	PP4	0,868	0,025	34,673	0,00			
	PP5	0,890	0,024	37,031	0,00			
Process Performance	PRC1	0,901	0,020	45,605	0,00	0,902	0,822	0,783
	PRC2	0,912	0,015	60,813	0,00			

To enable discriminant validity, the square roots of the AVE calculated for each variable must be more significant than the values of the correlation of the construct with other constructs (Fornell and Larcker, 1981). Table 4 shows that the square root of AVE for each variable was more significant than its correlation with other constructs. Furthermore, the heterotrait-monotrait ratio (HTMT) was tested for all latent variables to prove discriminant validity. In Table 5, it is seen that the correlations were below the heterotrait-monotrait ratio (HTMT) threshold of 0.9 (Hair et al., 2016).

Table 4: Construct Correlations and the Squared Roots of AVE

	Requirements Analyzability	Process Performance	Interpersonal Trust	Product Performance
Requirements Analyzability	(0,806)			
Process Performance	0,495	(0,907)		
Interpersonal Trust	0,414	0,550	(0,882)	
Product Performance	0,447	0,567	0,684	(0,867)

Table 5: Heterotrait-Monotrait Ratio (HTMT)

	Requirements Analyzability	Process Performance	Interpersonal Trust
Process Performance	0,619		
Interpersonal Trust	0,474	0,643	
Product Performance	0,535	0,700	0,772

Survey-based data collection approaches can reveal common method bias problems even when the discriminant validity is satisfactory. According to Kock (2015), for PLS-SEM, a VIF occurrence greater than 3.3 is an indicator of pathological collinearity and an indicator of a model's common method bias. VIF values were calculated for all variables in this study, and VIF values were found below 3.3 (between 1.410-2.133). This finding shows that common method bias is not a possible contaminant of the results.

Structural model

After ensuring that the measurement model is valid and reliable, the structural model was tested in three stages. First, the second-order formative structure was evaluated, then the H1, H2, and H3 hypotheses were tested, and finally, the values of R^2 and f^2 were checked. Finally, a bootstrapping test was performed with 5000 subsamples to determine the statistical significance of the parameter estimates.

In the information technology literature, some studies modeled software performance as a reflective construct. However, according to the guidelines set by Diamantopoulos and Winklhofer (2001), it is more appropriate to model software performance as a formative construct. Software performance is a second-order formative construct derived from product- and process- performance. It is known that the criteria used to evaluate reflective construct are not valid for formative construct (Diamantopoulos & Winklhofer, 2001). The path weights of the sub-constructs were employed to assess the measurement model of the second-order formative variable. It is seen in Table 6 that all paths are significant at the 0.001 level. Another criterion for evaluating the formative measurement model is to reveal whether there is multicollinearity among subcomponents. When the variance inflation factor (VIF) values of the two sub-constructs for the software performance were examined, it was seen that these values ranged from 1,746 to 2,133. These values are lower than the recommended threshold of 10 (Hair et al., 20

Table 6: Structural Statistics of the Second-Order Construct (Software Performance)

	Weights	(STDEV)	*T* Statistics	*P* Values
Process Performance -> Software Performance	0,459	0,020	22,781	0,000
Product Performance -> Software Performance	0,663	0,023	28,405	0,000

After confirming the structure of second-order constructs, the hypotheses (H1, H2, and H3) were tested. Table 7 presents the results of path coefficients. First, the relationship between requirements analyzability and software performance was supported at the 0.01 level with a path coefficient of 0.28. This result shows that requirements analyzability has a significant positive effect on software performance (H1). Second, the relationship between interpersonal trust and software performance was supported at the 0.01 level with a path coefficient of 0.59. This result shows that interpersonal trust has a significant positive impact on software performance (H2). Finally, the relationship between requirements analyzability and interpersonal trust was supported at the 0.01 level with a path coefficient of 0.4. The findings show that requirements analyzability has a significant positive effect on interpersonal trust (H3). In addition, the results of hypothesis tests according to the mediating test guidelines of Baron and Kenny (1986) indicate that interpersonal trust may have a mediating role in the relationship between requirements analyzability and software performance. In order to test the mediating effect, the total effect of requirements analyzability on software performance was tested firstly. The total effect of requirements analyzability on software performance is positive and significant (β = 0.527, p ≤0.01). Subsequently, the indirect effect of requirements analyzability on software performance through the interpersonal trust is also positive and significant (β = 0.244, p ≤0.01). The existence of full or partial mediation was assessed by examining the direct effect. The path coefficient for the direct effect of requirements analyzability on software performance, after the inclusion of the mediating variable, was found to be positive and significant (β = 0.282, p≤0.01) (H4). Thus, partial mediation was concluded.

In addition, as a result of testing the structural analysis model, the effect size (f^2) values were examined. Cohen (1988) categorized the effect size values related with R^2 as small (0.02-0.14), moderate (0.15-0.34), and large (above 0.35). According to the analysis results, requirements analyzability has a moderate effect on software project performance (0.15) and interpersonal trust (0.20). The effect size of interpersonal trust on software performance (0.66) is large.

Table 7: Hypothesis Tests

Paths	Paths Coefficients	(STDEV)	*T* Statistics	Effect size (f^2)	*P* Values
Requirements Analyzability-> Software Project Performance (H1)	0,282	0,071	3,977	0,152	0,000
Interpersonal Trust-> Software Project Performance (H2)	0,591	0,071	8,265	0,668	0,000
Requirements Analyzability-> Interpersonal Trust (H3)	0,414	0,089	4,656	0,206	0,000
Requirements Analyzability-> Interpersonal Trust (H4) -> Project Performance	0,244	0,058	4,191	----	0,000

As a result of the structural model analysis, the R^2 values for the dependent variables of software performance and interpersonal trust are 0.567 and 0.171, respectively. The requirements analyzability and interpersonal trust together explain 56% of the variance of software performance. The requirements analyzability explains 17% of the variance of interpersonal trust.

Moderation testing

SPSS PROCESS macro was used to test H5 (Hayes, 2013; model 1). As a result of the analysis, it was found that the interaction effect of requirement analyzability and customization on software performance is significant (B = 0,1057; t = 2.139; $p < 0.05$). Table 8 presents the results for hypotheses H5. With the data obtained, Figure 3 was formed to determine whether the effect of requirements analyzability on software performance represents significant differences according to a degree of customization. The results show that as the customization degree of the software increases, the effect of requirement analyzability on software performance increases. Hypothesis 5a is supported. In addition, it was found that as customization increases, the effect of requirements analyzability on interpersonal trust is significant at the level of 0.1 (B = 0,1005; t = 1.684; p = 0.094). With the data obtained, Figure 4 was formed to determine whether the effect of requirements analyzability on interpersonal trust represents significant differences according to a degree of customization. Hypothesis H5b is partially supported.

Table 8: Regression Results for H5

	B	SE	t	p
Dependent variable: Software Performance				
Requirements Analyzability (RA)	-,0918	,2719	-,3375	,7363
Degree of Customization (CD)	-,4153	,2688	-1,5449	,1247
RAxCD	,1057	,0494	2,1397	,0342
The conditional indirect effect at software performance = M ±1 SD				
-1 SD (4,122)	,3441	,0933	3,6893	,0003
M (5,398)	,4791	,0731	6,5566	,0000
+1 SD (6,674)	,6140	,0996	6,1618	,0000
Dependent variable: Interpersonal Trust				
Requirements Analyzability (RA)	-,0728	,3283	-,2216	,8250
Degree of Customization (CD)	-,4824	,3247	-1,4859	,1397
RAxCD	,1005	,0597	1,6843	,0944
The conditional indirect effect at interpersonal trust= M ±1 SD				
-1 SD (4,122)	,3416	,1127	3,0327	,0029
M (5,398)	,4699	,0882	5,3253	,0000
+1 SD (6,674)	,5982	,1203	4,9708	,0000

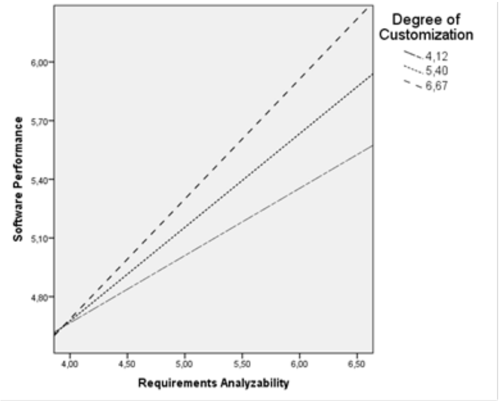

Figure 3: Interaction of Requirements Analyzability and Customization on Software Project Performance

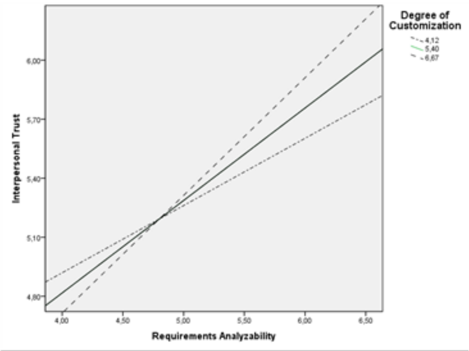

Figure 4: Interaction of Requirements Analyzability and Customization on Interpersonal Trust

Discussion

Findings

This study empirically investigated the influence of requirements analyzability and interpersonal trust over software project performance. Beyond this, customization was examined as a possible moderator variable in the analyzability-trust and analyzability-performance links. Overall, this study provides empirical evidence that interpersonal trust mediates between requirements analyzability and software project performance. More specifically, the results from the data analysis support the hypotheses of the research. First, it is concluded that requirement analyzability has a significant direct effect on software project performance. This finding is fundamental because it may be the first in the software literature to prove the direct effect of requirements analyzability on software performance. Previous studies have primarily addressed requirements uncertainty, partially inferred by analyzability (Na, 2004).

According to the information system literature, requirements analysis is the most critical phase in software development (Wallace et al., 2004). Accuracy of requirements analysis affects all subsequent stages in the project life cycle. This study confirms a strong effect of requirements analyzability on software project performance. The finding here is consistent with the Turkish context, characterized as a high uncertainty avoidance culture (Hofstede, 2003). Tolerance for uncertainty is extremely low in Turkish culture (Sargut, 2001). In societies with high uncertainty avoidance, individuals rely on written rules and official procedures. Requirements analyzability facilitates crafting a complete formal contract. With a low tolerance for uncertainty in Turkey, a complete contract can lead to better project performance in software development.

Second, this research focuses on the role of interpersonal trust rather than inter-organizational trust in software project performance. According to Qi and Chau (2013), software outsourcing requires close working relationships between users and developers so that the effect of interpersonal trust would be more pronounced in software project performance. Consistent with this argument, the present study was examined the effect of interpersonal trust on software performance. The results demonstrated significant effects of interpersonal trust. Consistent with prior research (e.g. Qi & Chau, 2013), this paper provides further support for the performance-enhancing effect of interpersonal trust in software projects. The significance of interpersonal trust in project management may vary across cultures; however, the results obtained are expected in the Turkish context.

Third, it was concluded that a higher degree of requirements analyzability breeds interpersonal trust. This finding contributes to project management because no prior evidence existed for trust in drafting an accurate analysis of requirements. An accurate analysis of the requirements is related to increased interpersonal information sharing and a decrease in uncertainty. In turn, requirements analyzability prevents interpersonal conflicts and breeds interpersonal trust. This study proves trust development in software projects applying reliable and accurate techniques to convert user requirements.

Fourth, this study examined the mediating effect of interpersonal trust on the relationship between requirements analyzability and software project performance. This study revealed that requirements analyzability heavily influences interpersonal trust, a critical factor in software project performance. Thus, requirements analyzability affects performance through interpersonal trust. In other words, a portion of the effect of requirement analyzability on software project performance can be accomplished by building trust between individuals.

Fifth, this study examined the moderating effects of the degree of customization in software projects. The results support the moderation role of customization on the relationship between requirements analyzability and software project performance. In addition, however, partially support was provided for the moderating effect of customization on the relationship between requirements analyzability and interpersonal trust.

Implications for theory and practice

This research has some important implications for both academicians and practitioners. First, this study validates the measure of project performance as a second-order formative construct. Some previous studies were carried out to measure project performance, but they either use a reflective construct to tap different facets of project performance (Na et al., 2004; Wallace, 2004) or reproduce a general construct for project performance by combining all the items of the two sub-dimensions (i.e. product performance and process performance) (Haq et al., 2019; Han and Huang, 2007). Furthermore, software outsourcing literature has generally focused on project performance in different degrees of inter-organizational trust, and a limited number of previous studies exist to investigate the effect of interpersonal trust (Qui & Chau, 2013). This research contributes to the literature by investigating the effect of competence-based trust between boundary spanners on software project development performance. This study supports the main effect of requirements analyzability on software performance. Second, this study considers requirements analyzability as a value-increasing and a capability-based mechanism in adjusting the project scope instead of a risk-reducing mechanism. Doing so adds more insights to the information processing view by highlighting the importance of requirements analyzability on project performance. More importantly, it proves that interpersonal trust partially mediates the relationship between requirements analysis and software project performance. Third, unlike previous studies, this study demonstrates the moderator role of customization, especially in the relationship between requirements analyzability and software project performance. In addition to all these theoretical contributions, consistent with the call of Lacity et al. (2010), this study extends the software outsourcing literature when conducting in a non-western context.

Limitation and future research

This study has some limitations that need caution in interpreting results. First, the software development process as a mutually dependent relation was evaluated from the buyer's perspective (single source). Performing a similar study from the software developers' perspective will provide a more in-depth understanding of software project performance. Second, this study was only conducted in Turkey, and cultural factors may impact the results. In particular, the impact of interpersonal trust as a soft factor in project performance may vary across cultures. Therefore, extreme caution should be exercised when generalizing the findings to different cultures. Third, this study did not consider some critical condition factors such as relationship length and contract duration. Future research may consider including some condition factors for more detailed insights into software project management. Finally, this study is a cross-sectional study. Therefore, a longitudinal study can discover the potential differential effects of requirement analyzability and interpersonal trust during the project life cycle.

Conclusion

In sum, this study has empirically investigated the effect of requirements analyzability and interpersonal trust on software project performance. Based on the information-processing view, a conceptual model was proposed and empirically tested using a cross-sectional survey. The research data were collected from 138 IT managers working at the buyer side in Turkey. Empirical results supported most of the research hypotheses. Furthermore, the findings demonstrated that interpersonal trust mediates the relationship between requirements analysis and software project performance partially. Moreover, the results indicated that interpersonal trust, a soft factor in an exchange relationship, significantly impacts software performance. All in all, this study contributes to the literature, pointing to the critical implications for academic researchers and information technologies practitioners.

Peer-review:

Externally peer-reviewed.

Conflict of interests:

The author(s) has (have) no conflict of interest to declare.

Grant Support:

The authors declared that this study has received no financial support.

Author Contributions:

Idea/Concept/Design: **D. E., T.G.Y.** Data Collection and/or Processing: **D.E**. Analysis and/or Interpretation: **D.E**. Literature Review: **T.G.Y**, **D.E**. Writing the Article: **T.G.Y, D.E.** Critical Review: **T.G.Y.** Approval: **T.G.Y., D.E**

References

Anderson, J. C. (1995). Relationships in business markets: Exchange episodes, value creation, and their empirical assessment. *Journal of the Academy of Marketing Science, 23*(4), 346-350. https://doi.org/10.1177/009207039502300415

Arpaci, I. (2017). Design and development of educational multimedia: The software development process for mobile learning. In Khosrow-Pour, M. (Eds.), (2nd Ed.), *Blended learning: Concepts, methodologies, tools, and applications* (pp. 366-384). Hershey, PA: IGI Global, Information Science Reference. https://doi.org/ 10.4018/978-1-5225-0783-3.ch018.

Ashnai, B., Henneberg, S. C., Naudé, P., & Francescucci, A. (2016). Inter-personal and inter-organizational trust in business relationships: An attitude-behavior–outcome model. *Industrial Marketing Management, 52,* 128-139. https://doi.org/10.1016/j.indmarman.2015.05.020

Bagozzi, R.P., Yi, Y. (1988). On the Evaluation of Structural Equation Models. *Jams 16,* 74-94. https://doi.org/10.1007/BF02723327.

Barthélemy, J., & Quélin, B. V. (2006). Complexity of outsourcing contracts and ex post transaction costs: An empirical investigation. *Journal of Management Studies, 43*(8), 1775-1797. https://doi.org/10.1111/j.1467-6486.2006.00658.x

Bensaou, M., & Anderson, E. (1999). Buyer-supplier relations in industrial markets: when do buyers risk making idiosyncratic investments? *Organization Science, 10*(4), 460-481.

Bensaou, M., & Venkatraman, N. (1996). Inter-organizational relationships and information technology: A conceptual synthesis and a research framework. *European Journal of Information Systems, 5*(2), 84-91.

Cao, Z., & Lumineau, F. (2015). Revisiting the interplay between contractual and relational governance: A qualitative and meta-analytic investigation. *Journal of Operations Management, 33-34*(1), 15-42. https://doi.org/10.1016/j.jom.2014.09.009

Carson, S., Madhok, A., & Wu, T. (2006). Uncertainty, opportunism, and governance: The effects of volatility and ambiguity on formal and relational contracting. *The Academy of Management Journal, 49*(5), 1058-1077.

Chen, H.-G., Jiang, J. J., Chen, J.-C., & Shim, J. T. (2004). The impacts of conflicts on requirements uncertainty and project performance. *Journal of International Technology and Information Management, 13*(3), 15-168.

Chen, I. J., & Paulraj, A. (2004). Towards a theory of supply chain management: The constructs and measurements. *Journal of Operations Management, 22*(2), 119-150. https://doi.org/10.1016/j.jom.2003.12.007

Chin, W. W. (1998). Commentary: Issues and opinion on structural equation modeling. *MIS Quarterly, 22*(1), 7-16.

Cusumano, M., MacCormack, A., Kemerer, C. F., & Crandall, W. (2003). A Global Survey of Software Development Practices. *Center for eBusiness@ MIT*, 178, 1-17.

Deephouse, C., Mukhopadhyay, T., Goldenson, D. R., & Kellner, M. I. (1996). Software processes and project performance. *Journal of Management Information Systems, 12*(3,), 187-205.

Diamantopoulos, A., & Winklhofer, H. M. (2001). Index construction with formative indicators: An alternative to scale development. *Journal of Marketing Research, 38*(2), 269-277. https://doi.org/10.1509/jmkr.38.2.269.18845

Dittrich, Y., Vaucouleur, S., & Giff, S. (2009). ERP customization as software engineering: Knowledge sharing and cooperation. *IEEE Software, 26*(6), 41-47. https://doi.org/10.1109/MS.2009.173

Fornell, C., & Larcker, D. F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research, 18*(1), 39-50. https://doi.org/10.2307/3151312

Galbraith, J. R. (1974). Organization design: An information processing view. *Interfaces, 4*(3), 28-36. https://doi.org/10.1287/inte.4.3.28

Gefen, Wyss, & Lichtenstein. (2008). Business familiarity as risk mitigation in software development outsourcing contracts. *MIS Quarterly, 32*(3), 531-551. https://doi.org/10.2307/25148855

Hair, J. F., Hult, G. T. M., Ringle, C., & Sarstedt, M. (2016). *A primer on partial least squares structural equation modeling (PLS-SEM).* Sage Publications.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2014). *Multivariate data analysis*. Pearson Education Limited.

Han, W.-M., & Huang, S.-J. (2007). An empirical analysis of risk components and performance on software projects. *Journal of Systems and Software, 80*(1), 42-50. https://doi.org/10.1016/j.jss.2006.04.030

Haq, S. U., Gu, D., Liang, C., & Abdullah, I. (2019). Project governance mechanisms and the performance of software development projects: Moderating role of requirements risk. *International Journal of Project Management, 37*(4), 533-548. https://doi.org/10.1016/j.ijproman.2019.02.008

Hayes, A. F. (2013). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach.* The Guilford Press.

Henderson, J. C., & Lee, S. (1992). Managing I/S design teams: A control theories perspective. *Management Science, 38*(6), 757-777. https://doi.org/10.1287/mnsc.38.6.757

Hofstede, G. (2003). *Culture's consequences: comparing values, behaviors, institutions and organizations across nations.* Sage Publications.

Jaffee, D. (2001). *Organization theory: Tension and change*. McGraw-Hill International Editions.

Jiang, J. J., Klein, G., Wu, S. P. J., & Liang, T. P. (2009). The relation of requirements uncertainty and stakeholder perception gaps to project management performance. *Journal of Systems and Software, 82*(5), 801-808. https://doi.org/10.1016/j.jss.2008.11.833

Kern, T., & Willcocks, L. (2002). Exploring relationships in information technology outsourcing: The interaction approach*. European Journal of Information Systems, 11*(1), 3-19. https://doi.org/10.1057/palgrave.ejis.3000415

Kock, N. (2015). Common method bias in PLS-SEM: A full collinearity assessment approach. *International Journal of E-Collaboration, 11*(4), 1-10. https://doi.org/10.4018/ijec.2015100101

Kumar, N., Scheer, L. K., & Steenkamp, J.-B. E. M. (1995). The effects of perceived interdependence on dealer attitudes. *Journal of Marketing Research, 32*(3), 348-356. https://doi.org/10.2307/3151986

Lacity, M. C., Khan, S., Yan, A., & Willcocks, L. P. (2010). A review of the IT outsourcing empirical literature and future research directions. Journal of Information Technology, 25(4), 395-433. https://doi.org/10.1057/jit.2010.21

Lau, E., & Rowlinson, S. (2009). Interpersonal trust and inter-firm trust in construction projects. *Construction Management and Economics, 27*(6), 539-554. https://doi.org/10.1080/01446190903003886

Lioliou, E., Zimmermann, A., Willcocks, L., & Gao, L. (2014). Formal and relational governance in IT outsourcing: Substitution, complementarity and the role of the psychological contract. *Information Systems Journal, 24*(6), 503-535. https://doi.org/10.1111/isj.12038

Liu, J. Y.-C., Chen, H.-G., Chen, C. C., & Sheu, T. S. (2011). Relationships among interpersonal conflict, requirements, uncertainty, and software project performance. *International Journal of Project Management, 29*(5), 547-556. https://doi.org/10.1016/j.ijproman.2010.04.007

Luna-Reyes, L. F., Black, L. J., Cresswell, A. M., & Pardo, T. A. (2008). Knowledge sharing and trust in collaborative requirements analysis. *System Dynamics Review, 24*(3), 265-297. https://doi.org/10.1002/sdr.404

McManus, J. & Wood-Harper, T. (2007). Understanding the sources of information systems project failure. *Management Services, 51*(3), 38-43.

Miranda, S. M., & Kavan, C. B. (2005). Moments of governance in IS outsourcing: Conceptualizing effects of contracts on value capture and creation. *Journal of Information Technology, 20*(3), 152-169. https://doi.org/10.1057/palgrave.jit.2000045

Moynihan, T. (2000). Coping with "requirements-uncertainty": The theories-of-action of experienced IS/software project managers. *Journal of Systems and Software, 53*(2), 99–109. doi:10.1016/s0164-1212(00)00049-2.

Na, K.-S., Li, X., Simpson, J. T., & Kim, K.-Y. (2004). Uncertainty profile and software project performance: A cross-national comparison. *Journal of Systems and Software, 70*(1-2), 155-163. https://doi.org/10.1016/S0164-1212(03)00014-1

Nidumolu, S. (1995). The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable. *Information Systems Research, 6*(3), 191-219. https://doi.org/10.1287/isre.6.3.191

Nidumolu, S. R. (1996). A comparison of the structural contingency and risk-based perspectives on coordination in software development projects. *Journal of Management Information Systems, 13*(2), 77-113. https://doi.org/10.1080/07421222.1996.11518124

Nooteboom, B. (2004). *Inter-firm collaboration, learning and networks: An integrated approach.* Psychology Press.

Oza, N. V., Hall, T., Rainer, A., & Grey, S. (2006). Trust in software outsourcing relationships: An empirical investigation of Indian software companies. *Information and Software Technology, 48*(5), 345-354. https://doi.org/10.1016/j.infsof.2005.09.011

Parthasarathy, S., & Daneva, M. (2016). An approach to estimation of degree of customization for ERP projects using prioritized requirements. *Journal of Systems and Software, 117*, 471-487. https://doi.org/10.1016/j.jss.2016.04.006

Perrow, C. (1967). A framework for the comparative analysis of organizations. *American Sociological Review, 32*(2), 194-208. https://doi.org/10.2307/2091811

Premkumar, G., Ramamurthy, K. & Saunders, C.S. (2005). Information processing view of organizations: An exploratory examination of fit in the context of interorganizational relationships. *Journal of Management Information Systems,* 22(1), 257-294. https://doi.org/10.1080/07421222.2003.11045841

Podsakoff, P. M., & Organ, D. W. (1986). Self-reports in organizational research: Problems and prospects. *Journal of Management, 12*(4), 531-544. https://doi.org/10.1177/014920638601200408

Qi, C., & Chau, P. (2015). Relationship or contract? Exploring the key factor leading to IT outsourcing success in China. *Information Technology & People, 28*(3), 466-499.

Qi, C., & Chau, P. Y. K. (2013). Investigating the roles of interpersonal and interorganizational trust in IT outsourcing success. *Information Technology & People, 26*(2), 120-145. https://doi.org/10.1108/ITP-09-2012-0088

Ringle, C. M., Wende, S., & Becker, J. M. (2015*). SmartPLS 3.* SmartPLS GmbH. Boenningstedt.

Ringle, C., Sarstedt, M. & Straub, D. W. (2012). Editor's comments: A Critical look at the use of PLS-SEM in "MIS Quarterly". *MIS Quarterly, 36*(1), iii-xiv. https://doi.org/10.2307/41410402

Sargut, S. (2001). *Kültürler arası farklılaşma ve yönetim.* Ankara: İmge Kitabevi Yayınları.

Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international delphi study. *Journal of Management Information Systems, 17*(4), 5-36. https://doi.org/10.1080/07421222.2001.11045662

Srinivasan, S. S. (2009). Performance under requirements uncertainty: A personality perspective. *Proceedings of Pacific Asia Conference on Information Systems,* PACIS, Hyderabad, India.

Standish Group Internatıonal, Inc. (2014). Third Quarter Research Report. Available at: https://www.projectsmart.co.uk/white-papers/chaos-report.pdf (access date: 11.02.2021)

Susarla, A., Subramanyam, R., & Karhade, P. (2010). Contractual provisions to mitigate hold-up: Evidence from information technology outsourcing. *Information Systems Research, 21*(1), 37-55. https://doi.org/10.1287/isre.1080.0204

Turner, J. (1992). A Comparison of the process of knowledge elicitation with that of requirements determination. In W.W. Cotterman et al. (Eds.), *Challenges and strategies for research in systems development* (pp. 415-430). New York: Wiley.

Wade & Hulland. (2004). The Resource-based view and information systems research: Review, extension, and suggestions for future research. *MIS Quarterly, 28*(1), 107-142. https://doi.org/10.2307/25148626

Wallace, L., Keil, M., & Rai, A. (2004). How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. *Decision Sciences, 35*(2), 289-321. https://doi.org/10.1111/j.00117315.2004.02059.x

Wang, E. T. G. (2002). Transaction attributes and software outsourcing success: An empirical Investigation of transaction cost theory. *Information Systems Journal, 12*(2), 153-181. https://doi.org/10.1046/j.1365-2575.2002.0012

Wang, E. T. G., Shih, S.-P., Jiang, J. J., & Klein, G. (2006). The relative influence of management control and user–IS personnel interaction on project performance. *Information and Software Technology, 48*(3), 214-220. https://doi.org/10.1016/j.infsof.2005.04.003

Zucker, L. (1987). Institutional theories of organization. *Annual Review of Sociology, 13,* 443-464.

Zaheer, A., McEvily, B., & Perrone, V. (1998). Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance. *Organization Science, 9*(2), 141-159. https://doi.org/10.1287/orsc.9.2.141

Zowghi, D., & Nurmuliani, N. (2002). A study of the impact of requirements volatility on software project performance. *Ninth Asia-Pacific Software Engineering Conference,* pp.3-11. https://doi.org/10.1109/APSEC.2002.1182970