Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 10, October 2021: 4540-4546

Open Issues and Challenges for Software Component Integrations in Component-Based Software Development

[1] Shambhu Kumar Jha, [2] Prof. Ganesh Gopal Varshney, [3] Dr. R K Mishra
[1] Research Scholar, Mewar University, Chittorgarh, Rajasthan, skjha2@amity.edu
[2] Department of Computer Science, Mewar University, Chittorgarh, Rajasthan, drggvarshney@gmail.com
[3.] Dy. D G, National Informatic Centre, New Delhi, r.k.mishra@nic.in

Abstract

The integration phase of software product is one of the most significant part of software development process. Numerous problems originating from previous phases become visible at this stage and it results in delays and rephrases in the development process. Component-Based Software Development is mainly focusing on building of large and complex software systems through the integration of pre-existing software components. The focus of CBSD is promote software reuse. Component integration is the most critical challenges faced by component integrator during this reuse paradigm, and thus effective and efficient component integration methods and tools are in high demand. This paper highlights various possible problems including security, reliability, and increased development cost with the integration of pre-existing components. Through a systematic literature review and analysis of existing work by different stakeholders of component-based software, this paper is also trying to highlight a solution to this component integration problem towards the success of CBSD.

Keywords: CBSD, COTS, Component integration, component repository, Component Adaption

I. Introduction

The integration phase in traditional software development is always taking place at the tail end of implementation, whereas in component-based development integration of a software component is the centerpiece of the approach. Components must be incorporated through some well-defined architecture (Ayub, Mustafa & Maqsood ,2015). Lots of regulation for component-based software development (CBSD) has been proposed which affects the component reusability, development cost and quality of the end product (Jha and Mishra , 2019). In CBSD the identification of reusable software components and its systematic integration are burning issues among the development community. The Majority of software component being black box in nature communicates with another component through its interface only (Amin, Mahmood, and Oxley 2011). Therefore, component integrators have a huge burden of caring for security, reliability, and increased cost with the development of component-based software. Software Component identified for reuse from component repository has a trustworthiness issue. As the source code of the component is unavailable and the component is also lacking in proper documentation (Jha and Mishra, 2019). Lack of component documentation is affecting the software reuse policy as component integrators are not confident about security and other related issues. Component security is mostly overlooked and ignored during the initial stage of software development as most of the stakeholders of software components are mainly focusing on its functionality (Jha and Mishra ,2018). Delay in the selection process of a software component from the repository also has an impact on CBS quality (Jha and Mishra, 2019. Delay in the selection process takes place because there is no proper classification of components in the repository. Components are only retrieved from the repository based on keyword search methods.

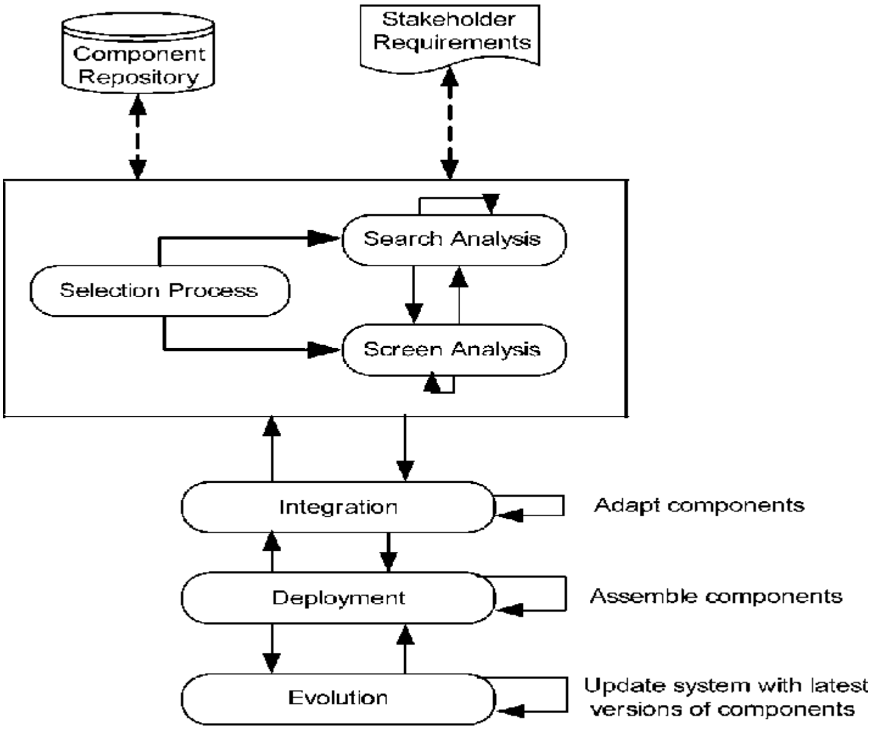

Figure 1: **Component-Based Software Development Process**

The component integrator is also responsible for the replacement of the old version of the component with its latest version (Ayub, Mustafa, & Maqsood ,2015). Finding adequate replacement of old components and integrating into existing component architecture is not an easy task (Dogra, Sharma, and Singh 2016). This has a serious impact on security, reliability, and economic issues for component-based software development.

This paper is structured as follows:

“In the introduction session of the paper we have discussed about the various critical issues related to integration process of component-based software development. Second part of this research paper has discussed about component integration methods. In the third part of the research paper, we have analyzed about Issues and challenges in Component Integration process. Further experimental results and related discussion is mentioned. Finally, it is completed with concluding remarks

2. Component Integration Methods

The method for integration of software components takes place in four major steps as mentioned below in figure 2.

a) **Component qualification** is a process of identifying functionally most suitable component for use from the component repository. It is one of the most challenging tasks for the component integrator as most of the component available in the repository is a completely black box in nature.

b) **Component Adaption** is the level to which a component's internal organization is available for necessary changes as per requirement for the development of new component-based software. In the case of a white box component where source code is available for significant change so that it can operate with the new component whereas for gray box component source code is not modifiable but provides its own extension language or API.

c) **Assembly of component into the system**: Components must be combined through some defined architecture. This architecture provides the binding that forms a system from the different components developed by the different developers using different hardware-software environment (Zhong, and Huang, 2016)

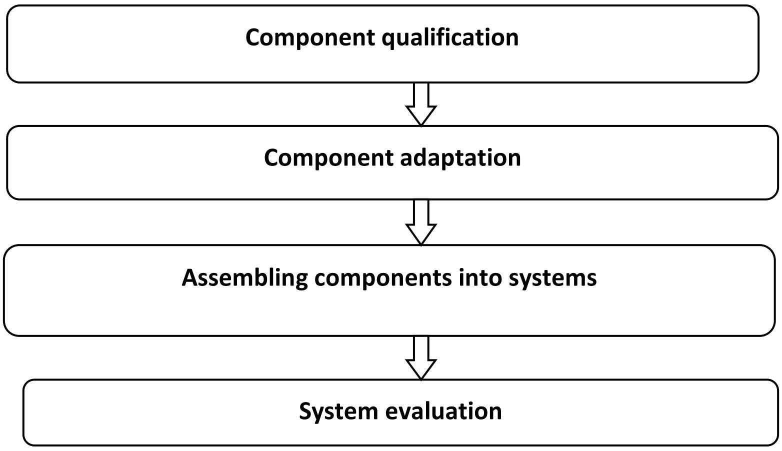

Figure 2: Steps for Component Integration in CBSD

d) **System Evaluation**: After successful completion of component-based software development it seems comparatively unproblematic at the initial stage to evolve and improve as per changing customer need. The substitution of one software component with its latest version is often a lengthy and tough job since the new the will not be able to successfully replace its pre-existing component. The newly developed components need to be meticulously tested in isolation as well as a combination with the rest of the system (Amin, Mahmood, and Oxley 2011).

3. Analyzing the Challenges in Component Integration Process.

Component-based software development process operates on three different levels:

i) Selecting most suitable software component from component repository.

ii) Fitting those selected components together as per defined component architecture.

iii) Integrating them together as per specific user need accordingly.

CBSD focuses more on component integration as it becomes a significant factor for the successful development of component-based software. It is already discussed in the introduction section of this paper that the integration of software components is the centerpiece of the component-based development approach. However, component integrators are facing huge challenges during component integration process (Zhong, and Huang, 2016). Based on detailed survey from various literature and communication with different stakeholders of component-based development majorly identified issues and challenges during the component integration process are summarized in Table 1. In this paper, efforts were made to contact practitioners, researchers, vendors and component developers in order to find out about impacts of mentioned issues and challenges during the component integration process in terms of security, reliability, and increased software cost. More than 75 such stakeholders were contacted through Email, WhatsApp, Linkedin and other social media platform from different backgrounds in terms of job profile, years of work experience in industry and domain specialization. They were asked to give their rating about the impact of issues and challenges on security, reliability, and increased software cost from 0 to 10.

Table 1: Issues and challenges in Component Integration for CBSD

S. No.	**Factors** (Jha and Mishra, 2019)
A	Interface Mismatch of the selected component with another existing component
B	Architectural Mismatch for functionally suitable component
C	Lack of detail functional understanding of software component by the component integrator
D	Lack of documentation for Software component by the component developer
E	Delay in the retrieval of a software component from the component repository
F	Unavailability of the functionally suitable component due to different project constraints.

The respondents were not willing to reveal their identity while giving their rating about the impact of issues and challenges of component integration process on security, reliability, and increased software cost. Out of 90 target respondents 70 have properly responded.

4. Experimental Result

Input received from all the above respondents were thoroughly analyzed and presented in form of line graph and pi graph to represent the impact of various issues and challenges of component integration process on security, reliability, and software cost.

open issues and challenges for software component integrations in component-based software development

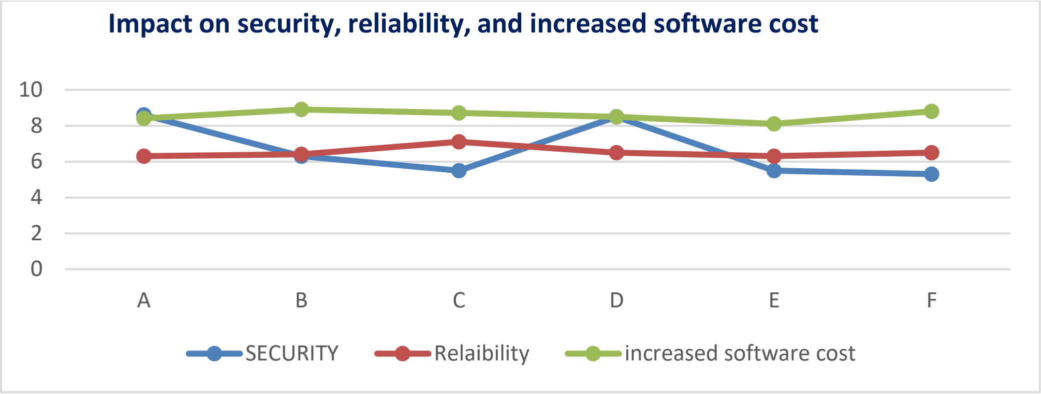

Fig 3: Impact of component integration process on security, reliability, and software cost

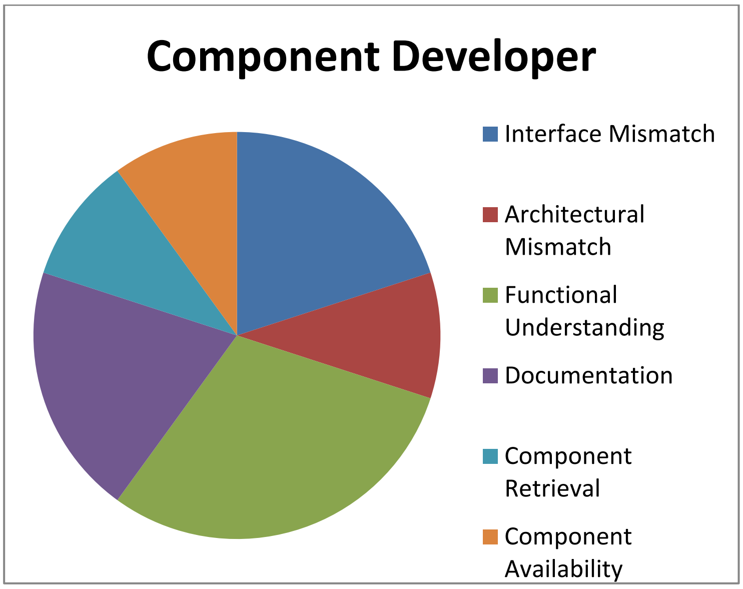

Fig 3.1: Component Developer's view as on component integration process

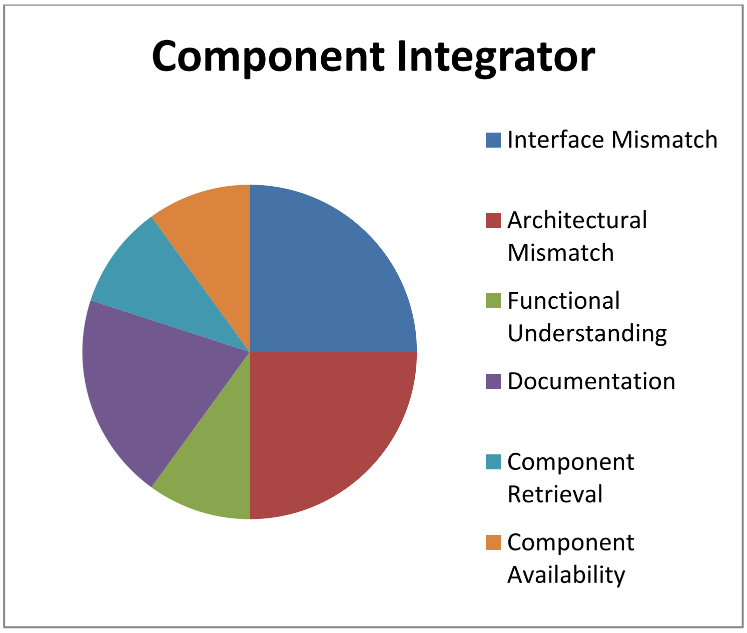

Fig 3.2: Component Integrator's view as on component integration process

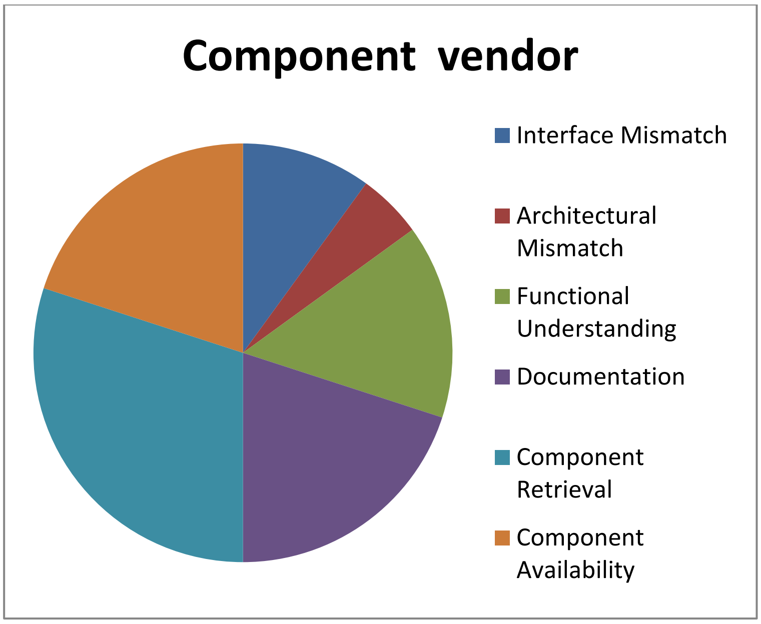

Fig 3.3: Component Vendor's view as on component integration process

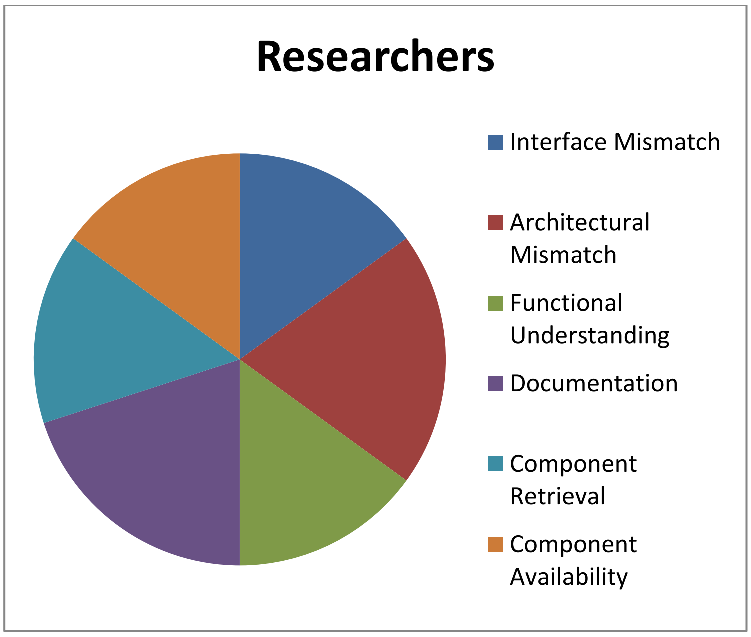

Fig 3.4: Component Integrator's view as on component integration process

5. **Discussion**

From the above graphical representation, it is observed that various issues and challenges of component integration process has maximum impact on software development cost. All the factors mentioned in Table 1 are highly influencing development cost of component-based software. Architectural Mismatch and unavailability of functionally suitable component are causing delay in implementation and increasing the development time. Therefore, they have higher impact on the development cost as compared to other remaining factors. Security of component-based software is highly influenced by interface Mismatch and Lack of documentation for the selected component and least influenced by unavailability of the functionally suitable component. By Using suitable glue code can be helpful for the component user to overcome interface mismatch (Basha and Moiz , 2012). In case functionally suitable component is not available in component repository then the component integrator must look for certified vendors for new component. Reliability factors of component-based software are equally influenced by all the factors mentioned in table 1. It is further suggested to the component integrator to look at the security issues and fix the security preferences as early as possible during the component integration process. This can be helpful to restrict the increased development cost of the software.

6. Conclusion

In this paper main emphasis is given on the overall impact of component integration process on reliability, development cost and security of component-based software. Security aspect in CBSD is mostly given less preference as compared to development cost and reliability. The development cost of the component-based software is highly influenced by all the factors mentioned in Table 1. It is further suggested to the user of software component to look forward for development of new software component if currently required component is unavailable in the component repository. Reliability of software component is confirmed after same component is being used in multiple application under different operating environment.

References

1. Ayub, I., T. Mustafa, and A.Maqsood (2015). “Integration of Traceability and Change Management to Support Migration Issue in CBD”. Science International journal. 27(5): 4379-4383.
2. Amin F., Mahmood A.K., Oxley A. (2011), “A Mixed method study to identify factors affecting software reusability in reuse intensive development”, IEEE.
3. Basha N.J., Moiz S.A. (2021), “Component-Based Software Development: A state of Art”, IEEE-Intl Conf. on Advances in Engineering, Science, and Management (ICAESM)
4. Dogra N. K., Sharma A. K., and Singh H. (2016) “Component Integration: A challenge for Component-Based Software Development” International Journal of Latest Trend in Engineering and Technology (IJLTET), ISSN: 2278-621X, Page 37-40.
5. Gao J, Shih M-C. (2005) “A component testability model for verification and measurement”. Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC), Edinburgh, U.K., vol. 2. IEEE Computer Society Press: Los Alamitos, CA, 2005; 211–218.
6. Jha S. K. and Mishra R.K. (2019), “A Review on Reusability of Component-Based Software Development” Reliability: Theory and Applications (RTA) ISSN: 1932-2321, Volume-14, Issue-4, Page No: 32-36,
7. Jha S.K. and Mishra R.K. (2018) “Predicting and Accessing Security Features into Component-Based Software Development: A Critical Survey”. Software Engineering. Advances in Intelligent Systems and Computing, Springer, Singapore. ISBN: 978-981-10-8848-3. Page: 287-294,
8. Memon AM (2005), “A process and role-based taxonomy of techniques to make testable COTS components. Testing Commercial-off-the-Shelf Components and System”, Springer: Berlin, Page 109–140.
9. Zhong, I., J. Xia, and X. Huang. (2016) “The Framework and its Implementation for Managing Component-Based Software Evolution”. 3rd international conference on information science and control engineering. (ICISCE). 711-715.