

# An Empirical Study on Off-the-Shelf Component Usage in Industrial Projects

Jingyue Li<sup>1</sup>, Reidar Conradi<sup>1,2</sup>, Odd Petter N. Slyngstad<sup>1</sup>, Christian Bunse<sup>3</sup>,

Umair Khan<sup>3</sup>, Marco Torchiano<sup>4</sup>, and Maurizio Morisio<sup>4</sup>

<sup>1</sup>Department of Computer and Information Science,  
Norwegian University of Science and Technology (NTNU),  
NO-7491 Trondheim, Norway  
{jingyue, conradi, oslyngst}@idi.ntnu.no

<sup>2</sup>Simula Research Laboratory, P.O.BOX 134, NO-1325 Lysaker, Norway

<sup>3</sup>Fraunhofer IESE, Sauerwiesen 6,  
D- 67661 Kaiserslautern, Germany  
{Christian.Bunse, khan}@iese.fraunhofer.de

<sup>4</sup>Dip. Automatica e Informatica, Politecnico di Torino  
Corso Duca degli Abruzzi, 24, I-10129 Torino, Italy  
{maurizio.morisio, marco.torchiano}@polito.it

**Abstract.** Using OTS (Off-The-Shelf) components in software projects has become increasingly popular in the IT industry. After project managers opt for OTS components, they can decide to use COTS (Commercial-Off-The-Shelf) components or OSS (Open Source Software) components instead of building these themselves. This paper describes an empirical study on why project decision-makers use COTS components instead of OSS components, or vice versa. The study was performed in form of an international survey on motivation and risks of using OTS components, conducted in Norway, Italy and Germany. We have currently gathered data on 71 projects using only COTS components and 39 projects using only OSS components, and 5 using both COTS and OSS components. Results show that both COTS and OSS components were used in small, medium and large software houses and IT consulting companies. The overall software system also covers several application domains. Both COTS and OSS were expected to contribute to shorter time-to-market, less development effort and the application of newest technology. However, COTS users believe that COTS component should have good quality, technical support, and will follow the market trend. OSS users care more about the free ownership and openness of the source code. Projects using COTS components had more difficulties in estimating selection effort, following customer requirement changes, and controlling the component's negative effect on system security. On the other hand, OSS user had more difficulties in getting the support reputation of OSS component providers.

## 1. Introduction

Due to market requirements concerning cost and time-to-market, software developers are searching for new technologies to improve their projects with respect to these qualities. Software components promise to have a positive impact on software reuse, resulting in time and cost efficient development. Therefore, software developers are using an increasing amount of COTS (Commercial-Off-The-Shelf) and OSS (Open Source Software) components in their projects. Although both COTS and OSS component are claimed to save development effort, they are still very different. COTS components are owned by commercial vendors, and their users normally do not have access to the source code of these components. On the other hand, OSS components are provided by open source communities. Thus, they offer full control on the source code [15].

When planning a new software project, project decision makers need to decide whether they should buy a COTS component, or acquire an OSS component if it was decided to use OTS components. To make such a decision, it is important to investigate previous projects using such components and summarize the relevant decision-making processes and project results.

The study presented in this paper has investigated 71 finished software projects using only COTS components and 39 projects using only OSS components. It compared the COTS components with the OSS components in three dimensions: (1) Who is using OTS components, (2) Why project members decide to use them, and (3) What were the results of using them.

The remainder of this paper is organized as follows: Section two presents some previous studies on benefits and risks of using OTS components. Section three describes the research design applied. Section four presents the collected data, whereby the discussion of results is given in section five. Finally, conclusions and future research are presented in section six.

## 2. Previous studies on OTS component

COTS components promise faster time-to-market and increased productivity of software projects [1]. At the same time, COTS software introduces many risks, such as unknown quality of the COTS components, which can be harmful for the final product, or economic instability of the COTS vendor who may terminate maintenance support [2]. Furthermore, the use of OSS in industrial products is growing rapidly. The basic idea behind open source is very simple: When programmers can read, re-distribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing [15].

OSS has many proposed advantages [3]: *OSS is usually freely available for public download. The collaborative, parallel efforts of globally distributed developers allow much OSS to be developed more quickly than conventional software. Many OSS*

*products are recognized for high reliability, efficiency, and robustness.* Despite its wide appeal, OSS software faces a number of serious challenges and constraints. *The popularity of OSS increases the risk that so-called net-negative-producing programmers will become involved. Many software tasks, such as documentation, testing, and field support are lacking in OSS projects. If developers produce or sell a product, which integrates OSS as source code, they may need the licensor's permission. Otherwise, the licensor might claim for damages or force them to end the product's further development, delivery and sale [4].* Furthermore, many common perceptions about OSS need further empirical clarification. For example, there is still no empirical evidence that OSS fosters faster system growth [10]. There is also no strong evidence that OSS is more modular than closed source software [10].

Some previous studies have investigated the process of using COTS components in software development, such as COTS component selection [5] and COTS-based development processes [6]. Other studies have investigated how to use OSS software in product development [4], such as mission-critical development [11] and information system infrastructure development [12]. Although these studies present recommendations on how to integrate COTS or OSS component into the final product, few studies have investigated a higher level question: Why should I use COTS component instead of OSS components, or vice versa?

### **3. Research design**

This study is the second phase of a systematic study on process improvement and risk management in OTS based development. We started from a pre-study focused on COTS components, which was performed in the form of structured interviews of 16 COTS projects in Norwegian IT companies [9]. This study presented in this paper extended the pre-study in two dimensions. First, it included OSS components because they represent an alternative to COTS components. Second, this study included samples from Norway, Italy and Germany. In addition, the sample was selected randomly instead of on convenience as in the pre-study.

#### **3.1. Research questions**

To investigate the motivation of using OTS components, we first want to know who is using OTS components. Therefore, the first research question is:

**RQ1:** *What are the commonalities and differences in profiles on projects using COTS components vs. those using OSS components?*

After we know who is using OTS components, we want to know why they decided to use OTS components. Thus, the second research question is:

**RQ2:** *What are the commonalities and differences in the motivation of projects using COTS components vs. those using OSS components?*

After we know who and why, we need to know the possible problems of using OTS components. We intended to investigate whether the current motivation and ex-

pectations of using OTS components are proper or not. Therefore, the third research question is:

**RQ3:** *What are the commonalities and differences in possible risks (problems) of projects using COTS components vs. those using OSS components?*

### 3.2. Research method

In this study, we used a questionnaire to collect data, organized in six sections:

1. Background questions to collect information on the company, and respondents.
2. Background information concerning projects, such as development environment, application domain, and non-functional requirements emphasis.
3. The motivation of using OTS components.
4. The specific motivation of using COTS or OSS components.
5. Information relevant to process improvement and risk management in OTS component-based development.
6. Detailed information about one specific OTS component used in the actual project.

### 3.3. Concepts used in this study

Concepts used in this study are listed in the first page of the questionnaire, for example:

- **Component:** Software components are program units of independent production, acquisition, and deployment that can be composed into a functioning system. We limit ourselves to components that have been explicitly decided either to be built from scratch or to be acquired externally as an OTS-component. That is, to components that are not shipped with the operating system, not provided by the development environment, and not included in any pre-existing platform.
- **OTS component** is provided (by a so-called **provider**) from a COTS vendor or the OSS community. The components may come with certain obligations, e.g. payment or licensing terms. An OTS component is not controllable, in terms of provided features and their evolution and is mainly used as closed source, i.e. no source code is usually modified.

### 3.4. Data collection

**Sample Selection.** The unit of this study is a completed software development project. The projects were selected based on two criteria:

- The project should use one or more OTS components
- The project should be a finished project, possibly with maintenance, and possibly with several releases.

We used random selection to gather representative samples in three European countries:

- In Norway, we gathered a company list from the Norwegian Census Bureau (SSB) [7]. We included mostly companies which were registered as IT companies. Based on the number of employees, we selected the 115 largest IT companies (100 biggest IT companies plus 15 IT departments in the largest 3 companies in 5 other sectors), 150 medium-sized software companies (20-99 employees), and 100 small-sized companies (5-19 employees) as the original contacting list.
- In Italy, we first got 43580 software companies from the yellow pages in the phone book. We then randomly selected companies from these. For these randomly selected companies, we read their web-site to ensure they are software companies or not. 196 companies were finally clarified as software companies, and were used as the original contact list.
- In Germany, we selected companies from a list coming from an organization comparing to the Norwegian Census Bureau. We then used the existing IESE customer database to get contact information of relevant IT/Software companies, in line with the Norwegian selection.

**Data collection procedure.** The final questionnaire was first designed and pre-tested in English. It was then translated into the native language of the actual country and published on the SESE web tool at Simula Research Lab [8]. Possible respondents were first contacted by telephone. If they had suitable OTS-based projects and would like to join our study, a username and password was sent to them, so that they could log into the web tool to fill in the questionnaire (they can also use a paper version). The average time to fill in the questionnaire is between 20 to 30 minutes.

### 3.5. Data analysis

According to the focus of the different research questions, we used different data analysis methods:

- For **RQ1**, we analyzed the distribution of the variables.
- For **RQ2**, we want to see both the commonalities and differences of variables. To compare differences, we first used box-plots to show the median and distribution of the variable. We then compared the mean difference of variables. S.S. Stevens classified “permissible” statistical procedures [18] according to four main scales: nominal, ordinal, interval and ratio scale [19]. For the ordinal scales, the permissible statistics included median, percentiles, and ordinal correlations. Mean and standard deviations are allowed for interval and ratio scales. In our study, although the scale of our data is Likert scales, we also compared the mean value of different variable to see if they are significant different. A number of reasons account for this analysis: First, Spector [20] showed that people tend to treat categories in Likert scales as equidistant, regardless of the specific categories employed. Second, using parametric tests for scales that are not strictly interval does not lead, except in extreme cases, to wrong statistical decisions [21]. Third, D.J.Hand concluded that *restrictions on statistical operations arising from scale type are more*

*important in model fitting and hypothesis testing contexts that in model generation or hypothesis generation contexts. In the latter, in principle, at least, anything is legitimate in the initial search for potentially interesting relationship [22]. Although we used a t-test to compare the mean difference, the intention of this study is still model generation and hypothesis generation. We therefore believe that it is permissible to treat Likert-scales items as leading to interval scales measurements that can be analyzed with parametric statistics.*

- For **RQ3**, we used the same analysis method as in RQ2.

## 4. Research results

Although the data collection is still on-going in Germany and Italy, we have gathered results from 115 projects (47 from Norway, 25 from Italy, and 43 from Germany). In these 115 projects, 71 used only COTS components, 39 used only OSS components, and five used both COTS and OSS components. In this study, we discarded the projects using both COTS components and OSS component, because they will confound the results and cover only 4% of the total projects.

Most respondents of the 110 projects, which used either COTS components or OSS components, have a solid IT background. More than 89% of them are IT managers, project managers, or software architects. More than 88% of them have more than 2 year experiences with OTS-based development. All of them have at least a bachelor degree in informatics, computer science, or telematics.

### 4.1 Answers to RQ1 – Who is using OTS components

To answer RQ1, we designed two sub-questions:

- **RQ1.1:** What are the commonalities and differences in profiles of the companies and projects using COTS components vs. using OSS components?
- **RQ1.2:** What are the commonalities and differences in emphasis in projects using COTS component vs. using OSS components?

Results are coming from projects in several domains, as showed in Figure 1. The distribution of company size and the companies' main business area are shown in Figure 2 and Figure 3.

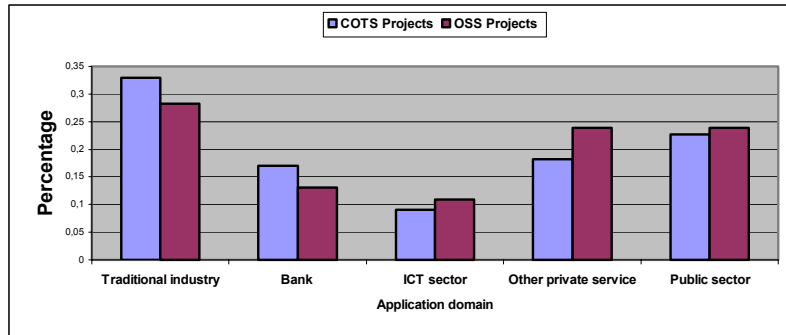


Fig. 1. Distribution of the application domain of the final system

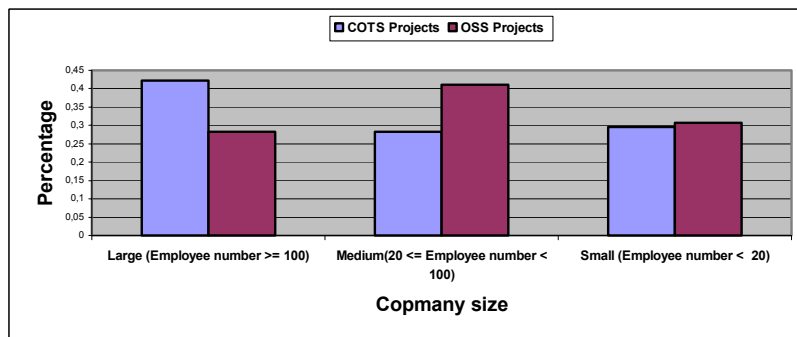


Fig. 2. Distribution of company size

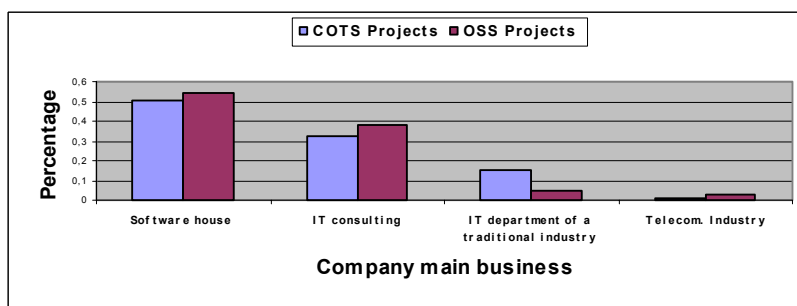


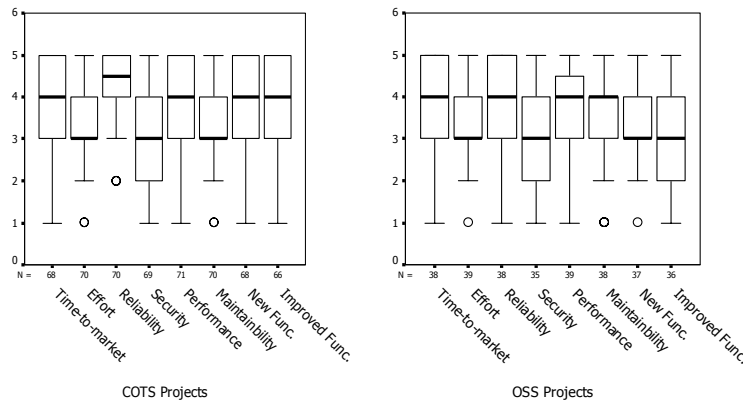
Fig. 3. Distribution of companies' main business areas

For **RQ1.2**, we listed some possible characteristics of the actual system:

- Time-to-market
- Effort (cost)
- Reliability

- Security
- Performance
- Maintainability
- New functionality (first launch in the market)
- Improved functionality (over competitors)

Respondents were asked to answer “don’t agree at all”, “hardly agree”, “agree somewhat”, “agree mostly”, “strongly agree”, or “don’t know”. We assign an ordinal number from 1 to 5 to the above alternatives (5 means strongly agree). The results are shown in Figure 4.



**Fig. 4.** The emphasis on characteristics of the system

To compare differences in the characteristics of the system, we compared the mean values of each characteristic in systems using COTS vs. systems using OSS. The results show that there is no significant difference. Answers to **RQ1** are summarized in Table 1.

**Table 1.** Answers to research question RQ1

Research question	Commonalities (COTS and OSS)	Differences (COTS vs. OSS)
RQ 1.1	Both are being used in systems of several application domains (see Figure 1).	No difference
RQ1.1	Both are being used in small, medium, and large IT companies (see Figure 2)	No difference
RQ 1.1	<b>Software house</b> and <b>IT consulting</b> companies are the main users (see Figure 3)	No difference
RQ 1.2	Both systems emphasis on <b>time-to-market</b> , <b>reliability</b> , and <b>performance</b> (see Figure 4)	No difference



## 4.2 Answers to RQ2 – Why was it decided to use OTS components?

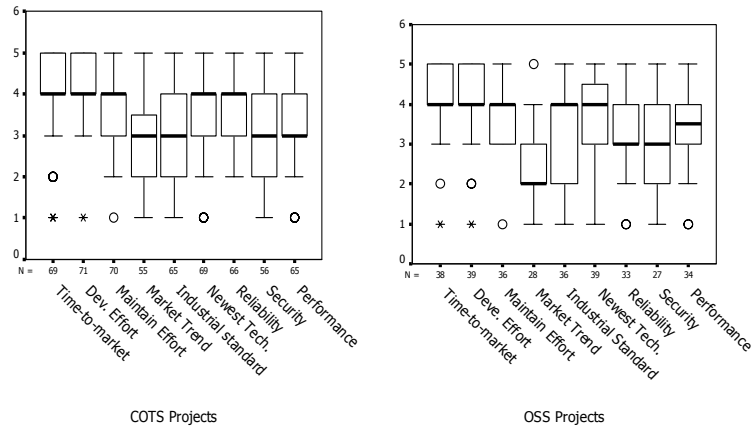
To answer **RQ2**, we designed three sub-questions:

- **RQ2.1:** What were the differences of users' general expectations on COTS components vs. OSS components?
- **RQ2.2:** What were the specific motivations of using COTS components?
- **RQ2.3:** What were the specific motivations of using OSS components?

For **RQ2.1**, we gathered some common expectations of using OTS components from our pre-study [9] and from literature reviews [1][3]:

- Shorter time-to-market
- Less development effort/cost
- Less maintenance effort/cost
- Larger market share
- Compliance with industrial standards
- Keeping up with the newest technology
- Better system reliability
- Better system security
- Better system performance

We used the same format and measurement as **RQ1.2**. The results are shown in Figure 5.



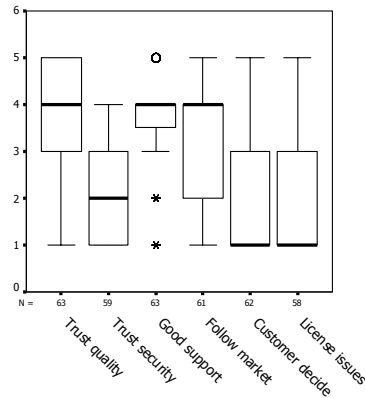
**Fig. 5.** The general expectations of using OTS components

We used SPSS 11.0 to compare the mean values of each motivation of using COTS vs. using OSS. The results show that there is no significant difference.

For **RQ2.2**, we gathered some specific motivations of using COTS components:

- Reduce risk of bad quality, because components were paid for
- Reduce risk of poor security, because source code was closed/unavailable
- Provider could give adequate technical support
- Provider may provide give components following market trends
- It was decided by customer

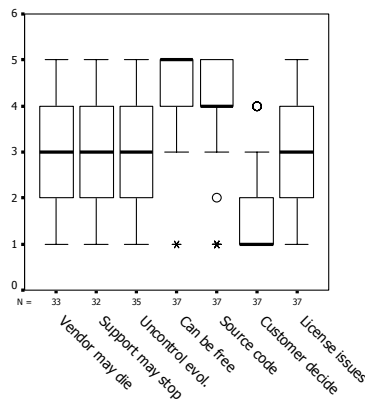
- Political reasons (company policy, licensing conditions)
- The results of **RQ2.2** are shown in Figure 6.



**Fig. 6.** Specific motivation of using COTS components

For **RQ2.3**, we gathered some specific motivations of using OSS components:

- Reduce risk of provider going out of business
  - Reduce risk of provider changing market strategies
  - Reduce risk of selected components evolving into an unwanted direction
  - Component can be acquired for free
  - Source code is available and can easily be changed
  - It was decided by the customer
  - Political reasons (company policy, licensing conditions)
- The results of **RQ2.3** are shown in Figure 7.



**Fig. 7.** Specific motivations of using OSS components

Answers to research question **RQ2** are summarized in Table 2.

**Table 2.** Answers to research question RQ2

Research question	Commonalities (COTS and OSS)	Differences (COTS vs. OSS)
RQ 2.1	Both COTS and OSS users expect that the components can contribute to shorter time-to-market, less development effort, less maintenance effort, and newest technology (see Figure 5)	No difference
RQ2.2	COTS users believe that paid software will give good quality and will follow market trends. They also believe that the COTS vendor will provide good technical support (see Figure 6).	
RQ 2.3	The main motivations of using OSS are that code could be acquired for free, source code is available, and to avoid the possible vendor support risk (see Figure 7).	

#### 4.3 Answers to RQ3 – What were the results of using OTS components

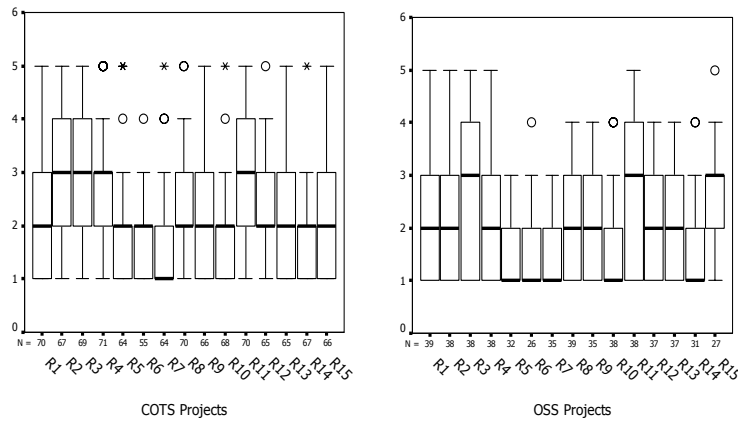
To answer **RQ3**, we formulized 15 possible risks (as showed in Table 3) on OTS-based development.

**Table 3.** Possible risks in OTS based development

Phase	ID	Possible risks
Project plan phase	R1	The project was delivered long after schedule [13].
	R2	Effort to select OTS components was not satisfactorily estimated [14].
	R3	Effort to integrate OTS components was not satisfactorily estimated [13].
Requirements Phase	R4	Requirement were changed a lot [14].
	R5	OTS components could not be sufficiently adapted to changing requirements [14].
	R6	It is not possible to (re)negotiate requirements with the customer, if OTS components could not satisfy all requirements [9].
Component integration phase	R7	OTS components negatively affected system reliability [4,16].
	R8	OTS components negatively affected system security [4, 15, 16].
	R9	OTS components negatively affected system performance [16].
	R10	OTS components were not satisfactorily compatible with the production environment when the system was deployed [16].
System mainte-	R11	It was difficult to identify whether defects were in-

nance and evolution		side or outside the OTS components [14].
	R12	It was difficult to plan system maintenance, e.g. because different OTS components had asynchronous release cycles [13].
	R13	It was difficult to update the system with the last OTS component version [13].
Provider relationship management	R14	Provider did not provide enough technical support/training [4, 13].
	R15	Information on the reputation and technical support ability of provider were inadequate [4, 13].

We asked respondents to fill in whether these problems had actually happened in the investigated project or not. We used the same measurement as **RQ1.2**. The results are shown in Figure 8.



**Fig. 8.** Existing Problems in projects using COTS components vs. projects using OSS components

To compare differences, we compared the mean values of each risk in projects using COTS vs. projects using OSS. The significant results (P-value < 0.05) are shown in the following Table 4.

**Table 4.** The mean differences of three risks between COTS projects and OSS projects (Independent Samples T-Test without considering equal variances)

Risk	Mean Difference(COTS - OSS)	Significance (2-tailed)
R2	0.54	0.033
R5	0.39	0.027
R8	0.50	0.014
R15	-0.56	0.033

Answers to research question **RQ3** are summarized in Table 5.

**Table 5.** Answers to research question RQ3

<b>Research question</b>	<b>Commonalities (COTS and OSS)</b>	<b>Differences (COTS vs. OSS)</b>
RQ 3	Common risk is that it is difficult to estimate the integration effort and to identify whether the defects are inside the component or outside.	COTS users had a higher risk on estimating the selection effort, following requirement changes, and controlling COTS components' negative effect on security. OSS users had a higher risk on getting the information of providers' support reputation.

## 5. Discussion

### 5.1. Comparison with related work

By analyzing all activities normally performed when integrating OTS components, Giacomo, P. D. concluded that the OSS component is not completely different from the COTS component [24]. Our results show that both COTS and OSS components were used in projects and companies with similar profiles. Software houses and IT consulting companies were their main users.

COTS and OSS components promise shorter time-to-market and increased productivity [1]. Some previous studies claim that OTS components are generally more reliable than in-house built components, because they have been used by different users in different environments. Our results confirm that both COTS and OSS components were used in projects with emphases on time-to-market, reliability, and performance. In addition, the key motivations of using OTS components were to save development time and effort, and to get newest technology.

A COTS component generally is a black-box and is provided by the commercial vendor. Our results show that COTS users believe that the paid component should have good quality and good technical support. However, our results show that they were not more satisfied in reliability and performance of the components than OSS users. On the other hand, COTS users had more difficulties in controlling the components' negative security effects than OSS users. Some previous studies have argued that OSS software is more secure than COTS because its code can be reviewed by a huge number of coders. Our results give further support on this conclusion. In addition, our data shows that COTS users had more difficulties in estimating the selection effort and following requirement changes than OSS users. The possible reason is that COTS users cannot access the source code of the components. It is therefore difficult for them to review the source code and perform necessary changes when necessary.

OSS components are generally provided by an open source community with source code. Our results show that the main motivation of using OSS components is

the open and possible free code, which can be changed if necessary. Although OSS user did not have commercial support contract with OSS component providers, they were not less satisfied on the technical support than COTS users. However, we need to explain this fact carefully. The reason might be that OSS users would like to read and change the code themselves, instead of asking for help, as we found in the key motivations of using OSS.

Holck J. et al. concluded that a major barrier on using OSS in industrial project is the customer's uncertainty and unfamiliarity with OSS vendor relationship [23]. Our data showed that OSS users were less satisfied on knowing OSS provider's technical support reputation. How to establish credible and mutually acceptable combinations of OSS delivery and procurement models therefore needs future research.

## 5.2. Possible treats to validity

**Construct validity** In this study, most variables and alternatives are taken directly, or with little modification, from existing literature. The questionnaire was pre-tested using a paper version by 10 internal experts and 8 industrial respondents. About 15% questions have been revised based on pre-test results.

**Internal validity.** We have promised respondents in this study a final report and a seminar to share experience. The respondents were persons who want to share their experience and want to learn from others. In general, we think that the respondents have answered truthfully.

**Conclusion validity.** This study is still on-going. More data will be collected from Germany and Italy. A slightly larger sample will be gathered to give more significant statistical support on the conclusions of this study.

**External validity.** We used different random selection strategies to select samples in different countries. It is because the limited availability of the necessary information. In Italy, the information of the official organization as a national "Census Bureau" in Norway and Germany is not available. The samples had to be selected from "yellow pages". The method problems by performing such a survey in three countries will be elaborated in a future paper. Another possible limitation is that our study focused on fine-grained OTS components. Conclusions may be different in projects using complex and large OTS products, such as ERP, Content management systems, and web service in general.

## 6. Conclusion and future work

This paper has presented preliminary results of a state-of-the practice survey on OTS-based development in industrial projects. The results of this study have answered three questions:

- **RQ1: Who is using OTS components?**

Both COTS and OSS components were used in projects with similar profiles. There is also no difference on the profiles of companies using COTS or OSS components.

– **RQ2: Why was it decided to use OTS components?**

The main motivation of using either COTS or OSS component was to get shorter time-to-market, less development effort, and to get newest technology. COTS users have bigger trust in the COTS quality and COTS vendor support ability. Possible free source code is the key motivation of OSS users. OSS users prefer to have access the source code so that they can revise it when necessary.

– **RQ3: What are the possible problems of using OTS components?**

It was more difficult for COTS component users to follow requirement changes than OSS users. It is also more difficult for COTS users to estimate the selection effort and to control COTS components' negative effect on system security. OSS users were more uncertain on OSS provider support reputation than COTS users.

Results of this study have shown state-of-the-practice data. The next step is to do a follow up qualitative study to investigate the cause-effects of the conclusions of this study. We have held a seminar at OSLO in Feb. 2005 and preliminary investigated possible cause-effects of our results with participants of this study and other industrial colleagues. The next step is to do a larger qualitative study with personal interviews to further study these cause-effect assumptions.

## 7. Acknowledgements

This study was partially funded by the INCO (INcremental COmponent based development) project [17]. We thank the colleagues in these projects, and all the participants in the survey

## 8. References

1. Voas, J.: COTS Software – the Economical Choice?. IEEE Software, 15(2):16-19, March/April 1998.
2. Voas, J.: The challenges of Using COTS Software in Component-Based Development. IEEE Computer, 31(6): 44-45, June 1998.
3. Fitzgerald, B.: A Critical Look at Open Source. IEEE Computer, 37(7):92-94, July 2004.
4. Ruffin, M. and Ebert, C.: Using Open Source Software in Product Development: A Primer. IEEE Software, 21(1):82-86, January/February 2004.
5. Brownsword, L., Oberndorf, T., and Sledge, C.: Developing New Processes for COTS-Based Systems. IEEE Software, 17(4):48-55, July/August 2000.
6. Lawlis, P. K., Mark, K. E., Thomas, D. A., and Courtheyn, T.: A Formal Process for Evaluating COTS Software Products. IEEE Computer, 34(5):58-63, May 2001.
7. SSB (Norwegian Census Bureau) (2004): <http://www.ssb.no>
8. SESE web tool (2004): <http://sese.simula.no>
9. Li, J., Bjørnson, F. O., Conradi, R., and Kampenes, V. B.: An Empirical Study of Variations in COTS-based Software Development Processes in Norwegian IT Industry. Proc. of the

- 10th IEEE International Metrics Symposium, Chicago, USA, September, 2004, IEEE CS Press (2004) 72-83.
10. Paulson, J.W., Succi, G. and Eberlein, A.: An empirical study of open-source and closed-source software products. *IEEE Transactions on Software Engineering*, 30(4):246-256, April 2004.
  11. Norris, J. S.: "Mission-Critical Development with Open Source Software", *IEEE Software*, 21(1):42-49, January/February 2004.
  12. Fitzgerald, B. and Kenny, T.: Developing an Information Systems Infrastructure with Open Source Software. *IEEE Software*, 21(1):50-55, January/February 2004.
  13. Rose, L. C.: Risk Management of COTS based System development. Cechich, A., Piattini, M., Vallecillo, A. (Eds.): *Component-Based Software Quality - Methods and Techniques*. LNCS Vol. 2693. Springer-Verlag, Berlin Heidelberg New York (2003) 352-373.
  14. Boehm, B. W., Port, D., Yang, Y., and Bhuta, J.: Not All CBS Are Created Equally COTS-intensive Project Types. Proc. of the 2<sup>nd</sup> International Conference on COTS-Based Software Systems. Ottawa, Canada, February, 2003, LNCS Vol. 2580. Springer-Verlag, Berlin Heidelberg New York (2003) 36-50.
  15. Open Source Initiative (2004): <http://www.opensource.org/index.php>
  16. Padmal Vitharana: Risks and Challenges of Component-Based Software Development. *Communications of the ACM*, 46(8):67-72, August 2003.
  17. INCO project description, 2000, <http://www.ifi.uio.no/~isu/INCO>
  18. Stevens, S.S: *Mathematics, Measurement, and Psychophysics*. In S.S.Stevens (Ed.) *Handbook of Experimental Psychology* (1951), New York: Wiley.
  19. Stevens, S.S: *On the Theory of Scales of Measurement*, Vol. 103. *Science* (1946) 677-680.
  20. Spector, P.: Ratings of Equal and Unequal Response Choice Intervals. *Journal of Social Psychology* (1980), Vol. 112:115-119.
  21. Velleman, P. F. and Wilkinson L.: Nominal, Ordinal, Interval, and Ratio Typologies Are Misleading. *Journal of the American Statistician*, 47(1):65-72, February 1993.
  22. Hand, D.J.: *Statistics and Theory of Measurement*. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 159(3): 445-492, 1996.
  23. Holck J., Larsen, M. H., and Pedersen, M. K.: Managerial and Technical Barriers to the Adoption of Open Source Software. Proc. of the 4<sup>th</sup> International Conference on COTS-Based Software Systems. Bilbo, Spain, February, 2005, LNCS Vol. 3412. Springer-Verlag, Berlin Heidelberg New York (2005) 289-300.
  24. Giacomo, P. D.: COTS and Open Source Software Components: Are They Really Different on the Battlefield? Proc. of the 4<sup>th</sup> International Conference on COTS-Based Software Systems. Bilbo, Spain, February, 2005, Vol. 3412. Springer-Verlag, Berlin Heidelberg New York (2005) 301-310.