

## Issues on the Application of Component-Oriented Software Development: Formulation of Research Areas

<sup>1</sup>H. Aris and <sup>2</sup>S.S. Salim

<sup>1</sup>Department of Software Engineering, College of Information Technology,  
Universiti Tenaga Nasional, Km. 7, Jalan Kajang-Puchong, 43009 Kajang, Selangor, Malaysia  
<sup>2</sup>Department of Software Engineering, Faculty of Computer Science and Information Technology,  
University of Malaya, 50603, Kuala Lumpur, Malaysia

---

**Abstract:** This research presents the outcomes of a study performed with the aim of identifying the issues affecting the application of Component-Oriented Software Development (COSD) amongst the software developers, which consist of those who use components in their software development and those who do not. The information required in the form of problems faced in using components, factors affecting the use of components and reasons for not using components in software development was obtained through questionnaires distribution amongst the software developers in Malaysia. The information gathered was used to determine the issues that affect the application of COSD. These issues are presented in a cause-effect diagram and categorised into technical, managerial and ethical. It serves as the basis for the derivation of a list of possible research areas that can be explored by the researchers in this field. Each research area is then linked to the issues that it solves. This is to ensure that the research area of choice will optimally contribute to the development of COSD in order to promote COSD application amongst the software developers.

**Key words:** Survey, COSD application, state of practice, research area

---

### INTRODUCTION

The idea of Component-Oriented Software Development (COSD) is as old as the software engineering itself. Its inception dates back in the late sixties. From then onwards, it has become one of the central research topics in the field of software engineering and its evolution has taken place in many different forms and from various aspects. The main motivation behind the research is believed to be the list of advantages envisaged by the successful implementation of COSD as mentioned by Vitharana (2003). Despite the presence of a number of challenges along the path in adopting COSD (Crnkovic and Larsson, 2000), researchers are still optimistic about the future of COSD. Hence, research work on COSD is becoming more and more intense with each addressing one or more challenges imposed on COSD. This phenomenon has left interested researchers with wide varieties of COSD research areas to choose from.

In order to ensure that the research area of choice will optimally contribute to the development of COSD, it is important to first understand the current situation before the selection is made. This is also important to ensure that

the research pursued is relevant to the current needs of the parties involved in COSD, who may range from component developers to end users.

A number of work are also found that study the state of component usage in the software development industry at a number of locations. However, it is difficult to make any useful conclusion that relates their studies to ours, as each is focusing on the different aspects and focus of component usage. For instance, the existing studies were performed on the component users only whereas in present study, we are also taking into consideration responses from the non-component users in deriving the issues affecting the application of COSD. This makes the formulated research areas applicable to wider group of targets. These other studies are briefly described below.

The first one studied the situation at Kingston city of Jamaica (Pyne *et al.*, 2005). This study, which was done based on the feedback from eight prominent software development companies in Kingston, covered:

- The level of components reuse in software development

- The quality of software systems created with components reuse
- The average number of software systems created per year
- The cost associated with components reuse

In particular, it concluded that all of the companies involved in the case study have experienced the benefits of COSD (i.e., improved quality, higher productivity and reduced development cost) and that the main success factor of COSD was attributed to the existence of good components repositories.

Another study found performed an empirical study on Off-The-Shelf (OTS) component usage in industrial projects (Chen *et al.*, 2007) covering software development companies in three European countries; Norway, Italy and Germany. OTS components in this study were categorised into Commercial Off-The-Shelf (COTS) components and Open Source Software (OSS) components. COTS components are owned by commercial vendors and their users usually do not have access to the source code of these components, whereas OSS components are provided by open source communities that offer full control of the source code. The study investigated a total of 71 projects that exclusively used COTS components and 39 projects that exclusively used OSS components. Three dimensions under investigation are:

- The users of OTS components
- The reasons for deciding to use the OTS components
- The outcomes of using the components

The study concluded that the users of COTS and OSS components are companies with similar profiles where software houses and information technology consulting companies were the main users of these components. Meanwhile, the top three reasons for using the OTS components, both the COTS and OSS, are time-to-market, reliability and performance, with the key motivations being to save development time and effort and to get newest technology. The results of using OTS components were formulated in term of the risks experienced during projects development. Fifteen possible risks were investigated and the result showed that some risks were common to both OTS and OSS component users and some others were different.

Another study that investigated software development using OSS components in the Chinese Software Industry was also found (Li *et al.*, 2005). This survey particularly focused on three issues in reusing OSS components. These issues are component selection, licensing terms and system maintenance. On component

selection, they concluded that the selection of OSS components were made initially using existing web search engines, followed by local expertise for evaluation of the selected components. On licensing terms, their survey discovered that the OSS licensing terms were not seen as a barrier to the software companies in China when they reuse the components in their software development projects. Finally, with regard to system maintenance, they concluded that 84% of the maintenance work is dedicated to bug fixing or other code changes in the selected OSS components. Even though the developers admitted the need for active participation in the OSS community for this purpose, close participation with OSS community was still rare.

This study describes about similar study, which was performed with the aim of identifying the issues affecting the application of COSD amongst the software developers in Malaysia. The following objectives have been set for the study:

- To identify issues affecting the application of COSD in Malaysia
- To formulate research areas that addresses the identified issues

The first objective was accomplished through questionnaires distribution involving 104 software developers who are working at various companies in Malaysia. Based on the responses received, respondents of the survey can be divided into two categories; component users and non-component users. The component users are those who admit that they use components in their software development projects. Conversely, the non-component users are those who do not use components in their software development projects. From the component users, the survey intended to identify:

- The problems that they face in using components in their software development projects
- The factors that they believe are important towards successful application of COSD in their software development projects

From the non-component users, the survey intended to find out the 'reasons' why they do not use components or do not apply COSD in their current software development projects.

The problems, factors and reasons identified were then analysed to derive a list of issues that influence the application of COSD in Malaysia. The list of issues was then used to formulate a number of COSD research areas, with each area addressing one or more issues.

**Identifying issues:** The first objective of this study, as mentioned earlier, is to identify the issues that affect the application of COSD amongst the software developers in Malaysia. For this purpose, we have extracted from the questionnaire returns, three information; problems faced in using components in software development, factors affecting the use of components and reasons for not using components, which are elaborated in the next three subsections.

**Problems:** One of the questions in the questionnaire, which was targeting at the component users, asked about the problems that they faced in using components in their software development projects. To facilitate them in answering the question, possible problems were listed, taking into consideration the classification of issues and risks raised by Brereton and Budgen (2000) and Vitharana (2003), respectively. Component users can indicate the problems that they are facing by marking the checkbox next to each problem listed. Component users are also allowed to write additional problems that were not listed in the space provided. The list of problems faced, as indicated by the responses received includes:

- Availability of tools to support COSD (P1)
- Practical hands-on training on the application of COSD (P2)
- Sufficient knowledge on component technology (P3)
- Strong support from the management to shift to COSD (P4)
- Strong financial support or funding to start up COSD (P5)
- Willingness to learn and adopt new skills and technology (P6)
- Proper procedures on components ownership, transfer et cetera in place (P7)
- Willingness to share components with other developers (P8)

Figure 1 shows the feedback received in percentage form for each problem.

**Factors:** Another question on the questionnaire is asking the component users on the factors that they believe play important roles towards successful implementation of COSD. These factors are:

- Lack of tools to support the development process (F1)
- Insufficient components to be used and reused (F2)
- No proper procedures for developing, depositing and retrieving components from the repository (F3)

- Focus not given to COSD (F4)
- Insufficient funding in shifting to COSD (F5)
- Trust in using components developed by other people (F6)
- Prefer to develop the whole application afresh (F7)
- Ownership issues of the shared components (F8)

The result also varies as shown in Fig. 2.

**Reasons:** From the non-component users, which accounts for 42.57% of the total respondents, the questionnaire intends to identify the reasons why they do not use components in their current software development projects. Similar to discovering the problems and factors from the component users, a list of possible reasons is given in the questionnaire for them to choose from. On top of that, they can also specify reasons which are not listed. Various reasons are given for not using components. These are:

- Lack of knowledge about components (R1)
- Not exposed to the component technology (R2)
- Lack of available supporting tools (R3)
- Lack of support from the management (R4)
- Insufficient funding to shift to COSD (R5)
- Do not feel the need to change the current practice (R6)
- Not willing to change the current development practice (R7)
- Not willing to learn new technology (R8)
- COSD is not the common development practice in Malaysia (R9)

The percentage of each reason received from the non-component users is shown in Fig. 3.

## FORMULATING RESEARCH AREAS

From the list of problems faced by the component users, factors affecting the application of COSD by component users and reasons for not using component by the non-component users presented earlier, it can be seen that some of the problems, factors and reasons coincide with one another. Therefore, to avoid redundancy, the overlapping problems, factors and reasons are grouped into issues that affect the application of COSD in Malaysia as shown in Table 1.

These issues are then further categorised into technical, managerial and ethical as shown in the cause-effect diagram of Fig. 4. The categorisation is made based on the source of the problem and hence, the party that is responsible to solve the problem (Brereton and Budgen, 2000). For example, an issue

Table 1: Grouping of problems, factors and reasons into respective issues

Problems	Factors	Reasons	Issues
Availability of tools to support COSD (P1)	Lack of tools to support the development process (F1)	Lack of available supporting tools (R3)	Tools availability
Practical hands-on training on the application of COSD (P2)	-	-	Training
Sufficient knowledge on component technology (P3)	-	Lack of knowledge about components (R1)	Knowledge on COSD
Strong support from the management to shift to COSD (P4)	-	Lack of support from the management (R4)	Management support
Strong financial support or funding to start up COSD (P5)	Insufficient funding in shifting to COSD (F5)	Insufficient funding to shift to COSD (R5)	Financial
Willingness to learn and adopt new skills and technology (P6)	-	Not willing to learn new technology (R8)	Willingness to learn
Proper procedures on components ownership, transfer et cetera in place (P7)	No proper procedures for developing, depositing and retrieving components from the repository (F3)	-	Proper procedure
Willingness to share components with other developers (P8)	-	-	Willingness to share
-	Insufficient components to be used and reused (F2)	-	Insufficient components
-	Focus not given to COSD (F4)	-	No focus
-	Trust in using components developed by other people (F6)	-	Prefer own components
-	Prefer to develop the whole application afresh (F7)	-	Trust issues
-	Ownership issues of the shared components (F8)	-	Ownership issues
-	-	Not exposed to the component technology (R2)	Exposure to COSD
-	-	Do not feel the need to change the current practice (R6)	Need for change
-	-	Not willing to change the current development practice (R7)	Willingness to change
-	-	COSD is not the common development practice in Malaysia (R9)	Uncommon practice

belongs to technical category if software engineers, software developer and the like are expected to produce the solution for it. It belongs to managerial category if management support and adequate organisational structure are required to solve it. An issue belongs to ethical category if it requires change of mindset or belief in order to solve it and this is, probably, the most difficult to tackle. Assignment of a particular issue into one of the categories may sometimes be ambiguous though as one issue may depend on another or more.

From the results presented in Fig. 1-3, it can be seen that the priority order of the issues that are considered as problems, factors and reasons vary between the component users and non-component users. To the component users for example, the top three issues that are considered as problems in applying COSD are as shown in Table 2.

As for the factors influencing the successful application of COSD, they prioritise on the issues shown in Table 3.

For the non-component users on the other hand, the first three issues given by them as reasons for not using components in software development are shown in Table 4.

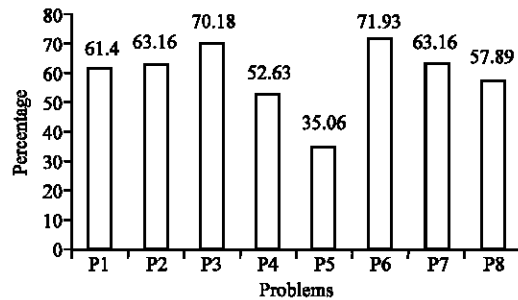


Fig. 1: Problems faced by the component users

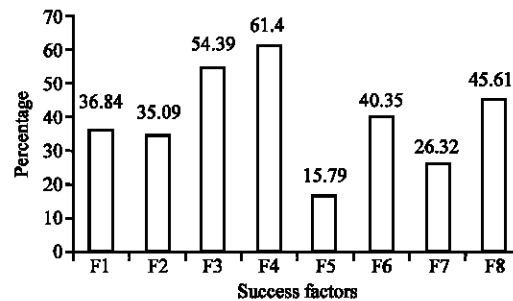


Fig. 2: Success factors in sustaining the application of COSD in the future

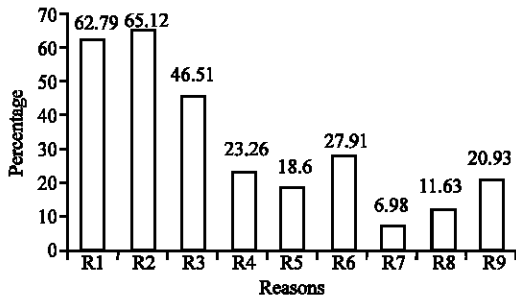


Fig. 3: Reasons for not using components

Table 2: Problems faced by the component users

Issue	Category
No focus	Managerial
Proper procedures	Technical
Ownership issues	Ethical

Table 3: Factors influencing successful application of COSD

Issue	Category
Willingness to learn	Ethical
Knowledge on COSD	Technical
Training needs	Technical

Table 4: Reasons for not using components

Issue	Category
Exposure to COSD	Technical
Knowledge on COSD	Technical
Tools availability	Technical

Due to this differing priority orders, it is therefore not feasible to address a single problem, factor or reason that will give the most effect on the application of COSD to both component users and non-component users alike. It is also not feasible to focus on any one of the three categories of issues due to this differing priority. Therefore, it is suggested that each issue should be equally treated. For this purpose, four possible solutions, labelled  $S_1$  to  $S_4$ , are proposed to address the problems and factors as shown in Fig. 5.

The first solution,  $S_1$ , is on the provision of trainings and workshops related to the application of COSD in software development. Issues that are directly influenced by this solution can be divided into two as shown separated by dashed line in Fig. 5. Issues above the dashed line are those that directly affect the software developers. Meanwhile, issues below the dashed line are those that affect the management of a software company. For the software developers, two categories of trainings and workshops can be further identified.

Soft-skill training that focuses on information dissemination to instil awareness on the benefits of using components and Hard-skill training that focuses on the use of tools, techniques and model to facilitate the development of software using COSD approach.

The second solution,  $S_2$ , is on the definition of procedures and regulations pertaining to the use of components. This includes the procedures to store components into the repositories that may include the issues of labelling and tagging a component, procedures to retrieve components from the repositories that may include the issues of acknowledgment, royalty, copyrights and the extent of which a component can be modified. Regulations that govern the use of third party components, including accountability issues, should also be defined. Proper procedures and regulations in place help to resolve issues on the lack of proper procedure, willingness to share, trust and ownership issues as shown in Fig. 5.

The third solution,  $S_3$ , on the development of tools to support COSD process is the research area where, most of the current research on COSD is focusing on (Crnkovic and Larsson, 2000). This solution concerns with the production of tools that can be used to facilitate the software developers in developing software using COSD approach. The tools can support individual or a group of processes in COSD. Tools can also include framework or models that can be applied by the software developers to assist the development study.

The above three are considered as the primary solutions where each solution opens up for potential research areas to be explored in order to address the issues under it. Each research areas in turn can be directly initiated to solve a specific issue. In combination, all of the primary solutions described above will be able to address all of the issues shown in Fig. 4 and 5, with the exception of two issues; insufficient components and uncommon practice. These two issues cannot be directly addressed by any one of the solutions described before. However, their intensity can be subsequently reduced with the maturity of COSD. As more and more developers are using COSD in their software development projects, more components will be produced, hence overcoming the issue of insufficient components. Also, with the growing number of software developers who are using COSD approach, COSD will no longer be seen as uncommon practice, hence resolving the uncommon practice issue.

All of the three primary solutions described above contribute to the maturity of COSD,  $S_4$ . Therefore, it is regarded as the secondary solution resulting from the primary solutions to the issues in COSD application in Malaysia, indicated by the arrow that spans the primary solutions towards the secondary solution as shown in Fig. 5.

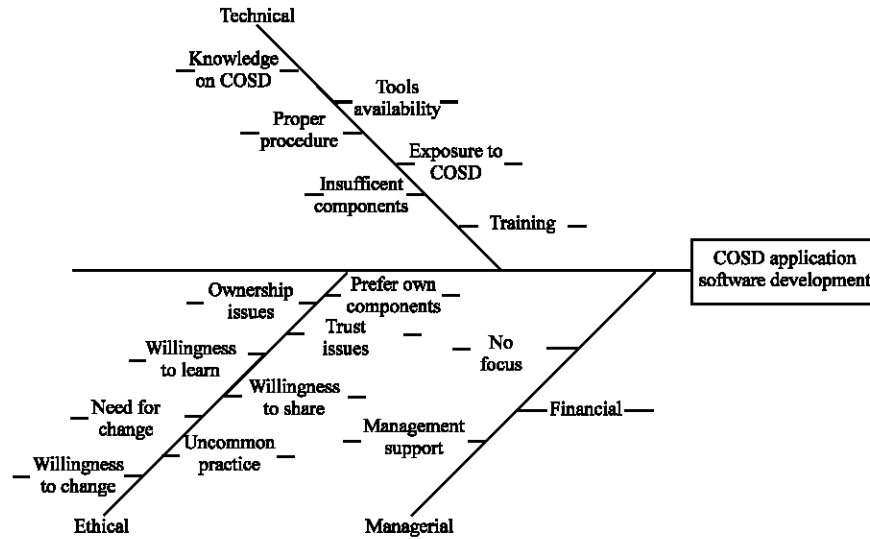


Fig. 4: Cause-effect diagram of the issues affecting the application of COSD

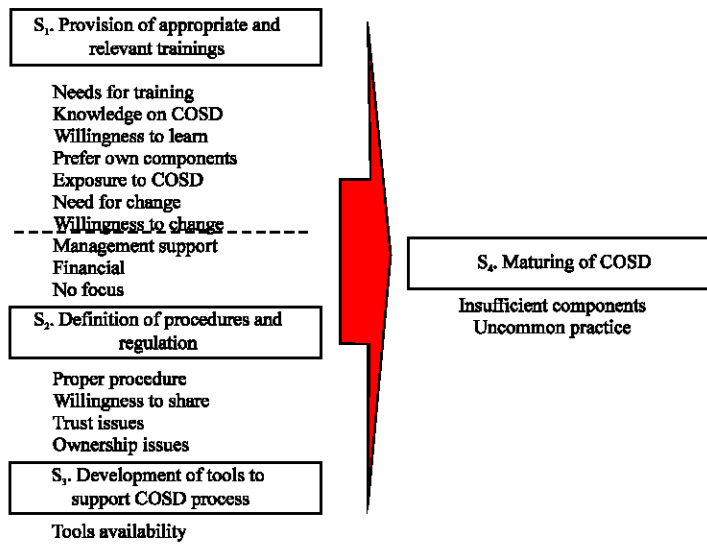


Fig. 5: Model for solutions and issues that each covers

**CONCLUSION**

This study presents the results of a survey performed on the software developers with the aim of identifying the problems faced by the component developers in using components in their software development, the factors affecting the use of components amongst the component users and the reasons for not using components from the non-component users. The problems, factors and reasons are grouped according to their description to derive a list of issues that affect the application of COSD. The issues are then further

classified into technical, managerial and ethical based on the parties required to address the issues. The list of issues is then used to determine a list of possible research areas that can be explored in order to promote COSD application amongst the software developers.

**REFERENCES**

Brereton, P. and D. Budgen, 2000. Component-based systems: A classification of issues. IEEE Comput. Mag., 33: 54-62.

- Chen, W., J. Li, J. Ma, R. Conradi, J. Ji and C. Liu, 2007. A survey of software development with open source components in Chinese Software Industry. *Lecture Notes Comput. Sci.*, 4470: 208-220.
- Crnkovic, I. and M. Larsson, 2000. *Building Reliable Component-Based Software Systems*. 1st Edn., Artech House Publishers, ISBN: 1-58053-327-2.
- Li, J., R. Conradi, O.P.N. Slyngstad, C. Bunse, U. Khan, M. Torchiano and M. Morisio, 2005. An empirical study on off-the-shelf component usage in industrial projects. *Lecture Notes Comput. Sci.*, 3547: 54-68.
- Pyne, R., S. McNamarah, M. Bernard, D. Hines, G. Lawrence and D. Barton, 2005. An evaluation on the state of component-based software engineering in Jamaica. *Proceedings of the IEEE Southeast Conference*, April 2005, IEEE Computer Society, Ft. Lauderdale, Florida, pp: 570-575.
- Vitharana, P., 2003. Risks and challenges of component-based software development. *Commun. ACM.*, 46: 67-72.