The Nucleus **48**, No. 2 (2011) 107-119

The Nucleus
A Quarterly Scientific Journal of Pakistan
Atomic Energy Commission
NCLEAM, ISSN 0029-5698

MEASUREMENT OF THE MATURITY OF THE SOFTWARE REQUIREMENTS CHANGE MANAGEMENT PROCESS

M. W. BHATTI and I.A. MANARVI

Engineering Management Department, Center for Advanced Studies in Engineering (CASE), Sector G-5/1, Islamabad, Pakistan

(Received April 29, 2011 and accepted in revised form May 19, 2011)

In this paper, an extensive analysis is performed to develop a research instrument to measure the maturity of the software requirements change management process. The items of the research instrument are extracted from the existing literature of the software requirements change management process. A pilot study was conducted on small sample size (46 responses) for basic analysis of the data. After pilot study, a comprehensive analysis is performed on the large sample (162 responses) of data. In detailed analysis, the reliability analysis of the instrument is performed by computing the value of the cronbach's alpha. The content validity of the instrument is performed by a comprehensive review of the existing literature, by discussion of the items with the domain experts, by conducting a pilot study and by taking feedback from the professionals of the industry. The construct validity of the instrument is analyzed by the correlation matrix of the items, the value of the determinant, KMO and Bartlett's Test, total variance through eigenvalues, scree plot and component matrix. The criterion validity analysis of the instrument is computed by measuring the correlation, adjusted R-square and F-value with the "organizational performance". The results of the analysis show that the selected items form a reliable and valid instrument for the measurement of the maturity of the software requirements change management process.

Keywords: Software requirements, Change management process, Research instrument

1. Introduction

The complete requirements of a software project can't be correctly understood during the first analysis of the requirements. The new requirements keep on adding and existing requirements keep on changing during the development of a software project [2]. The dynamic changes in technology and requirements affect the performance of the software engineering activities [3]. The changes in the requirements affect the progress control, cost analysis, and life cycle of a software project [4]. CMMI [5] helps in qualitative analysis of the software requirements change management process. Auditors have their templates and checklists to verify the practices of relevant process areas. However, for research purpose, it is difficult to analyze the impact of software requirements change management process on any other import variable of software engineering. Therefore, in this research paper, a research instrument is developed to analyze the maturity of the software requirements change management process. The items of the instrument are selected from the existing literature. After the selection of items, empirical investigation is performed for the reliability and validity of the instrument.

The requirements management process area and the requirements change requests span the complete life cycle of the development of a software project [6]. The requirements keep on changing, even when the testing of the developed software project is started [7]. If the requirements can change at any point during the development of a software project, then, for a mature software requirements change management process, the development team should be ready to accept the change requests during the entire life cycle of a software project. The item 1 (Table 1) of the instrument is developed, that asks about the acceptance of the requirements during the development of the software project. The item 1 states that, "Customer's requirements change requests are accepted during the development of the software".

* Corresponding author : mwasim_bhatti@yahoo.com

Process assessment and improvement is one of the major concerns of the software development organizations [8]. In software development, complexity increases because of the variety of technologies and dynamic customer demands [5, 9]. Therefore, a complete process for the management of changing requirements is necessary during the development of a software project. Several methodologies are proposed to manage the changing requirements of a software project [10]. Therefore, all change requests should be addressed through proper requirements change management process. Thus, the item 2 of the instrument states that, "Customer's requirements change requests are received on a standard change request form". The item 3 of the instrument states that, "Requirements change requests are processed through change management process".

The change impact analysis is an important activity of requirements change management process [11]. The dependency of changed requirements on unchanged requirements causes difficulties in analyzing the impact of existing requirements [4]. A change request is analyzed by different experts of project teams. After analyzing the change requests, the request is approved or disapproved [12]. Thus, the item 4 of the instrument states that, "The impact of change in requirements is analyzed on existing requirements of the software".

Requirement engineering is a decision-intensive process [13]. Different teams are involved in analyzing and deciding about the change requests [12]. The participation of the quality assurance team in the requirements change management process helps to ensure the correct execution of the process. Thus, the item 5 of the instrument states that, "Quality assurance team participates in software requirements change management process".

The change control board (CCB) is an organizational unit within the software development department. CCB evaluates the change requests in the requirements of a software project. After evaluation, CCB approves or disapproves the change requests [9]. Therefore, the evaluation of change requests by CCB is also an indicator of the maturity of the software requirements change management process. Thus, the item 6 states that, "Software requirements change requests are approved from Change Control Board (CCB)". After approval of the change requests, the project plan and related documents should be updated accordingly. Therefore, the item 7 states that, "If any change occurs in requirements management plan then project plan is updated accordingly".

An automated tool was proposed to analyze the changes and the impact of changes on the source code of a software project [14]. The tool also maintains the traceability of changes automatically. An agent-based system was proposed that can manage the change requests efficiently [12]. A tool was presented that supports the integrated functionality of software configuration management and traceability management [15]. This tool helps to process the change requests and helps to manage the traceability of the changes. Thus, any automated software or workflow application for the management of change requests is an indicator of the maturity of the software requirements change management process in a software development organization. Thus, the item 8 of the instrument states that, "Software requirements change management process is processed through any workflow application".

The history and the rationale of the changed requirements should be maintained [16]. The history of the changing requirements helps to quantify the percentage changes in the requirements of a software project and helps to take corrective actions for the future activities of the project. Thus, the item 9 of the instrument states that, "Requirement changes are stored to maintain the history of the change requests".

The software configuration management system and traceability are important practices that help the requirements change management process in a software development organization. A software configuration management system helps to manage the changes in the artifacts and documents of a software project [15]. The configuration management system maintains the integrity and traceability of the configuration items and controls the changes in requirements systematically [6]. Therefore, the history of the changes in the requirements should be maintained under the umbrella of configuration management system. The item 10 of the instruments states that, "History of changes is maintained under the configuration management control."

1.2. List of Items

The complete list of the items of the instrument that measures the maturity of the software requirements change management process is shown in Table 1.

2. Limitations

Following are some of the limitations of this research work:

1. The data is collected from software industry of Pakistan only.
2. The criterion validity is tested with only one external variable (i.e. organization performance).

3. Intellectual Contribution

The CMMI [5,27] qualitatively defines the software requirements change management process. In this research paper, an instrument is developed to measure the concept of software requirements change management process. The quantitative measurement of the concept is the intellectual contribution of this research.

4. Research Questions

Following research questions are addressed in this research paper:

1. Whether the ten items that were summed to measure the "Maturity of Software Requirements Change Management Process" score formed a reliable scale?
2. Whether the ten items that were summed to measure the "Maturity of Software Requirements Change Management Process" score formed a valid scale?.

5. Research Methodology

To develop an instrument for the measurement of the maturity of the software requirements change management process, the following research methodology was employed in the study.

A questionnaire survey was conducted in the software industry of Pakistan in two phases. In first phase, the developed questionnaire was distributed among 75 professionals in the software organizations of Islamabad, Rawalpindi, and Lahore. 46 usable questionnaires were returned, representing a 61.3% response rate. In second phase, the same questionnaire was sent to 300 professionals of the software industry of Pakistan. As a result 162 usable questionnaires were returned, representing a 54% response rate. The purpose of the first phase of data collection was to perform a pilot study for the development of a research instrument for the measurement of the maturity of the software requirements change management process. The analysis of pilot study results into significant findings about the reliability of the instrument. These findings encouraged the researchers to collect large set of data and to perform the detailed analysis on large sample size. Therefore, in second phase, reliability, validity and other statistical analysis of the instrument was analyzed on a sample size of 162 respondents.

Table 1. Items to measure the "Maturity of Software Requirements Change Management Process"

No.	Items
1	Customer's requirements change requests are accepted during the development of the software.
2	Customer's requirements change requests are received on a standard change request form.
3	Requirements change requests are processed through change management process.
4	The impact of change in requirements is analyzed on existing requirements of the software.
5	Quality assurance team participates in software requirements change management process.
6	Software requirements change requests are approved from Change Control Board (CCB).
7	If any change occurs in requirements management plan then project plan is updated accordingly.
8	Software requirements change management process is processed through any workflow application.
9	Requirement changes are stored to maintain the history of the change requests.
10	History of changes is maintained under the configuration management control.

The participation of respondents was voluntary and surveys were completed on respondents' own time. The population for this study comprised of business analysts, manager analysts, team leads, project managers, quality assurance and quality control managers, working in software industry of Pakistan.

The questionnaire consists of 10 items. Each item is an indicator of the maturity of the software requirement change management process. Responses were obtained by using a 5-point Likert-

type scale, where 1=Never, 2=Rarely, 3=Occasionally, 4=Often, and 5=Always. In phase 1, the reliability analysis of the instrument is performed by computing Cronbach's Alpha of the instrument. The descriptive statistics of the items was analyzed by computing mean and std. deviation of each item. The correlation matrix of all the items is formulated to analyze the relationship of items with each other.

The purpose of Phase 2 analysis was to compute the reliability and validity of the instrument with large data set. Therefore, in phase 2 of the analysis, the reliability analysis of the instrument was performed by computing the Cronbach's Alpha of the instrument. Corrected Item-Total Correlation of all the items was calculated to analyze the relationship of each item with the sum of the other items. Items with low value of Corrected Item-Total Correlation were removed from the list of items and reliability analysis was performed again. The basic statistics of the items was computed to analyze the mean and std. deviation of each item. The content validity, the construct validity and the criterion validity of the instrument are computed to analyze the overall validity of the instrument. The content validity of the instrument was performed by reviewing the comprehensive literature on the subject, by discussing the items of instrument with the domain experts and by getting feedback from the professionals. The construct validity of the instrument was analyzed by performing factor analysis, KMO and Bartlett's, component matrix and scree plot. The correlation matrix of all the items was formulated to analyze the relationship of the items with each other. KMO and Bartlett's Test was performed to identify the value of Kaiser-Meyer-Olkin Measure of Sampling Adequacy. In component matrix, the loadings of the items were analyzed and the dimensions of the instrument were identified. For the criterion validity of the instrument, organizational performance (Dyba, 2000) was used as an external variable. The correlation was computed between the maturity of the software requirements change management process and organizational performance by using the Pearson's r method. The adjusted R-square and the significance of F-value were also computed between the two variables.

6. Analysis of Data in Phase 1

The phase 1 of the data analysis is a pilot test for the development of an instrument for the measurement of the maturity of the software requirements change management process. In this phase, the data analysis is performed on a small sample size (46 respondents). The purpose of this pilot study is to perform the basic analysis of the data. So that we should be able to decide that whether we should collect large set of data for complete analysis of the instrument or we should stop the analysis due to poor results from the data analysis. It was also intended to get the suggestions and feedback from the professionals of the industry to improve the contents of the instrument. In this pilot test, the reliability analysis of the instrument is performed to measure the consistency and stability of the items of the instrument. Descriptive statistics of the items is performed to analyze the mean and std. deviation of the items of the instrument. Correlation matrix is computed to analyze the relationship of the items among each other. The detailed data analysis of the instrument in Phase 1 is as follows;

6.1. Reliability Analysis of the Instrument

Reliability refers to the consistency and stability of an instrument [1]. To assess whether the ten items that were summed to measure the "Maturity of Software Requirements Change Management Process" score formed a reliable scale, Cronbach's Alpha [17] was computed. The Cronbach's Alpha is defined as;

$$\alpha = \frac{K}{K-1}\left(1 - \frac{\sum_{i=1}^{K} \alpha_{Y_i}^2}{\alpha_x^2}\right)$$

The value of Cronbach's Alpha for the ten items at sample size of 46 was 0.855, which indicates that the items form a scale that has good internal consistency reliability.

6.2 Descriptive Statistics

The Table 2 shows the descriptive statistics for each item of the "Maturity of Software Requirements Change Management Process" instrument. The Mean value 3.8 of the first item shows that maximum respondents believe that the customer's requirement change requests are accepted during the development of a software project. The Mean value 2.2 of the 8th item depicts

Table 2. Descriptive Statistics of the Items of the Instrument

Descriptive Statistics			
No.	Items	Mean	Std. Deviation
1	Customer's requirements change requests are accepted during the development of the software	3.80	.910
2	Customer's requirements change requests are received on a standard change request form.	2.48	1.243
3	Requirements change requests are processed through change management process.	2.70	1.348
4	The impact of change in requirements is analyzed on existing requirements of the software.	3.33	1.351
5	Quality assurance team participates in software requirements change management process.	2.63	1.254
6	Software requirements change requests are approved from Change Control Board (CCB).	2.39	1.483
7	If any change occurs in requirements management plan then project plan is updated accordingly.	3.24	1.320
8	Software requirements change management process is processed through any workflow application.	2.20	1.392
9	Requirement changes are stored to maintain the history of the change requests.	3.65	1.215
10	History of changes is maintained under the configuration management control.	2.72	1.544

that less number of software development organizations use any workflow application to process the software requirements change requests. The value of the Std. Deviation of first item is the minimum value among the Std. Deviation values of all items of the instrument. This value (i.e. 0.91) shows that all respondents have very convergent views about the opinion that the customer's requirement change requests are accepted during the development of a software project. The 10^{th} item of the instrument has the maximum value of the Std. Deviation. The value 1.544 shows that respondents have less convergent views about maintaining the history of the changes under the configuration management control. This means that some of the software development organizations maintain changes under configuration management control on regular basis, while some organizations never follow this practice.

6.3. Correlation Matrix

The correlation matrix of all items of the instrument is shown in Table 3. The correlation of the items is computed to assess the inter-item relationship of the instrument. The relationship among all the items of the instrument is calculated by using the Pearson's r method. The Item 1 has a significant relationship with Item 7 and Item 10 at the 0.05 level of significance. The Item 2 has a significant relationship with the Item 3, Item 5, Item 6, Item 8, and Item 9 at the 0.01 level of significance. The Item 2 has a significant relationship with the Item 10 at the 0.05 level of significance. The Item 3 has a significant relationship with the Item 4, Item 5, Item 6, Item 8, Item 9, and Item 10 at the 0.01 level of significance. The Item 4 has a significant relationship with the Item 5, Item 7, and Item 9 at the 0.01 level of significance. The Item 4 has a significant relationship with the Item 10 at the 0.05 level of significance. The Item 5 has a significant relationship with the Item 6, Item 8, Item 9, and Item 10 at the 0.01 level of significance. The Item 5 has a significant relationship with the Item 7 at the 0.05 level of significance. The Item 6 has a significant relationship with the Item 8 at the 0.01 level of significance. The Item 6 has a significant relationship with the Item 7, Item 9, and Item 10 at the 0.05 level of significance. The Item 7 has a significant relationship with the Item 9 at the 0.01 level of significance. The Item 8 has a significant relationship with the Item 9 at the 0.05 level of significance. The Item 9 has a significant relationship with the Item 10 at the 0.05 level of significance.

The significant correlation among the items of the instrument indicates that the items are highly correlated and associated with each other and will probably be grouped among each other by the factor analysis.

Table 3. Correlation Matrix of the Items of the Instrument.

Items	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
Customer's requirements change requests are accepted during the development of the software										
Customer's requirement change requests are received on standard change request form.	-.073									
Requirements change requests are processed through change management process.	-.032	.686**								
The impact of change in requirements is analyzed on existing requirements of the software.	.198	.249	.434**							
Quality assurance team participates in software requirements change management process.	.247	.487**	.695**	.401**						
Software requirements change requests are approved from Change Control Board (CCB).	-.024	.644**	.706**	.290	.641**					
If any change occurs in requirements management plan then project plan is updated accordingly.	.336*	.254	.442**	.379**	.296*	.337*				
Software requirements change management process is processed through any workflow application.	-.004	.381**	.423**	.166	.564**	.522**	.192			
Requirement changes are stored to maintain the history of the change requests.	.219	.436**	.612**	.409**	.395**	.349*	.594**	.304*		
History of changes is maintained under the configuration management control.	.339*	.315*	.395**	.365*	.599**	.350*	.263	.274	.325*	

*.Correlation is significant at the 0.05 level (2-tailed).

**.Correlation is significant at the 0.01 level (2-tailed).

Table 4. Item-Total Statistics of the Items of the Instrument (10 items).

Items	Corrected Item-Total Correlation	Cronbach's Alpha if Item Deleted
Customer's requirements change requests are accepted during the development of the software	-.080	.890
Customer's requirements change requests are received on standard change request form.	.594	.849
Requirements change requests are processed through change management process.	.806	.831
The impact of change in requirements is analyzed on existing requirements of the software.	.567	.851
Quality assurance team participates in software requirements change management process.	.744	.836
Software requirements change requests are approved from Change Control Board (CCB).	.721	.837
If any change occurs in requirements management plan then project plan is updated accordingly.	.591	.849
Software requirements change management process is processed through any workflow application.	.541	.854
Requirement changes are stored to maintain the history of the change requests.	.630	.847
History of changes is maintained under the configuration management control.	.579	.851

7. Analysis of Data in Phase 2

The detailed data analysis of the instrument in Phase 2 is as follows;

7.1. Reliability Analysis of the Instrument

To assess whether the ten items that were summed to measure the "Maturity of Software Requirements Change Management Process" score formed a reliable scale at sample size of 162, Cronbach's Alpha was computed.

7.2. Cronbach's Alpha (10 Items)

The value of Cronbach's Alpha for the ten items at sample size of 162 was 0.864, which indicates that the ten items form a scale that has good internal consistency reliability.

7.3. Item-Total Statistics

In Table 4, the column "Corrected Item-Total Correlation" shows the correlation of each specific item with the total of the other items in the scale.

Other than first item of Table 4 ("Customer's requirements change requests are accepted during the development of the software"), the correlation of all items is moderately high to high (e.g., 0.40+). This means that other than first item, all items have moderately high to high correlation. Item 1 doesn't fit into this scale, because, it has a lower Corrected Item-Total Correlation. Therefore, after deleting the Item 1 from the list, the items 2-10 will make a good component of a summated rating scale.

The last column of the Table 4 shows the value of the Cronbach's alpha, if we delete a particular item from the scale. From this column we can see that the Cronbach's alpha increases a little if Item 1 is deleted from the scale. By deleting any other item from the scale the Cronbach's alpha goes down from its original value.

Now we have two reasons to delete the Item 1 from the scale. The first reason is that the Corrected Item-Total Correlation of the Item 1 is very low (e.g., -0.080) and the value of alpha increases a little if we delete the Item 1 from the scale. Therefore, we deleted the first item from the initial list of the instrument. Now the further analysis will be based upon the remaining nine items of the instrument.

7.4. Cronbach's Alpha (9 Items)

The value of Cronbach's Alpha for the nine items was 0.890. This value indicates that the nine items form a scale that has good internal consistency reliability.

Table 5. Descriptive Statistics of the Items of the Instrument

Items	Mean	Std. Deviation
Customer's requirements change requests are received on standard change request form.	2.76	1.310
Requirements change requests are processed through change management process.	2.81	1.337
The impact of change in requirements is analyzed on existing requirements of the software.	3.39	1.264
Quality assurance team participates in software requirements change management process.	2.78	1.355
Software requirements change requests are approved from Change Control Board (CCB).	2.56	1.452
If any change occurs in requirements management plan then project plan is updated accordingly.	3.10	1.314
Software requirements change management process is processed through any workflow application.	2.49	1.410
Requirement changes are stored to maintain the history of the change requests.	3.80	1.159
History of changes is maintained under the configuration management control.	3.04	1.541

7.5. Descriptive Statistics

The Table 5 shows the descriptive statistics for nine items of the "Maturity of Software Requirements Change Management Process" instrument. The Mean value 3.8 of the 8^{th} item shows that the maximum respondents believe that the requirements should be stored to maintain the history of the change requests. The Mean value 2.49 of the 7^{th} item depicts that less number of software development organizations use any workflow application to process the software requirements change requests. The value of the Std. Deviation of 8^{th} item is the minimum value among the Std. Deviation values of all items of the instrument. This value (i.e. 1.159) shows that all respondents have very convergent views about the opinion that the requirements should be stored to maintain the history of the change requests. The 9^{th} item of the instrument has the maximum value of the Std. Deviation. The value 1.541 shows that respondents have less convergent views about maintaining the history of the changes under the configuration management control. This means that some of the software development organizations maintain changes under configuration management control on regular basis, while some organizations never follow this practice.

7.6. Validity Analysis of the Instrument

For an instrument, the reliability is the precondition for the validity of the instrument. Reliability refers to the consistency, while the validity refers to the accuracy of the instrument. An instrument is valid, if it measures what it is supposed to measure [1]. To test the validity of the instrument, content validity and construct validity analysis are performed.

7.7. Content Validity of the Instrument

Content validity is concerned with the degree to which the items of the instrument represent the domain of the concept under study. Content validation is based upon the systematic examination of the contents or items of the instrument [18]. Following steps are performed for the content validation of the instrument of the maturity of the software requirements change management process.

1. A comprehensive review of the existing literature of the software requirements change management process is performed for the selection of the items of the instrument.
2. Selected items are discussed with the domain experts of the software requirements change management process.
3. Before collection of large set of data, a pilot study is conducted to validate the contents of the instrument on small set of data and to get the feedback from the professionals of the industry.

Table 6. Correlation Matrix of the 9 Items

Items	1	2	3	4	5	6	7	8	9
Customer's requirements change requests are received on standard change request form.									
Requirements change requests are processed through change management process.	.640**								
The impact of change in requirements is analyzed on existing requirements of the software.	.318**	.542**							
Quality assurance team participates in software requirements change management process.	.453**	.695**	.531**						
Software requirements change requests are approved from Change Control Board (CCB).	.639**	.781**	.458**	.701**					
If any change occurs in requirements management plan then project plan is updated accordingly.	.364**	.537**	.490**	.448**	.449**				
Software requirements change management process is processed through any workflow application.	.436**	.473**	.258**	.531**	.520**	.310**			
Requirement changes are stored to maintain the history of the change requests.	.451**	.592**	.442**	.450**	.401**	.570**	.354**		
History of changes is maintained under the configuration management control.	.384**	.479**	.361**	.516**	.395**	.340**	.455**	.448**	

*.Correlation is significant at the 0.05 level (2-tailed).

**.Correlation is significant at the 0.01 level (2-tailed).

Therefore, we argue that the research instrument (the maturity of the software requirements change management process) has content validity. This means that the items of the instrument represent the domain of the maturity of the software requirements and change management process.

7.8. Construct Validity of the Instrument

Construct validity examines whether the individual items of the instrument measure the same thing that the complete instrument measures [19]. Factor analysis is one of the powerful methods of analyzing the construct validity of an instrument [20]. The principal component analysis with varimax rotation was used to analyze the construct validity and to assess the underlying structure for the nine items of the instrument of the maturity of the software requirements change management process. For the construct validity of the instrument a detailed item analysis is performed. Following statistical methods are applied to analyze the construct validity of the instrument; the correlation matrix of the items, the value of the determinant, KMO and Bartlett's Test, total variance [21,26] through eigenvalues, scree plot [22] and component matrix.

7.8.1. Correlation Matrix

The correlation matrix of nine items of the instrument is shown in Table 6. The relationship among all the items of the instrument is calculated by using the Pearson's r method. The Table 6 shows that all the items of the instrument are highly correlated with each other. These all items will probably be grouped with each other by the factor analysis.

7.8.2. Determinant

The value of Determinant is 0.09. This value indicates that the collinearity of the items of the instrument is not high. This means that a factor analytical solution can be obtained from the selected items.

7.8.3. KMO and Bartlett's Test

The value of Kaiser-Meyer-Olkin Measure of Sampling Adequacy is 0.883. This value indicates that there are sufficient items in each factor of the instrument. The Bartlett's Test of Sphericity is

Table 7. Total Variance Explained by 9 items

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	4.854	53.938	53.938	4.854	53.938	53.938
2	.937	10.412	64.351			
3	.750	8.330	72.681			
4	.679	7.541	80.222			
5	.543	6.029	86.251			
6	.415	4.613	90.864			
7	.390	4.332	95.196			
8	.251	2.788	97.983			
9	.182	2.017	100.000			
Extraction Method: Principal Component Analysis.						

significant at 0.000. This indicates that the correlation matrix is significantly different from an identity matrix, in which correlation between variables are all zero. This significance also means that the items of the instrument are correlated highly enough to provide a reasonable basis for factor analysis.

7.8.4. Total Variance Explained

Eigenvlaues rule is most widely used for the factors extraction of an instrument [21]. The eigenvalue rule is based upon retaining the components that have their eigenvalues greater than 1. The Table 7 shows how the variance is divided among the 9 possible components. The principal component analysis is used as the extraction method to analyze the variance explained by each component. The first component has eigenvalue greater than 1. This means that only first component is useful. The Table 7 shows that 53.93 % variance is accounted for by the first component.

7.8.5. Scree Plot

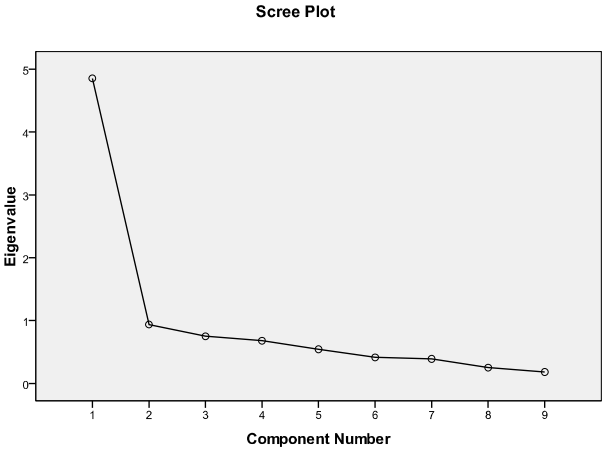

Figure 1. Scree Plot for 9 factors

Scree plot helps in identifying the number of components of an instrument [22]. The scree plot in Figure 1 shows that after first component, the eigenvalues decline, and they are less than 1.0. Both the scree plot and the eigenvalues support the conclusion that these nine variables can be grouped into one component.

Table 8. Component Matrix of 9 items .

Items	Loadings for Component 1
Customer's requirements change requests are received on standard change request form.	.707
Requirements change requests are processed through change management process.	.881
The impact of change in requirements is analyzed on existing requirements of the software.	.661
Quality assurance team participates in software requirements change management process.	.815
Software requirements change requests are approved from Change Control Board (CCB).	.822
If any change occurs in requirements management plan then project plan is updated accordingly.	.685
Software requirements change management process is processed through any workflow application.	.642
Requirement changes are stored to maintain the history of the change requests.	.702
History of changes is maintained under the configuration management control.	.653

7.8.6. Component Matrix

The Table 8 shows the Component Matrix of nine items. The component matrix is computed by using the Principal Component Analysis method of extraction. The table shows that the items cluster into one group defined by high loadings. This means that only one component is extracted from the nine items of the instrument. If the loadings of an item is greater than 0.45, it will be considered fair, if the loadings is greater than 0.55, it will be considered as good, if the loadings is greater than 0.63, it will be considered very good, and if the loading is greater than 0.71, it will be considered as excellent [23]. If the sample size is less than 150, then the loading of each item should be greater than 0.6. Each item of the instrument has a loading of .60 or higher in single component [24]. This means that all items have very good loadings in single component; therefore, the maturity of the software requirements change management process instrument is a one-dimensional instrument.

7.9. Criterion Validity of the Instrument

In criterion validity, the instrument is validated with any relevant criterion variable. The criterion validity is also called the external validity, because in criterion validity, the instrument is validated with some external variable [1]. Organizational performance is the ultimate criterion for the maturity of any process in a software development organization [25]. Therefore, the organizational performance is considered a criterion variable to test the criterion validity of the maturity of the software requirements change management process instrument. Organizational performance is measured by using the organizational performance instrument, developed by [25]. The five items of the instrument were modified with respect to the maturity of the software requirements change management process. The modified list of items is shown in Table 9.

Table 9. Items to measure the Organizational Performance (Dyba, 2000)

No.	Items
1	Software requirements change management process has substantially increased our software engineering competence.
2	Software requirements change management process has substantially improved our overall performance.
3	Over the past 3 years, we have greatly reduced the cost of software development.
4	Over the past 3 years, we have greatly reduced the cycle time of software development.
5	Over the past 3 years, we have greatly increased our customer's satisfaction.

The respondents were asked to fill the questionnaire by using a 5-point Likert-type scale, where "1=Strongly disagree", "2=Disagree", "3=Neither agree nor disagree", "4=Agree", and "5=Strongly agree". The reliability coefficient of the organizational performance instrument was 0.75.

This indicates that the organizational performance instrument has good internal consistency reliability.

To measure the criterion validity of the maturity of the software requirements change management process, the correlation by using the Pearson's r method was computed between the maturity of the software requirements change management process and the organizational performance. The coefficient of correlation between the two variables is 0.533. This relationship is significant at the 0.01 level (1-tailed). The adjusted R-square 0.279 indicates that 27.9 percent of the variance in the organizational performance is because of the maturity of the software requirements change management process. The F-value 63.36 was highly significant at 0.000. This indicates that the maturity of the software requirements change management process significantly predicts the organizational performance of a software development organization. The results of the analysis show that the instrument has good criterion validity.

8. Findings

1. The ten items that were summed to measure the "Maturity of Software Requirements Change Management Process" score formed a reliable scale.
2. The ten items that were summed to measure the "Maturity of Software Requirements Change Management Process" score formed a valid scale.
3. The instrument is a single-dimensional scale.
4. The external validity analysis shows that the developed instrument can be tested with other variables of software engineering.
5. The instrument supports the concept the different variables of software engineering can be quantified.

9. Conclusion

The detailed analysis of the items of the instrument shows that, these items form a reliable and valid instrument to measure the maturity of the software requirements change management process. The value of the Cronbach's Alpha (0.890) shows that the nine items of the instrument has good internal consistency. The correlation of the items of the instrument with the total of the other items of the instrument is moderately high to high (e.g., 0.50+). The content validity analysis shows that the items of the instrument represent the domain of the maturity of the software requirements and change management process. The results of the correlation matrix of the items, the determinant, KMO and Bartlett's Test, eigenvalues, scree plot and component matrix shows that the instrument has the construct validity. The eigenvalues shows that 53.93 % variance is accounted for by the first component of the instrument. The component matrix shows that the items cluster into one group, defined by high loadings. This means that only one component is extracted from the nine items of the instrument. The significant correlation, adjusted R-square and F-value between the maturity of the software requirements change management process and the organizational performance shows that the instrument has good criterion validity. Thus, we have a reliable and valid instrument to measure the maturity of the software requirements change management process.

References

[1] A. Anastasi and S. Urbina, *Psychological Testing*, Seventh edition, Upper Saddle River, New Jersey: Prentice-Hall, (1997).

[2] Q. Wang, and X. Lai, Requirement Management for the Incremental Development Model, Proceedings of the Second Asia-Pacific Conference on Quality Software (2001).

[3] M.S. Krishnan and M.I. Kellner, IEEE Transactions on Software Engineering **25**, No. 6. (1999).

[4] H.B. Yang, Z.H. Liu and Z.H. Ma, Journal of Information and Computing Science **2**, No. 1, (2007) 48.

[5] SEI, CMMI Product Team, CMMI for development, version 1.2. tech. report CMU/SEI-2006-TR-008 (2006).

[6] P. Bourque, R. Dupuis and A. Abran, IEEE Software **16**, No. 6 (1999).

[7] C. Kaner and W.P. Bond, Software Engineering Metrics: What Do They Measure and How Do We Know? 10th International software metrics symposium (2004).

[8] J.J.M. Trienekens, R.J. Kusters, D. Kriek and P. Siemons, Software Quality Journal **17**, No. 3 (2009) 231.

[9] J. Samalikova, R. Kusters, J. Trienekens, T. Weijters and P. Siemons, Software Quality Journal, Springer (2010).

[10] M.W. Bhatti, N. Ehsan, A. Ishaque, F. Hayat, S. Ahmed and E. Mirza, A Methodology to Manage the Changing Requirements of a Software Project, CISIM 2010, International Conference on Computer Information Systems and Industrial Management Applications, Cracow, Poland, (2010).

[11] P. Rovegard, L. Angelis and C. Wohlin, IEEE Transactions on Software Engineering **34**, No. 4 (2008).

[12] D. Habhouba, A. Desrochers and S. Cherkaoui, Agent-Based Assistance for Engineering Change Management: An Implementation Prototype, Proceedings of the 2009 13th International Conference on Computer Supported Cooperative Work in Design, (2009).

[13] A. Aurum and C. Wohlin, Information and Software Technology **45,** No. 14 (2003) 945.

[14] M.I. Collard and J.I. Maketic, Automatically identifying changes that impact code-to-design traceability during evolution, Software Quality Journal, Springer (2010).

[15] K. Mohan, P. Xu, L. Cao and B. Ramesh, Decision support system **45**, No. 4 (2008) 922.

[16] SEI, Product and Development Team, Capability Maturity Model Integration (CMMI), version 1.1, Continuous Representation, tech. report CMU/SEI-2002-TR-011, Carnegie Mellon University, Pittsburgh (2002).

[17] L.J. Cronbach, *Psychometrica* **16** (1951) 297.

[18] D. Davis, Business Research for Decision Making, Fourth Edition, Belmont, California: Duxbury Press (1996).

[19] J.C. Nunnally and I.A. Bernstein, Psychometric Theory, Third edition, New York: McGraw-Hill (1994).

[20] F. Kerlinger, Foundations of Behavioral Research, Holt, Rinehart and Winston (1986).

[21] H.F. Kaiser, Educational and Psychological Measurements **20** (1960) 141.

[22] R.B. Cattell, Multivariate Behavioral Research **1** (1966) 245.

[23] A. Comrey, A First Course on Factor Analysis, London: Academic Press (1973).

[24] J. Stevens, Applied Multivariate Statistics for the Social Sciences, London: Lawrence Erlbaum (1992).

[25] T. Dyba, Emperical Software Engineering **5** (2000) 357.

[26] H.F. Kaiser, Psychometrika **35** (1970) 401.

[27] SEI, Product and Development Team, Capability Maturity Model Integration (CMMI), Version 1.1, Staged Representation, tech. report CMU/SEI-2002-TR-012, Carnegie Mellon University, Pittsburgh (2002).