International Journal of Advances in Software Engineering & Research Methodology– IJSERM
Volume 2: Issue 2 **[ISSN : 2374-1619]**
Publication Date : 30 October, 2015

Study on the Risk Factors during Component Integration in Component Based System Development

Marjan Goodarzi
Nazean Jomhari
M. R. Safaei
Mahidzal Dahari

Abstract— ***Component integration is commonly identified as a process that plays a crucial role in the whole development of Component Based System (CBS). Because of its integration centric nature, CBS integration comes with high risk, especially because of the rare chance of perfect component match and readiness for 'plug and play'. There has been a lack in detailed component documentation which is a major focus in CBS development because of its profound effect on the phase of integration in the lifecycle of CBS development. It is challenging for system integrators to find out the component capabilities, if the components are not properly documented. This study aids in demonstrating the risk factors during integration of components. This aids in identifying the gaps in the existing component documentation and also in identifying crucial risk factors that can take place during the process of component integration. In this study, we report an industrial survey's results done among system integrators for analyzing the risk factors during component integration as well as the effect of in the CBS.***

Keywords—Risk Factors; Component Integration; Component Based System Development.

I. Introduction

Component Based System (CBS) development is complicated and prone to risk [1] process that has to undergo careful assessment for risk areas with the help of a system integrator for attaining possible gains of lowered time to market, enhanced productivity and establishment of a quality system [2]. It does not matter which tools, methods or techniques are used for CBS development when it comes to its risk proneness. Kotonya et al. [3] identified lack of source code and less or unknown design related information as well as disparity in the cycles of component evolution as the major risk areas in the integration phase. A fault in integration can be because of the improper or incorrect understanding of a component or it may be because of the components that are externally acquired [4]. In a similar way Rashid et al. [5] pointed out the significance of a good understanding of a component for the process of deployment and integration. Rashid mentioned that successful integration takes place only when a component has adequate documentation in a standardized manner and also when documentation of its usage history, details on the current version and test reports are available. Also, lack in the standardization of documents and procedures to review quality are crucial risks for a CBS integration process.

There are many risks and challenges with regards to the component's phase of integration. We got to find out various factors associated with risk which can result in failure during the phase of component integration [6, 7, 8, 3, 9, 10]. System integrators got to manage some information for ensuring that component is successfully integrated and the system is free of risks.

The objective of this study is to explore the risk factors that are faced by system integrators during the phase of component integration. It is crucial that we gain information regarding negatives elements in integration from that point of view of a system integrator for avoiding any risks related to integration. We got to identify as well as correlate these factors of risk and their effect during the phase of integration.

Till date, empirical studies on CBS has been focusing on identifying the risks related to identification of components, their selection as well as processes for maintaining them. Anyhow, to our best knowledge no works explores the risk factors in the phase of integration so far, that too form the perspective of a system integrator, in the lifecycle of CBS development.

Marjan Goodarzi
Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya
Kuala Lumpur, Malaysia

Nazean Jomhari
Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya
Kuala Lumpur, Malaysia

M. R. Safaei
Young Researchers and Elite Club, Mashhad Branch
Islamic Azad University
Mashhad, Iran

Mahidzal Dahari
Department of Mechanical Engineering, Faculty of Engineering
University of Malaya
Kuala Lumpur, Malaysia

II. Research Question and Hypothesis

We wish to get information regarding the risk elements that affects the integration process. It has been understood from this study that risk identifications in early phases can reduce the extent of effect during the process of component integration. Figure 1 illustrates the theoretical model designed for this. The main aim of this model is to address the below Research Question (RQ).

RQ1: What are the possible risks/problems during component integration phase?

The hypotheses to access the impact of risk factors during component integration is as follows.

HN1: There is no correlation between "Lack of Quality" and "System trends to risk"

HA1: There is a correlation between "Lack of Quality" and "System trends to risk"

HN2: There is no correlation between "Insufficient Testing" and "System trends to risk"

HA2: There is a correlation between "Insufficient Testing" and "System trends to risk"

HN3: There is no correlation between "Unavailability of source code" and "System trends to risk"

HA3: There is a correlation between "Unavailability of source code" and "System trends to risk"

HN4: There is no correlation between "Lack of Component Certification" and "System trends to risk"

HA4: There is a correlation between "Lack of Component Certification" and "System trends to risk"

HN5: There is no correlation between "Lack of System Interoperability" and "System trends to risk"

HA5: There is a correlation between "Lack of System Interoperability" and "System trends to risk"

HN6: There is no correlation between "Lack of Information for Version Control" and "System trends to risk"

HA6: There is a correlation between "Lack of Information for Version Control" and "System trends to risk"

Figure 1: Theoretical Research Model

III. Data and Result Analysis

In this part, we showcase the results in a tabular form together with description. The Pearson Correlation, Partial Least Square (PLS), Spearman Correlation techniques have been used for analyzing the gathered data for validating each hypotheses which were presented earlier in this section [2]. SPSS was used in these statistical calculations.

A. Hypotheses testing phase-I

For testing H1-H6 hypotheses, we examined Pearson Correlation Coefficient [2] among various elements of the research model illustrated in Figure 1. The result of the statistical calculations for Pearson Correlation Coefficient is shown in Table 1. Between "system trends to risk" and "lack of Quality", the Pearson correlation coefficient was .50 which is a weak yet positive relation, at $P < 0.05$ which gave justification for accepting HA1 alternate hypothesis and rejecting HN1 null hypothesis. HA2 alternate hypothesis was accepted as per the Pearson correlation coefficient of .62 which is a weak yet positive relation at $P < 0.05$ among "system trends to risk" and "insufficient testing". 0.50 was the correlation coefficient which as weak yet positive relation at $P < 0.05$ between "system trends to risk" and "unavailability of source code" which justified acceptance of HA3 alternate hypothesis. 0.52 was the Pearson correlation coefficient between "system trends to risk" and "lack of components certification" at $P < 0.05$ which was a weak positive relation that justified the acceptance of alternate hypothesis HA4 and rejects the HN4 null hypothesis. 0.41 the correlation coefficient is a weak positive relation at $P < 0.05$ seen between "system trends to risk" and "lack of system interoperability" which justified acceptance of HA5 alternate hypothesis and rejection of null hypothesis HN5. 0.01 was the correlation coefficient at $P < 0.05$ seen between "system trends to risk" and "lack of information for version control" which justified the acceptance of HN6 null hypothesis and rejection of alternate hypothesis.

As such it was seen and reported that H6 null hypothesis is accepted as alternate hypothesis is rejected, whereas the hypotheses H1, H2, H3, H4, H5 were found to be relevant and have weak though positive relation.

Table 1: Hypotheses testing using Pearson Correlation Coefficient

H	Variable 1	Variable 2	Coefficient	p-value
H1	Lack of Quality	System trends to risk	0.50	0.000
H2	Insufficient Testing	System trends to risk	0.62	0.000
H3	Unavailability of source code	System trends to risk	0.50	0.000
H4	Lack of Component Certification	System trends to risk	0.52	0.000
H5	Lack of System Interoperability	System trends to risk	0.41	0.002
H6	Lack of Information for Version Control	System trends to risk	0.01	0.915

B. Hypotheses testing phase-II

In this stage we did non-parametric statistics via Spearman Correlation Coefficient [2] for testing H1-H6 hypotheses. Table 2 reports the findings from testing stage. Between "system trends to risk" and "lack of quality", the Spearman coefficient was .50 which indicates a weak positive link, at $P < 0.05$ which gave a justification for accepting alternate hypothesis HA1 as well as reject the HN1 null hypothesis. HA2 the alternate hypothesis was accepted as per the Spearman coefficient of correlation being .62 which is a weak positive link at $P < 0.05$ between "insufficient testing" and "system trends to risk". 0.54 the correlation coefficient which is also a weak positive link at $P < 0.05$ was seen among "system trends to risk" and "unavailability of source code" which gave a justification for accepting alternate hypothesis HA3. Between "system trends to risk" and "lack of component certification" was .48 which is a weak positive link at $P < 0.05$ which gave justification for accepting alternate hypothesis HA4 as well as to reject HN4 null hypothesis. 0.40 the spearman correlation coefficient among "system trends to risk" and "lack of system interoperability" at $P < 0.05$ is a weak positive link which justified rejection of HN5 null hypothesis and acceptance of HA5 alternate hypothesis. 0.02, very weak association was the correlation found at $P < 0.05$ between "system trends to risk" and "lack of information for version control" which justified rejection of alternate hypothesis and acceptance of HN6 null hypothesis.

As such it was seen and reported that H6 null hypothesis is accepted as alternate hypothesis is rejected, whereas the hypotheses H1, H2, H3, H4, H5 were found to be relevant and have weak though positive relation.

Table 2: Hypotheses testing using Spearman rank-order correlation coefficient

H	Variable 1	Variable 2	Coefficient	p-value
H1	Lack of Quality	System trends to risk	0.50	0.000
H2	Insufficient Testing	System trends to risk	0.62	0.000
H3	Unavailability of source code	System trends to risk	0.54	0.000
H4	Lack of Component Certification	System trends to risk	0.48	0.000
H5	Lack of System Interoperability	System trends to risk	0.40	0.001
H6	Lack of Information for Version Control	System trends to risk	0.02	0.445

C. Hypotheses testing phase-III

In Phase III of testing of hypothesis, Partial Least Square (PLS) regression method was used for overcoming some limitations and also to cross validate the results gained during approaches used in the Phases I and II [2]. PLS test results are shown in Table3 and Table4 indicates the results' cross validation. Between "system trends to risk" and "lack of quality", the PLS coefficient was .50 which is a weak yet positive relation at $P < 0.05$ which justified the acceptance of HA1 alternate hypothesis and HN1 null hypothesis rejection. As 0.62 PLS coefficient between "system trends to risk" and "insufficient testing" is a weak yet positive relation at $P < 0.05$, HA2 alternate hypothesis was accepted. .50 at $P < 0.05$ is the weak yet positive relation indicating coefficient between "system trends to risk" and "Unavailability of source code" which justified the HA3 alternate hypothesis acceptance. 052 was the weak yet positive correlation coefficient between "system trends to risk" and "lack of component certification" which justified the acceptance of HA4 alternate hypothesis and rejection of HN4 null hypothesis. 0.41 is the PLS coefficient between "system trends to risk" and "lack of system interoperability" at $P < 0.05$ which is a weak yet positive relation that justified the acceptance of HA5 alternate hypothesis and rejection of null hypothesis HN5. .01 is the PLS coefficient $P < 0.05$ between "system trends to risk" and "lack of information version control" which shown a nil or little relation that justified the acceptance of HN6 null hypothesis and rejection of alternate hypothesis.

As such it was seen and reported that the null hypothesis for H6 was accepted whereas alternate hypothesis rejected where hypotheses H1, H2, H3, H4, and H5 were found to be relevant though with weak positive relation.

Table 3: Hypotheses testing using Partial Least Square Regression (PLS)

H	Variable 1	Variable 2	Coefficient	p-value
H1	Lack of	System trends to	0.50	0.000

	Quality	risk		
H2	Insufficient Testing	System trends to risk	0.62	0.000
H3	Unavailability of source code	System trends to risk	0.50	0.000
H4	Lack of Component Certification	System trends to risk	0.52	0.000
H5	Lack of System Interoperability	System trends to risk	0.41	0.002
H6	Lack of Information for Version Control	System trends to risk	0.01	0.915

Table 4: Comparison of results from different tests

H	**Pearson Correlation**		**Spearman Correlation**		**Partial Least Square**	
H	**Coefficient**	**p-value**	**Coefficient**	**p-value**	**Coefficient**	**p-value**
H1	0.50	0.000	0.50	0.000	0.50	0.000
H2	0.62	0.000	0.62	0.000	0.62	0.000
H3	0.50	0.000	0.54	0.000	0.50	0.000
H4	0.52	0.000	0.48	0.000	0.52	0.000
H5	0.41	0.002	0.40	0.001	0.41	0.002
H6	0.01	0.915	0.02	0.445	0.01	0.915

IV. Result and Discussion

From the survey results done in the industry, the below observations are made.

- *Lack of Quality*: The developer of software should develop something which hasn't yet accomplished the desired functions. For a successful software development, the developer has to develop it in such a way that it satisfies the needs of the end users. In the result analysis, the coefficient indicates that there exists a weak positive link between lack of requirement conformance and system getting prone to risk and failure of the system if there are unambiguous requirements. System integrators who are part of this research show that they have made sure that they chose the perfect component via comparison of their requirements with the features of a component. Even before the start of integration process, they make their component requirements and CBS process much clearer. As such we believe that it is essential for mapping entire requirements of a single component before the integration process for avoiding any possible risks.
- *Insufficient Testing:* Testing is considered as an act targeted at assessing an attribute or a program's capability or that of a system for determining whether it meets the desired results [11]. It can be defined as a process of program execution or that of a system which the aim of error finding [11] which can further aid in lowering the risk possibility. In the result analysis the coefficient indicates that there exists a weak yet positive relation among lack of sufficient testing and system gets risk prone and the alternate hypothesis was accepted by us. This indicates the possibility of CBS risk due to insufficient testing. It is clear from the research that testing enables determination of quality as well as reliability of the component by giving sample data. We believe that detailed testing should be done for ensuring that the process of integration as well as CBS is out of risks.

- *Unavailability of source code:* If there is no access to source code, then it may be difficult to trace integration and testing problems to COTS products. Moreover, Glue Code or the Wrappers is necessary when one component wants to make use of another component, but there exists an incompatibility among the needed interfaces and providers of these components. The major concern of wiring wrappers is for solving mismatches among integrated components so that they can effectively communicate. Highly defined interfaces and with adequate component documentation, the building systems can speed up by the built components being integrated. If there are many providers of the components, then the interfaces for the specific components are not clearly specified. In the result analysis, the coefficient indicates a weak yet positive relation between glue code and system gets risk prone and thus alternate hypothesis has been accepted by us. This shows a chance for risk during wrapper writings among the components during the integration process. The respondents of the survey faced this issues which at times was challenging for writing or modifying wrappers and thus could result in failure of component integration. They indicated that it is not easy for maintaining or debugging wrappers among various components. We believe that the wrapper use during the process of integration has to be minimal so as to avoid any CBS risks.

- *Lack of Component Certification:* Coefficients from the results indicates a weak yet positive relation between uncertified components and systems gets risk prone and thus we accepted the alternate hypothesis. This indicates that the components got to be certified so as to make the integration process and well as CBS free of risk. The respondents of the survey mentioned that system integrator got to ensure that the chosen component is certified even before the beginning of integration process. They believed that there has to be a way for checking and verifying the component certification which can go further through component testing process. As such, we believe the system integrator has to ensure that the component is certified well in advance if the integration process.
- *Lack of System Interoperability:* It is crucial to follow some set standards during development of component interfaces to use in communication. From the

results of correlation studies it has been shown that there exists a weak yet positive relation between lack of interoperability standards and system gets risk prone and thus alternate hypothesis was accepted by us. This indicates that if the components and its interface is developed without any standards then the chances for risk in CBS is high as it will be hard for a system integrator for understanding the code which has nil standard and conventions as well as it will be challenging for the system integrator while modifying the components and component interfaces if needed or during wrapper writings or glue code among the components. Respondent system integrators of this study shows that they follow global standards such as ISO and COSO as well as they have IT compliance for auditing department for this purpose. They also shows that it is not easy to understand the code written by another individual with their own interface standards as well as declaration and it also made it tough on detecting defects or to modify the component interface in CBS. As such we believe that it is crucial to have in place well established development standards of component interface for communicating with other components, which in return can reduce the risks during the process of integration.

- *Lack of Information for Version Control:* The research shows that there exists nil or little relation between lack of version control information and system gets risk prone and as such null hypothesis has been accepted by us. This indicates that maintenance of version information of a specific component has nil high effect on risk during the phase of integration. The respondents of the survey believed that components should have previous versions' as well as new ones' features and this information has to be stored well in the repository of each component. Respondents believed that for version upgrades, appropriate versioning has to be done followed by maintenance of appropriate builds. Upon ensuring all these functions, the component's right version has to be given to system integrator for the process of component integration. As such we believe that maintenance of a component's version control information is crucial but it is not considered as a risk factor of the highest level.

Acknowledgement

The first and second authors gratefully acknowledge Department of Software Engineering, Faculty of Computer Science and Information Technology in support conducting this research.

References

[1] F. Ahmed, L. F. Capretz, "The software product line architecture: An empirical investigation of key process activities Information and Software Technology" 2008. **50**: p. 1098–1113.

[2] W. Mendenhall, T. Sincich, Statistics for Engineering and the Sciences. 5th ed. 2006: Prentice Hall. 1060.

[3] J. Li, et al., "A State-of-the-Practice Survey of Risk Management in Development with Off the-Shelf Software Components" IEE Transactions on Software Engineering, 2008. **34**(2): p. 271 - 286.

[4] J. M. Beaver, G.A. Schiavone, and J.S. Berrios. "Predicting Software Suitability Using a Bayesian Belief Network" in Proceedings of the Fourth International Conference on Machine Learning and Applications. 2005.

[5] U. Assman, "A component model for invasive composition". in ECOOP 2000 Workshop on Aspects and Dimensions of Concerns. 2000. Cannes, France.

[6] S. Mahmood, R. Lai, and Y.S. Kim, "Survey of component-based software development IET Software 2007". **1 Issue:2** (April 2007): p. 57 - 66.

[7] I. Crnkovic, and M. Larsson, "Challenges of component-based development". The Journal of Systems and Software, 2002. **61**(3): p. 201 - 212.

[8] I. Crnkovic, "Component-based Software Engineering - New Challenges in Software Development, in Information Technology Interfaces", 2003. ITI 2003. Proceedings of the 25th International Conference on 2003. p. 9 - 18.

[9] A. Rashid and G. Kotonya, "Risk Management in Component-based Development: A Separation of Concerns Perspective", in ECOOP Workshop on Advanced Separation of Concerns (ECOOP Workshop Reader). 2001.

[10] Y. Ding and N. Napier. "Measurement Framework for Assessing Risks in Component based Software Development". in Proceedings of the 39th Hawaii International Conference on System Sciences. 2006.

[11] J. Pan, Software Testing in Dependable Embedded Systems. 1999.