Research Journal of Applied Sciences, Engineering and Technology 13(2): 85-97, 2016
DOI:10.19026/rjaset.13.2919
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 02, 2014 Accepted: March 24, 2014 Published: July 15, 2016

Research Article

Managing and Controlling Design Process Issues by Using Stepwise Approach Modelling

[1, 2]Abdelrafe Elzamly, [1]Burairah Hussin, [1]Norhaziah Salleh and [1]Abdul Samad Shibghatullah
[1]Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia
[2]Department of Computer Science, Al-Aqsa University, Palestine

Abstract: The objective of study is to introduce stepwise regression approach modelling to mitigate and manage the software design process issues in software project by using proposed controls. In addition, the software design process issues were controlled and modelled by using stepwise regression approach except risk 3. Furthermore, we illustrated the design process issues were mitigated by control techniques in Table 40. However, we need to combine more techniques and artificial optimal algorithms to mitigate the issues in software design process.

Keywords: Design process, risk controls, software risk management, software design issues, stepwise approach modelling

INTRODUCTION

Despite much research and progress in the area of software project management, software development projects still fail to deliver acceptable systems on time and within budget. In addition, risk is an uncertainty that can have a negative or positive effect on meeting project objectives. Risk management is the process of identifying, analyzing and controlling risk throughout the life of a project to meet the project objectives (Schwalbe, 2010). However, Software Development Lifecycle is the processes of creating and risk control techniques are used to mitigate issues it should involve in all phases include: Planning, analysis, design, implementation and maintenance. In addition, we focused on Design phase (Hoffer *et al.*, 2011): It involves the actual creation and design of a system. This involves putting together the different pieces that will create the system. Techniques and models for mitigating risk in software development projects classified into three categories–namely, qualitative, quantitative and intelligent approaches (Elzamly and Hussin, 2015b). The objective of this study is: To identify issues in the software design process in the Palestinian software development organizations, to manage and mitigate the software design issue factors by using quantitative approach techniques.

LITERATURE REVIEW

We improved quality of software projects of the participating companies while estimating the quality-affecting risks in IT software projects. The results show that there were 40 common risks in software projects of IT companies in Palestine. The amount of technical and non-technical difficulties was very large (Elzamly and Hussin, 2011a). However, we presented the chi-square (χ^2) test to control the risks in a software project. Fourteen risk factors and eighteen control factors were used (Khanfar *et al.*, 2008). Further, we present new mining technique that uses the fuzzy regression analysis modelling techniques to manage the risks in a software development project. Top ten software risk factors in planning phase and thirty risk management techniques were presented to respondents (Elzamly and Hussin, 2014b). Therefore, regression test (simultaneous selection procedure) and effect size test proposed to managing the risks in a software project and reducing risk with software process improvement (Elzamly and Hussin, 2011b). Generally, to manage software risk we need to know the software risk factors and control factors. Stepwise Regression analysis is used to compare to compare the controls to each of the risk factors in implementation phase to determine if they are effective in mitigating the occurrence of each risk (Elzamly and Hussin, 2013a). In addition, we presented a new mining technique that introduced the fuzzy multiple regression analysis to manage the risks in a software development project in the maintenance phase (Elzamly and Hussin, 2014c). Finally, risk management methodology that has five phases: Risk identification (planning, identification, prioritization), risk analysis and evaluation (risk analysis, risk evaluation), risk treatment, risk controlling, risk communication and

Corresponding Author: Abdelrafe Elzamly, Department of Computer Science, Al-Aqsa University, P.O. Box: 4051, Gaza, Palestine, Tel.: + 970599/855174

documentation these relied on three categories techniques as risk qualitative analysis, risk quantitative analysis and risk mining analysis throughout the life of a software project to meet the goals (Elzamly *et al*., 2015a; Elzamly and Hussin, 2014a).

TOP 10 SOFTWARE DESIGN PROCESS ISSUE FACTORS

Indeed, we classify the top ten software design process issue factors in SDLC and thereafter these issues need to be mitigated and controlled, these issues were explain in this section and Table 1 below:

Risk 01: Introduction of new technology: New technology is always regarded as required technology in a software project reported by Addison (2003), Boehm (1991, 2007), Han and Huang (2007), Ewusi-Mensah (2003), Nakatsu and Iacovou (2009) and Surie (2004). Introduction of new technology usually bring about a number of training needs. This technology may be used operationally in processes that involve packaging, quality testing, money transaction and other (Buckley and Caple, 2009; Ewusi-Mensah, 2003). As a result software project that applied poor technology investments will have a high risk to dealing with software project complexity and competitiveness (Bennatan, 2006; Han and Huang, 2007; Huang and Han, 2008; Pandian, 2007).

Risk 02: Developing the wrong software functions and properties: There are possibilities of developing wrong software functions and properties in software project (Addison and Vallabh, 2002; Boehm, 1991; Dash and Dash, 2010; Selby, 2007). Functionality risk is defined as the software risk factors in a completed system that does not meet users' expectations or a development of software functions that are not needed or are wrongly specified (Kaur and Malhotra, 2013). Developing the wrong software function will lead to problems such as incorrect operation of the software, unexpected results, increase in maintenance for taking corrective actions, user dissatisfaction, overhead of rework, product rejection which adversely affects developer reputation and that lead the software project to the failure (Kaur and Malhotra, 2013).

Risk 03: Developing the wrong user interface: There are circumstances when wrong user interfaces are developed in software projects (Boehm, 1991, 2007; Dash and Dash, 2010; Sodhi and Sodhi, 2001; Surie, 2004). Although software projects did an adequate job software functionality, but they created every unfriendly user interface (Selby, 2007). As a result of unclear user interfaces unclear; software projects are not able to specify the planning of the human interface including screen formats, contents of menus and command structure (Jalote, 2008).

Risk 04: Insufficient procedures to ensure security, integrity and availability of the database: According to Addison (2003), Aritua *et al.* (2011), Ewusi-Mensah (2003), Nakatsu and Iacovou (2009) and Sumner (2000), insufficient procedures to ensure security, integrity and availability of the database is also related to major risks in a software project. Generally, insufficient attributes of software project requirements in database that include reliability, integrity, availability, security, maintainability and portability (Laplante, 2004) result in a loss of data and low quality factors to deliver the software within the stipulated date (Addison, 2003).

Risk 05: Lack of integrity/consistency: One of the main factors associated with failure risk is lack of integration in the software components (Sumner, 2000) and according to Aloini *et al.* (2007), Chen and Huang (2009) and Nakatsu and Iacovou (2009), it is also referred to a low degree of integration and flexibility. Furthermore, it also means that some implementations and processes are carried out in isolation with other processes (Rudd, 2010). Therefore, lack of this integrity

Table 1: Illustrate top ten software design processes issue factors (Elzamly and Hussin, 2013b)

Phase	No	Software design processes issue factors	Frequency
Design	1	Introduction of new technology (Boehm, 1991; Han and Huang, 2007; Khanfar *et al*., 2008; Ewusi-Mensah, 2003; Nakatsu and Iacovou, 2009).	5
	2	Developing the wrong software functions and properties (Addison and Vallabh, 2002; Boehm, 1991; 2002b; Ewusi-Mensah, 2003; Nakatsu and Iacovou, 2009).	5
	3	Developing the wrong user interface (Boehm, 1991, 2002a, 2002b, 2007).	4
	4	Insufficient procedures to ensure security, integrity and availability of the database (Addison, 2003; Ewusi-Mensah, 2003; Lyons and Skitmore, 2004; Sumner, 2000).	4
	5	Lack of integrity/consistency (Aloini *et al*., 2007; Chen and Huang, 2009; Nakatsu and Iacovou, 2009; Sumner, 2000).	4
	6	Lack of architecture and quality software project (Boehm, 2002b, 2007; Sumner, 2000).	3
	7	Absence of quality architectural and design documents (Boehm, 2007; Chen and Huang, 2009; Khanfar *et al*., 2008).	3
	8	Failure to redesign and design (blueprints) software processes (Kettunen and Laanti, 2005; Sumner, 2000).	2
	9	Lack of effective software project team integration between clients, the supplier team and the supply chain (Cliff Mitchell, 2011).	1
	10	Misalignment of software project with local practices and processes (Aloini *et al*., 2007).	1
	Total frequency		32

among software managers will lead to the integrated components that are incorrect, incomplete and inconsistent with their intended propose (Fairley, 2009).

Risk 06: Lack of architecture and performance software project: Certainly, poor software structures will lead to software failure as reported by Pandian (2007), Aritua *et al.* (2011), Ewusi-Mensah (2003) and Boehm (2007). This is even worst where process performance process performance data model to understand the organization's standard of processes (Fairley, 2009). With that matters process of each component is incomplete, incorrect and inconsistent with respect to the architectural design, iterative development processes that support incremental implementation, verification and validation of software need to be built (Fairley, 2009).

Risk 07: Absence of quality architectural and design documents: Besides, lack of reusable documentation, inadequate system documentation is another factor to be taken into consideration (Aloini *et al.*, 2007; Chen *et al.*, 2009; Khanfar *et al.*, 2008; Pandian, 2007). Architectural design of the software is concerned with specifying the major software components and interfaces, their interrelationships and their connections to the environment of the software project (Fairley, 2009). Therefore, it is also important to recognize the lack of a well–designed and documented process as a characteristic of the initial level of maturity (Ahmed, 2011).

Risk 08: Failure to redesign and design (blueprints) software processes: Aloini *et al.* (2007), Nakatsu and Iacovou (2009), Pandian (2007) and Sumner (2000) reported that lack of the designs risk (blueprints) of software process is another factor of software risk. Thus, any major redesign in the product's architecture which is not reflected by the blueprint lacks communication (Thomas and Uwe, 2010).

Risk 09: Lack of effective software project team integration between clients, the supplier team and the supply chain: According to Cliff Mitchell (2011) and Schmidt *et al.* (2001), lack of effective software project team integration between clients, the supplier team and the supply chain is critical in the software project. House of Commons (2009) reported that the software project always selected suppliers as the main delivery partners and did not seek a wider evaluation of their requirements from the market. Without good communication (House of Commons, 2009), this will lead to failure in the delivery of the software project on time and within budgets thus resulting in bad practices.

Risk 10: Misalignment of software project with local practices and processes: Paré *et al.* (2008) reported that misalignment of software project with local practices and processes might lead to a software failure (Paré *et al.*, 2008). For example, Aloini *et al.* (2007) and Yogesan *et al.* (2009), pontificate that aligning new systems with local work practices but due to the lack of consistencies between the new system and local work practices may result in processes that are less promising (Yogesan *et al.*, 2009).

RISK MANAGEMENT TECHNIQUES

We listed thirty risk controls that are considered important in minimizing the software issue factors in design process identified; these risk controls are (Elzamly and Hussin, 2015b; Elzamly *et al.*, 2015b; Elzamly and Hussin, 2014a, 2014b, 2014d, 2014f, 2015a):

C1 : Using of requirements scrubbing
C2 : Stabilizing requirements and specifications as early as possible
C3 : Assessing cost and scheduling the impact of each change to requirements and specifications
C4 : Develop prototyping and have the requirements reviewed by the client
C5 : Developing and adhering a software project plan
C6 : Implementing and following a communication plan
C7 : Developing contingency plans to cope with staffing problems
C8 : Assigning responsibilities to team members and rotate jobs
C9 : Have team-building sessions
C10: Reviewing and communicating progress to date and setting objectives for the next phase
C11: Dividing the software project into controllable portions
C12: Reusable source code and interface methods
C13: Reusable test plans and test cases
C14: Reusable database and data mining structures
C15: Reusable user documents early
C16: Implementing/Utilizing automated version control tools
C17: Implement/utilize benchmarking and tools of technical analysis
C18: Creating and analyzing process by simulation and modeling
C19: Provide scenarios methods and using of the reference checking
C20: Involving management during the entire software project lifecycle
C21: Including formal and periodic risk assessment
C22: Utilizing change control board and exercise quality change control practices
C23: Educating users on the impact of changes during the software project
C24: Ensuring that quality-factor deliverables and task analysis
C25: Avoiding having too many new functions on software projects

C26: Incremental development (deferring changes to later increments)
C27: Combining internal evaluations by external reviews
C28: Maintain proper documentation of each individual's work
C29: Provide training in the new technology and organize domain knowledge training
C30: Participating users during the entire software project lifecycle.

The literature review revealed the following question: Do experienced project managers control software project risk factors in design phase by using the controls identified in this study? To answer this question, the following objectives for the empirical work have been set forth: Identify the software design risk factors of software projects in the Palestinian software development organizations, to rank the software design risk factors according to their importance, severity and occurrence frequency based on data source, to identify the activities performed by software project managers to manage the software design project risks which identified.

EMPIRICAL STRATEGY (RESULTS AND DISCUSSION)

We collected data by using questionnaire and historical data for mitigating issues in software project. In addition, snowball and distribution personal methods are used to select sampling and then survey questionnaire were distributed to 76 software project managers in Palestinian software development organizations. In this study, we used correlation analysis, regression analysis models based on stepwise selection method and Durbin-Watson Statistic by using IBM SPSS statistics for manipulating and analyzing the data set.

Modelling software design process issues by controls variables: These tests were performed using stepwise modelling analysis, to mitigate software design process issues by controls.

R1: Risk of 'introduction of new technology' compared to controls: Table 2 to 5 show that the significant value is less than the assumed value at α = 0.05 level of significance, so there was a positive relation between controls 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 23 and risk 1. Any risk management techniques (controls) were no significant and thus were not reported. However, for model 1 it was goodness to predict risk 1 (the dependent variable) from independent variables such as control 3. In addition, control 3 had an impact on risk 1. In addition, the results showed that control 3 had a positive impact value of 0.513 and the value of R^2 is 0.264. This was interpreted as a percentage of 26.4 % from the dependent variable of risk 1. Therefore, there was no multicollinearity when collinearity diagnosis (Tolerance test and VIF test) were used. Furthermore, the Durbin–Watson statistic (D) was 1.986 and (d_U = 1.652, d_L= 1.598) based on K = 1, N = 76, at α = 0.05; there is no autocorrelation based on the rule ($d_U < D < 2+d_L$: No autocorrelation).

R2: Risk of 'developing the wrong software functions and properties' compared to controls: Table 6 to 9 show that the significant value is less than the assumed value at α = 0.05 level of significance. Therefore, there was a positive relation between controls 4, 19, 28 and risk 2. Any risk management techniques (controls) were no significant and thus were not reported. However, for model 1 it was goodness to predict risk 2 (the dependent variable) from independent variables such as control 4. In addition, control 4 has an impact on risk 2. Besides, the results showed that control 4 had positive impact value of 0.276 and the value of R^2 was 0.076. This was interpreted as a percentage of 7.6 % from the dependent variable of risk 2. Therefore, there was no multicollinearity when collinearity diagnosis (Tolerance test and VIF test) were used. Furthermore, the Durbin-Watson statistic (D) was 1.876 and (d_U = 1.652, d_L = 1.598) based on K = 1, N = 76, at α = 0.05. Thus, there is evidence of no autocorrelation based on the rule ($d_U < D < 2+d_L$: No autocorrelation).

R3: Risk of 'developing the wrong user interface' compared to controls: Table 10 and 11 show that the significant value is greater than the assumed value of α = 0.05 level of significance, this implies that is no relation between all controls and risk 3 and controls that do not have a relation (no significant) were not reported. In addition, we found the model is to be unfit.

R4: Risk of 'Insufficient Procedures to Ensure Security, Integrity and Availability of The Database' Compared to Controls: Table 12 to 15 show that the significant value is less than the assumed value at α = 0.05 level of significance. Thus, there was a positive relation between controls 2, 4, 8, 9, 10, 19, 23, 24, 25, 26, 28 and risk 4. In addition, controls 19 and 24 have an impact on the risk 24. Any risk management techniques were no significant and thus were not reported. However, it was goodness that model 2 is used to predict a risk 24 (the dependent variable) from independent variables such as controls 19, 24. In addition, the results showed that control 19 and 24 had a positive impact value of 0.335 and 0.273 respectively. Thus, the multiple correlation value was 0.403 and the value of R^2 was 0.162. This was interpreted as a percentage of 16.2 % from the dependent variable of risk 24. Therefore, there was no multicollinearity when collinearity diagnosis (Tolerance test and VIF test) were used. Furthermore, the Durbin-Watson statistic (D) was 2.106 and (d_u = 1.680, d_L= 1.571) based on K=2, N = 76, at α = 0.05; there is evidence of no autocorrelation ($d_U < D < 2+d_L$: No autocorrelation).

R5: Risk of 'lack of integrity/consistency' compared to controls: Table 16 to 19 show that the significant value is less than the assumed value at α = 0.05 level of significance, so there was a positive relation between controls 1, 2, 3, 4, 7, 8, 9, 10, 13, 19, 21, 23, 24, 25, 26, 27, 28, 29 and risk 5. Any risk management techniques (controls) were no significant and thus were not reported. It was goodness to predict risk 25 (the dependent variable) from independent variables such as controls 27 and 24 for model 2. Controls 24 and 27 had an impact on risk 5. In addition, the results showed that control 24 and 27 had a positive

Table 2: Illustrates the value of correlation

C1	C2	C3	C4	C5	C6	C7
0.275*	0.288*	0.513**	0.346**	0.347**	0.304**	0.304**
C8	C9	C10	C11	C15	C23	
0.256*	0.343**	0.331**	0.305**	0.234*	0.233*	

*: Correlation is significant at the 0.05 level (2-tailed); **: Correlation is significant at the 0.01 level (2-tailed)

Table 3: Illustrates the value of correlation and R square

Model	R	R^2	Durbin-Watson
1	0.513^a	0.264	1.986

Predictors: (Constant), c3

Table 4: Illustrates an analysis of variance ($ANOVA^b$)

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	15.294	1	15.294		
	Residual	42.745	74	0.5780	26.477	0.000^a
	Total	58.039	75			

Predictors: (Constant), c3; Dependent Variable: r1

Table 5: Illustrates the coefficients and distributed T ($Coefficients^a$)

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	T	Sig.
1	(Constant)	2.212		4.330	0.000
	c3	0.524	0.513	5.146	0.000

Dependent Variable: r1

Table 6: Illustrates the value of correlation

C4	C19	C28
0.276*	0.249*	0.252*

Table 7: Illustrates the value of correlation and R square

Model	R	R^2	Durbin-Watson
1	0.276^a	0.076	1.876

Predictors: (Constant), c4

Table 8: Illustrates an analysis of variance ($ANOVA^b$)

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	2.2890	1	2.289	6.103	0.016^a
	Residual	27.751	74	0.375		
	Total	30.039	75			

Predictors: (Constant), c4; Dependent Variable: r2

Table 9: Illustrates the coefficients and distributed T ($Coefficients^a$)

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	T	Sig.
1	(Constant)	3.746		8.643	0.000
	c4	0.212	0.276	2.470	0.016

Dependent Variable: r2

Table 10: Illustrates the value of correlation and R square

Model	R	R^2
1	0.513^a	0.264

a Predictors: (Constant), c30, c12, c25, c7, c20, c17, c13, c4, c23, c24, c29, c16, c15, c21, c27, c19, c9, c26, c14, c1, c22, c28, c11, c8, c2, c5, c3, c6, c10, c18 impact on the value of 0.387 and 0.398 respectively. In addition, the results showed that control 24 and 27 had a positive impact on the value of 0.387 and 0.398 respectively. The multiple correlation value was 0.480,

Table 11: Illustrates an analysis of variance (ANOVA[b])

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	9.408	30	0.314	0.537	0.962[a]
	Residual	26.276	45	0.584		
	Total	35.684	75			

Predictors: (Constant), c30, c12, c25, c7, c20, c17, c13, c4, c23, c24, c29, c16, c15, c21, c27, c19, c9, c26, c14, c1, c22, c28, c11, c8, c2, c5, c3, c6, c10, c18; Dependent Variable: r3

Table 12: Illustrates the value of correlation

C2	C4	C8	C9	C10	C19
0.274*	0.267*	0.300**	0.229*	0.277*	0.335**
C23	C24	C25	C26	C28	
0.256*	0.273*	0.251*	0.263*	0.289*	

Table 13: Illustrates the value of correlation and R square

Model	R	R^2	Durbin-Watson
1	0.335[a]	0.112	
2	0.403[b]	0.162	2.106

Predictors: (Constant), c19; Predictors: (Constant), c19, c24

Table 14: Illustrates an analysis of variance (ANOVA[c])

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	5.013	1	5.013	9.339	0.003[a]
	Residual	39.724	74	0.537		
	Total	44.737	75			
2	Regression	7.258	2	3.629	7.069	0.002[b]
	Residual	37.478	73	0.513		
	Total	44.737	75			

a. Predictors: (Constant), c19; b. Predictors: (Constant), c19, c24; c. Dependent Variable: r4

Table 15: Illustrates the coefficients and distributed T (Coefficients[a])

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	t	Sig.
1	(Constant)	2.650		3.851	0.000
	C19	0.404	0.335	3.056	0.003
2	(Constant)	1.538		1.794	0.077
	C19	0.361	0.300	2.763	0.007
	C24	0.262	0.227	2.091	0.040

Dependent variable: r4

Table 16: Illustrates the value of correlation

C1	C2	C3	C4	C7	C8
0.246*	0.372**	0.389**	0.267*	0.244*	0.269*
C9	C10	C13	c19	c21	c23
0.333**	0.258*	0.226*	0.228*	0.235*	0.384**
c24	c25	c26	c27	c28	c29
0.387**	0.233*	0.373**	0.398**	0.279*	0.240*

Table 17: Illustrates the value of correlation and R^2

Model	R	R^2	Durbin-watson
1	0.398[a]	0.158	
2	0.480[b]	0.230	2.150

Predictors: (Constant), c27; Predictors: (Constant), c27, c24

Table 18: Illustrates an analysis of variance (ANOVA[c])

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	5.950	1	5.950	13.897	0.000[a]
	Residual	31.682	74	0.428		
	Total	37.632	75			
2	Regression	8.660	2	4.330	10.910	0.000[b]
	Residual	28.972	73	0.397		
	Total	37.632	75			

Predictors: (Constant), c27; Predictors: (Constant), c27, c24; Dependent Variable: r5

Table 19: Illustrates the coefficients and distributed T (Coefficients[a])

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	T	Sig.
1	Constant	3.000		6.450	0.000
	C27	0.327	0.398	3.728	0.000
2	Constant	1.879		3.032	0.003
	C27	0.248	0.301	2.764	0.007
	C24	0.302	0.285	2.613	0.011

Dependent Variable: r5

and the value of R^2 was 0.230. This was interpreted as a percentage of 23.0 % from the dependent variable of risk 5. Thus, the use of collinearity diagnosis did not reveal multicollinearity. Furthermore, Durbin-Watson statistic (D) is 2.150 and (d_U = 1.680, d_L = 1.571) based on K = 2, N = 76, at α = 0.05. As such, there is evidence of no autocorrelation (d_U <D<2+d_L: No autocorrelation).

R6: Risk of 'lack of architecture, performance and quality software project' compared to controls: Table 20 to 23 show that the significant value is less than the assumed value at α = 0.05 level of significance, so there was a positive relation between controls 1, 2, 3, 4, 10, 11, 23, 24, 26, 28, 29, 30 and risk 6. Any risk management techniques (controls) were no significant and thus were not reported. However, the goodness of model 2 is that it was able to predict risk 26 (the dependent variable) from independent variables such as controls 26 and 30. Controls 26 and 30 had an impact on risk 6. In addition, the results showed that controls 26 and 30 have a positive impact value of 0.420 and 0.358 respectively. Thus, the multiple correlation value was 0.474 and the value of R^2 was 0.225. This was interpreted as a percentage of 22.5 % from the dependent variable of risk 6. Therefore, there is not multicollinearity by using collinearity diagnosis in Tolerance test and VIF test. Furthermore, the Durbin-Watson statistic (D) was 2.211 and (d_U = 1.680, d_L = 1.571) based on K = 2, N = 76, at α = 0.05; there was no evidence of autocorrelation (d_U < D < 2+d_L: No autocorrelation).

R7: Risk of 'absence of quality architectural and design documents' compared to controls: Table 24 to 27 show that the significant value is less than the assumed value at where α = 0.05 level of significance, so there was a positive relation between controls 1, 2, 7, 10, 24, 26 and risk 7. Any risk management techniques (controls) were no significant and thus were not reported. However, the model 1 was good to predict risk 7 (the dependent variable) from independent variables such as controls 2, 15. Controls 2 and 15 had an impact on the risk 7. In addition, the results showed that the multiple correlation value was 0.418 and the value of R^2 was 0.175. This was interpreted as a percentage of 17.5 % from the dependent variable of risk 7. Thus, the use of collinearity diagnosis did not reveal multicollinearity. Furthermore, the Durbin-Watson statistic (D) was 2.029 and (d_U = 1.680, d_L = 1.571) based on K = 2, N = 76, at α = 0.05; thus, there is no evidence of autocorrelation (d_U <D< 2+d_L: No autocorrelation).

R8: Risk of 'failure to design (blueprints) and redesign software processes' compared to controls: Table 28 to 31 show that the significant value is less than the assumed value at where α = 0.05 level of significance, so there was a positive relation between controls 2, 3, 4, 9, 10, 20, 21, 24, 27, 28, 29 and risk 8. Any risk management techniques (controls) were no significant and thus were not reported. However, model 4 was good to predict risk 8 (the dependent variable) from independent variables such as

Table 20: Illustrates the value of correlation

c1	c2	c3	c4	c10	c11
0.324**	0.331**	0.328**	0.268*	0.312**	0.241*
c23	c24	c26	c28	c29	c30
0.227*	0.376**	0.420**	0.315**	0.240*	0.358**

Table 21: Illustrates the value of correlation and R^2

Model	R	R^2	Durbin-Watson
1	0.420[a]	0.177	
2	0.474[b]	0.225	2.211

a. Predictors: (Constant), c26; b. Predictors: (Constant), c26, c30

Table 22: Illustrates an analysis of variance (ANOVA[c])

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	8.902	1	8.902	15.891	0.000[a]
	Residual	41.453	74	0.560		
	Total	50.355	75			
2	Regression	11.330	2	5.665	10.597	0.000[b]
	Residual	39.025	73	0.535		
	Total	50.355	75			

a. Predictors: (Constant), c26; b. Predictors: (Constant), c26, c30; c. Dependent Variable: r26

Table 23: Illustrates the coefficients and distributed T (Coefficients^{a})

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	t	Sig.
1	Constant	2.162		3.512	0.001
	C26	0.471	0.420	3.986	0.000
2	Constant	1.660		2.570	0.012
	C26	0.375	0.335	3.025	0.003
	C30	0.184	0.236	2.131	0.036

Dependent Variable: r6

Table 24: Illustrates the value of correlation

C1	C2	C7	C10	C24	C26
0.227**	0.355**	0.241**	0.280**	0.251**	0.283**

Table 25: Illustrates the value of correlation and R square

Model	R	R^2	Durbin-Watson
1	0.355^{a}	0.126	
2	0.418^{b}	0.175	2.029

a. Predictors: (Constant), c2; b. Predictors: (Constant), c2, c15

Table 26: Illustrates an analysis of variance (ANOVA^{c})

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	4.974	1	4.974	10.690	0.002^{a}
	Residual	34.434	74	0.465		
	Total	39.408	75			
2	Regression	6.882	2	3.441	7.723	0.001^{b}
	Residual	32.526	73	0.446		
	Total	39.408	75			

a. Predictors: (Constant), c2; b. Predictors: (Constant), c2, c15; c. Dependent Variable: r7

Table 27: Illustrates the coefficients and distributed T (Coefficients^{a})

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	T	Sig.
1	Constant	2.919		5.471	0.000
	C2	0.350	0.355	3.270	0.002
2	Constant	3.966		5.455	0.000
	C2	0.410	0.417	3.774	0.000
	C15	-0.267	-0.228	-2.069	0.042

a. Dependent Variable: r7

Table 28: Illustrates the value of correlation

C2	C3	C4	C9	C10	C20
0.311**	0.382**	0.379**	0.280*	0.288*	0.286*
C21	C24	C27	C28	C29	
0.290*	0.315**	0.254*	0.283*	0.407**	

Table 29: Illustrates the value of correlation and R^2

Model	R	R^2	Durbin-Watson
1	0.407^{a}	0.165	
2	0.481^{b}	0.231	
3	0.529^{c}	0.280	
4	0.570^{d}	0.325	1.765

a. Predictors: (Constant), c29;b. Predictors: (Constant), c29, c4; c. Predictors: (Constant), c29, c4, c24; d. Predictors: (Constant), c29, c4, c24, c5

Table 30: Illustrates an analysis of variance (ANOVA^{e})

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	6.230	1	6.230	14.657	0.000^{a}
	Residual	31.454	74	0.425		
	Total	37.684	75			
2	Regression	8.710	2	4.355	10.973	0.000^{b}
	Residual	28.974	73	0.397		
	Total	37.684	75			
3	Regression	10.553	3	3.518	9.335	0.000^{c}
	Residual	27.131	72	0.377		
	Total	37.684	75			
4	Regression	12.242	4	3.060	8.540	0.000^{d}
	Residual	25.443	71	0.358		
	Total	37.684	75			

a. Predictors: (Constant), c29; b. Predictors: (Constant), c29, c4 c. Predictors: (Constant), c29, c4, c24; d. Predictors: (Constant), c29, c4, c24, c5; e. Dependent Variable: r8 controls 29, 4, 24, 5. Controls 4, 5, 24 and 29 had an impact on risk 8. In addition, the multiple correlation value was 0.570 and the value of R^2 was 0.325. This was interpreted as a percentage of 32.5 % from the dependent variable of risk 8. Therefore, there was no multicollinearity shown in the use of collinearity diagnosis (Tolerance test and VIF test). Furthermore, the Durbin-Watson statistic (D) was 1.765 and (d_U = 1.515, d_L = 1.739) based on K = 4, N = 76, at $\alpha = 0.05$; thus stating there was no evidence of autocorrelation ($d_U < D < 2+d_L$: No autocorrelation).

R9: Risk of 'lack of effective software project team integration between clients, the supplier team and the supply chain' compared to controls: Table 32 to 35 show that the significant value is less than the assumed value at $\alpha = 0.05$ level of significance, so there was a positive relation between controls 4, 16, 19, 20, 24, 25, 26, 27, 28, 29, 30 and risk 9. Any risk management techniques (controls) were no significant and thus were not reported. However, it was good that model 1 was able to predict risk 9 (the dependent variable) from independent variables such as controls 24, 4, 1, 27. Controls 4, 24 and 27 had an impact on risk 29. In addition, the multiple correlation value was 0.526 and the value of R^2 was 0.277. This was interpreted as a percentage of 27.7 % from the dependent variable of risk 29. Therefore, there was no multicollinearity

Table 31: Illustrates the coefficients and distributed T (Coefficients^a)

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	t	Sig.
1	(Constant)	2.915		6.413	0.000
	C29	0.310	0.407	3.828	0.000
2	(Constant)	2.137		3.969	0.000
	C29	0.239	0.314	2.877	0.005
	C4	0.235	0.273	2.500	0.015
3	(Constant)	1.160		1.690	0.095
	C29	0.212	0.279	2.592	0.012
	C4	0.216	0.252	2.356	0.021
	C24	0.240	0.226	2.211	0.030
4	(Constant)	1.298		1.932	0.057
	C29	0.274	0.359	3.230	0.002
	C4	0.324	0.376	3.164	0.002
	C24	0.289	0.272	2.672	0.009
	C5	-0.248	-0.281	-2.171	0.033

a. Dependent Variable: r8

Table 32: Illustrates the value of correlation

C4	C16	C19	C20	C24	C25
0.276*	0.261*	0.273*	0.226*	0.368*	0.288*
C26	C27	C28	C29	C30	
0.302**	0.332**	0.250*	0.269*	0.257*	

Table 33: Illustrates the value of correlation and R^2

Model	R	R^2	Durbin-Watson
1	0.368^a	0.136	
2	0.431^b	0.186	
3	0.485^c	0.235	
4	0.526^d	0.277	1.678

a. Predictors: (Constant), c24; b. Predictors: (Constant), c24, c4; c. Predictors: (Constant), c24, c4, c1 d. Predictors: (Constant), c24, c4, c1, c27

Table 34: Illustrates an analysis of variance (ANOVA^e)

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	5.521	1	5.521	11.623	0.001^a
	Residual	35.150	74	0.475		
	Total	40.671	75			
2	Regression	7.566	2	3.783	8.341	0.001^b
	Residual	33.105	73	0.453		
	Total	40.671	75			
3	Regression	9.576	3	3.192	7.391	0.000^c
	Residual	31.095	72	0.432		
	Total	40.671	75			
4	Regression	11.247	4	2.812	6.784	0.000^d
	Residual	29.424	71	0.414		
	Total	40.671	75			

a. Predictors: (Constant), c24; b. Predictors: (Constant), c24, c4; c. Predictors: (Constant), c24, c4, c1; d. Predictors: (Constant), c24, c4, c1, c2; e. Dependent Variable: r9 shown in the use of collinearity diagnosis (Tolerance test and VIF test). Furthermore, the Durbin-Watson statistic (D) was 1.678 and (d_U = 1.739, d_L = 1.515) based on K = 4, N = 76, at α = 0.05; as such, there was evidence of inconclusive ($d_L < D < d_U$: Inconclusive).

R10: Risk of 'misalignment of software project with local practices and processes' compared to controls: Table 36 to 39 show that the significant value is less than the assumed value whereby α = 0.05 level of significance, so there was a positive relation between control 3, 13, 14 and risk 10. As none of risk management techniques (controls) were significant, they were not reported. However, it was good that model 3 could predict a risk 10 (the dependent variable) from independent variables such as controls 13, 3, 8. Controls 3 and 13 had an impact on risk 30. In addition, the results showed that the multiple correlation value was 0.451 and the value of R^2 was 0.203. This was interpreted as a percentage of 20.3% from the dependent variable of risk 10. Therefore, there was no multicollinearity shown in the use of collinearity

Table 35: Illustrates the coefficients and distributed T (Coefficients[a])

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	t	Sig.
1	Constant	2.505		4.108	0.000
	C24	0.406	0.368	3.409	0.001
2	Constant	1.682		2.366	0.021
	C24	0.369	0.335	3.139	0.002
	C4	0.202	0.227	2.123	0.037
3	Constant	2.035		2.856	0.006
	C24	0.468	0.425	3.787	0.000
	C4	0.287	0.322	2.845	0.006
	C1	-0.261	-0.263	-2.158	0.034
4	(Constant)	1.661		2.298	0.024
	C24	0.402	0.365	3.213	0.002
	C4	0.256	0.286	2.552	0.013
	C1	-0.287	-0.289	-2.409	0.019
	C27	0.189	0.222	2.008	0.048

a. Dependent Variable: r9

Table 36: Illustrates the value of correlation

C3	C13	C14
0.293**	0.324**	0.309**

Table 37: Illustrates the value of correlation and R^2

Model	R	R^2	Durbin-Watson
1	0.324[a]	0.105	
2	0.391[b]	0.153	
3	0.451[c]	0.203	1.688

a. Predictors: (Constant), c13; b. Predictors: (Constant), c13, c3; c. Predictors: (Constant), c13, c3, c8

Table 38: Illustrates an analysis of variance (ANOVA[d])

Model		Sum of squares	df	Mean square	F	Sig.
1	Regression	3.373	1	3.373	8.671	0.004[a]
	Residual	28.785	74	0.389		
	Total	32.158	75			
2	Regression	4.929	2	2.464	6.607	0.002[b]
	Residual	27.229	73	0.373		
	Total	32.158	75			
3	Regression	6.533	3	2.178	6.119	0.001[c]
	Residual	25.625	72	0.356		
	Total	32.158	75			

a. Predictors: (Constant), c13; b. Predictors: (Constant), c13, c3; c. Predictors: (Constant), c13, c3, c8; d. Dependent Variable: r30

Table 39: Illustrates the coefficients and distributed T (Coefficients[a])

		Unstandardized coefficients	Standardized coefficients		
Model		B	β	t	Sig.
1	Constant	3.282		7.216	0.000
	C13	0.271	0.324	2.945	0.004
2	Constant	2.659		4.925	0.000
	C13	0.224	0.268	2.408	0.019
	C3	0.172	0.227	2.042	0.045
3	Constant	3.247		5.451	0.000
	C13	0.217	0.259	2.388	0.020
	C3	0.236	0.311	2.693	0.009
	C8	-0.171	-0.238	-2.123	0.037

a. Dependent Variable: r10

Table 40: Software risk factors in the design phase are mitigated by risk management techniques

No	Software design process issue factors	Risk control factors
1	Introduction of new technology.	C3: Assessing cost and scheduling the impact of each change to requirements and specifications.
2	Developing the wrong software functions and properties.	C4: Develop prototyping and have the requirements reviewed by the client.
3	Developing the wrong user interface.	-------
4	Insufficient procedures to ensure security, integrity and availability of the database.	C19: Provide scenarios methods and using of the reference checking, C24: Ensuring that quality-factor deliverables and task analysis.
5	Lack of integrity/consistency.	C24: Ensuring that quality-factor deliverables and task analysis, C27: Combining internal evaluations by external reviews.
6	Lack of architecture, performance, quality software project.	C26: Incremental development (deferring changes to later increments), C30: Participating users during the entire software project lifecycle.
7	Absence of quality architectural and design documents.	C2: Stabilizing requirements and specifications as early as possible, C15: Reusable user documents early.
8	Failure to design (blueprints) and redesign software processes.	C29: Provide training in the new technology and organize domain knowledge training, C4: Develop prototyping and have the requirements reviewed by the client, C24: Ensuring that quality-factor deliverables and task analysis, C5: Developing and adhering a software project plan.
9	Lack of effective software project team integration between clients, the supplier team and the supply chain.	C24: Ensuring that quality-factor deliverables and task analysis, C4: Develop prototyping and have the requirements reviewed by the client, C1: Using of requirements scrubbing, C27: Combining internal evaluations by external reviews.
10	Misalignment of software project with local practices and processes.	C13: Reusable test plans and test cases, C3: Assessing cost and scheduling the impact of each change to requirements and specifications, C8: Assigning responsibilities to team members and rotate jobs.

diagnosis (Tolerance test and VIF test). Furthermore, the Durbin-Watson statistic (D) was 1.688 and (d_U = 1.709, d_L = 1.543) based on K = 3, N = 76, at $\alpha = 0.05$; there is evidence of inconclusive ($d_L < D < d_U$: Inconclusive).

Software design risk factors identification checklists and control factors (risk management techniques): Table 40 shows software design process issue Factors identification checklist with risk controls based on a questionnaire of experienced software project managers. He can use the checklist on software projects to identify and mitigate software design risk factors on life cycle software projects by risk management techniques.

CONCLUSION

The concern of this study is the mitigating software design process issues by using approach modelling. Indeed, we referred to the positive relation between software issues and controls. Furthermore, we listed the issues that were controlled by using stepwise approach in Table 40. In the future work, we need to use more techniques and artificial optimal algorithms to mitigate and manage the issues in software projects in the real world like fuzzy logic and genetic algorithm, etc.

ACKNOWLEDGMENT

This study is supported by the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM) and Al-Aqsa University, Palestine.

REFERENCES

Addison, T., 2003. E-commerce project development risks: Evidence from a Delphi survey. Int. J. Inform. Manage., 23(1): 25-40.

Addison, T. and S. Vallabh, 2002. Controlling software project risks: An empirical study of methods used by experienced project managers. Proceeding of the 2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology (SAICSIT, 2002), pp: 128-140.

Ahmed, A., 2011. Software Project Management: A Process-driven Approach. CRC Press, Taylor and Francis Group, Auerbach Publications, pp: 456.

Aloini, D., R. Dulmin and V. Mininno, 2007. Risk management in ERP project introduction: Review of the literature. Inform. Manage., 44(6): 547-567.

Aritua, B., N.J. Smith and D. Bower, 2011. What risks are common to or amplified in programmes: Evidence from UK public sector infrastructure schemes. Int. J. Proj. Manag., 29(3): 303-312.

Bennatan, E.M., 2006. Catastrophe Disentanglement: Getting Software Projects Back on Track. Addison-Wesley, Upper Saddle River, NJ, pp: 288.

Boehm, B.W., 1991. Software risk management: Principles and practices. IEEE Software, 8(1): 32-41.

Boehm, B., 2002a. Boehm's Top 10 2002-Software Risks. The USC Center for Systems and Software Engineering. Retrieved form: http://csse.usc.edu/BoehmsTop10/2006/survey.php . (Accessed on: February 28, 2012)

Boehm, B., 2002b. Software Risk Management: Overview and Recent Developments. pp: 90.

Boehm, B., 2007. USC CSSE workshop overview: Top 3 software-intensive systems risk items. pp: 34.

Buckley, R. and J. Caple, 2009. The Theory & Practice of Training. 6th Edn., Kogan page London, Philadelphia. pp: 369.

Chen, H.G., J.J. Jiang, G. Klein and J.V. Chen, 2009. Reducing software requirement perception gaps through coordination mechanisms. J. Syst. Software, 82(4): 650-655.

Chen, J.C. and S.J. Huang, 2009. An empirical analysis of the impact of software development problem factors on software maintainability. J. Syst. Software, 82(6): 981-992.

Cliff Mitchell, 2011. Why Don't We Learn from Our Project Failures. Retrieved from: https://tm.mbs.ac.uk/SearchResults/tabid/56/ArticleID/72/Why-don%E2%80%99t-we-learn-from-our-project-failures.aspx. (Accessed on: March 1, 2012)

Dash, R. and R. Dash, 2010. Risk assessment techniques for software development. Eur. J. Sci. Res., 42(4): 629-636.

Elzamly, A. and B. Hussin, 2011a. Estimating quality-affecting risks in software projects. Int. Manage. Rev., American Scholars Press, 7(2): 66-83.

Elzamly, A. and B. Hussin, 2011b. Managing software project risks with proposed regression model techniques and effect size technique. Int. Rev. Comput. Software, 6(2): 250-263.

Elzamly, A. and B. Hussin, 2013a. Managing software project risks (implementation phase) with proposed stepwise regression analysis techniques. Int. J. Inform. Technol. (IREIT), 1(4): 300-312.

Elzamly, A. and B. Hussin, 2013b. Managing software project risks (design phase) with proposed fuzzy regression analysis techniques with fuzzy concepts. Int. Rev. Comput. Software, 8(11): 2601-2613.

Elzamly, A. and B. Hussin, 2014a. An enhancement of framework software risk management methodology for successful software development. J. Theor. Appl. Inform. Technol., 62(2): 410-423.

Elzamly, A. and B. Hussin, 2014b. Identifying and managing software project risks with proposed fuzzy regression analysis techniques: Maintenance phase. Proceeding of Conference on Management and Engineering (CME, 2014), pp: 14.

Elzamly, A. and B. Hussin, 2014c. Managing software project risks (planning phase) with proposed fuzzy regression analysis techniques with fuzzy concepts. Int. J. Inform. Comput. Sci. (IJICS), 3: 31-40.

Elzamly, A. and B. Hussin, 2014d. Evaluation of quantitative and mining techniques for reducing software maintenance risks. Appl. Math. Sci., 8(111): 5533-5542.

Elzamly, A., and B. Hussin, 2014f. Managing software project risks (analysis phase) with proposed fuzzy regression analysis modelling techniques with fuzzy concepts. J. Comput. Inf. Technol., 22(2): 131-144.

Elzamly, A. and B. Hussin, 2015a. Classification and identification of risk management techniques for mitigating risks with factor analysis technique in software risk management. Rev. Comput. Eng. Res., 2(1): 22-38.

Elzamly, A. and B. Hussin, 2015b. Modelling and evaluating software project risks with quantitative analysis techniques in planning software development. J. Comput. Inf. Technol., 23(2): 123-139.

Elzamly, A., B. Hussin, S.S. Abu Naser and M. Doheir, 2015a. Predicting software analysis process risks using linear stepwise discriminant analysis: Statistical methods. Int. J. Adv. Inform. Sci. Technol., 38(38): 108-115.

Elzamly, A., B. Hussin and N. Salleh, 2015b. Methodologies and techniques in software risk management approach for mitigating risks: A review. Asian J. Math. Comput. Res., 2(4): 184-198.

Ewusi-Mensah, K., 2003. Software Development Failures: Anatomy of Abandoned Projects. The MIT Press, Cambridge, MA, pp: 276.

Fairley, R.E., 2009. Managing and Leading Software Projects. Wiley, Hoboken, NJ, pp: 492.

Han, W.M. and S.J. Huang, 2007. An empirical analysis of risk components and performance on software projects. J. Syst. Software, 80(1): 42-50.

Hoffer, J., J. George and J. Valacich, 2011. Modern Systems Analysis and Design. 6th Edn., Prentice Hall, New York, pp: 575.

House of Commons, 2009. The national offender management information system. Fortieth Report of Session 2008–09, House of Commons Public Accounts Committee, Parliament, pp: 41.

Huang, S.J. and W.M. Han, 2008. Exploring the relationship between software project duration and risk exposure: A cluster analysis. Inform. Manage., 45(3): 175-182.

Jalote, P., 2008. A Concise Introduction to Software Engineering. Springer-Verlag, London Ltd., pp: 288.

Kaur, R. and N. Malhotra, 2013. Estimating the effects of developing wrong software functions using fuzzy cognitive map. Int. J. Comput. Appl., 70(18): 38-41.

Kettunen, P. and M. Laanti, 2005. How to steer an embedded software project: Tactics for selecting the software process model. Inform. Software Tech., 47(9): 587-608.

Khanfar, K., A. Elzamly, W. Al-Ahmad, E. El-Qawasmeh, K. Alsamara and S. Abuleil, 2008. Managing software project risks with the chi-square technique. Int. Manage. Rev., 4(2): 18-29.

Laplante, P.A., 2004. Real-Time Systems Design and Analysis. 3rd Edn., J. Wiley, Hoboken, New Jersey, pp: 505.

Lyons, T. and M. Skitmore, 2004. Project risk management in the Queensland engineering construction industry: A survey. Int. J. Proj. Manag., 22(1): 51-61.

Nakatsu, R.T. and C.L. Iacovou, 2009. A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study. Inform. Manage., 46(1): 57-68.

Pandian, C.R., 2007. Applied Software Risk Management: A Guide for Software Project Managers. Risk Management. Auerbach Publications is an Imprint of the Taylor and Francis Group, pp: 246.

Paré, G., C. Sicotte, M. Jaana and D. Girouard, 2008. Prioritizing clinical information system project risk factors: A delphi study. Proceeding of the 41st Hawaii International Conference on System Sciences, pp: 1-10.

Rudd, C., 2010. Planning to Implement Service Management (PISM). TSO (The Stationery Office), pp: 305.

Schmidt, R., K. Lyytinen, M. Keil and P. Cule, 2001. Identifying software project risks: An international delphi study. J. Manage. Inform. Syst., 17(4): 5-36.

Schwalbe, K., 2010. Information Technology Project Management. 6th Edn.,. Course Technology, Cengage Learning, Boston, MA, pp: 490.

Selby, R., 2007. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research (Practitioners). 1st Edn., Wiley-IEEE Computer Society, pp: 832.

Sodhi, J. and P. Sodhi, 2001. IT Project Management Handbook. Management Concepts, Vienna, VA, pp: 264.

Sumner, M., 2000. Risk factors in enterprise-wide/ERP projects. J. Inform. Technol., 15(4): 317-327.

Surie, D., 2004. Evaluation and integration of risk management in CMMI and ISO / IEC 15504. Proceeding of the 8th Student Conference in Computing Science, Umeå, Sweden, pp: 14.

Thomas, S. and H. Uwe, 2010. Agile Software Development: Best Practices for Large Software Development Projects. Springer-Verlag, Berlin, Heidelberg, pp: 203.

Yogesan, K., P. Brett and M. Gibbons, 2009. Handbook of Digital Homecare. Series in Biomedical Engineering. Springer-Verlag, Berlin, Heidelberg, pp: 363.