Scientific Journal of Riga Technical University
Computer Science. Applied Computer Systems
2010
Volume 43

Reuse in Software Development Organizations in Latvia

Vladimir Kotov, *Riga Technical University*

***Abstract* - The paper presents the results of a survey, which aims to investigate the field of software reuse in software development organizations in Latvia. The topic has a particular significance since reuse may provide some economic benefits by reducing software development and operating costs, and by efficient utilization of development knowledge and corporate expertise. The objective of the study is to identify the key factors to be considered by the companies interested in establishing a software reuse program, thus improving time-to-market, costs and quality of software products by ingraining reuse into the entire software development process. Twenty factors organized into four categories were considered important within the framework of the survey. The results were obtained through the questionnaire involving software organizations in Latvia, whose responses were analyzed and used to relate the characteristics of organizations with their reuse experience. As a result, the influence of the selected factors on the success of the reuse is evaluated using four different levels (strong, weak, none, no data). Additionally, the paper includes review of related researches and comparison of the results. In conclusion, the author provides suggestions for the improvement of the reuse in Latvian software development organizations on the basis of the survey.**

Keywords: **software reuse, software reuse factors, survey**

I. Introduction

Software reuse is the process of creating software systems from existing software rather than building them from scratch [1], whereby an organization defines a set of systematic operating procedures to specify, produce, classify, retrieve, and adapt software artifacts with the purpose of using them in its development activities [2], so that similarities in requirements and/or architecture between applications can be exploited to achieve substantial benefits [3].

With more than 40-year history software reuse is recognized as an important mechanism to improve software quality and development productivity when properly understood and systematically deployed [4]. The topic has a particular significance in the current situation of financial crisis both in Latvia and all over the World, since reuse may provide some economic benefits by reducing software development and operating costs, and by efficient utilization of development knowledge and corporate expertise.

Retrospective analysis of the origins and main contributions in the research area performed in [4] identifies the following important aspects of the current state of the software reuse field:

- Software reuse is multidisciplinary and has deep and complex interactions with many areas of computer science.
- Reuse of large-scale software system components is more effective when systematically applied in the context of families of related systems.
- The final goal is to ingrain reuse into the entire software production process in organizations [5].

The end of the 20th century can be characterized as evolving of systematic software reuse as a field of research in software engineering [4]. Important contributions include analysis of organizational experience in [6] used to answer questions commonly asked by the organizations attempting to implement systematic reuse and analysis of the factors which influence establishing a software reuse program in [7] derived from the survey of projects for the introduction of reuse. However, even with some experiences from industry we need more empirical data about introducing reuse in software development organizations [8].

Willing to improve their time-to-market, costs and quality of software products by ingraining reuse into the entire software development process, organizations are usually faced to practical questions, both technical and non-technical, related to a) a kind of development approach to be used, b) organization of the development team, c) languages, tools and application domains more suitable for reuse and d) the steps needed to introduce reuse in the organization. Besides, according to [8] it is difficult to determine exactly what makes a reuse successful, mainly because it depends on the structure of the organization, its background, and budget, among other technical and non-technical factors.

This paper presents the results of the survey, which aims to investigate the field of software reuse in software development organizations in Latvia with the objective to identify the key factors to be considered by the companies interested in establishing a software reuse program. Section II provides a brief review of the survey, the questionnaire and data analysis approach. It is followed by discussion of the survey results and main reuse success factors in Section III. The related researches are reviewed and the conclusions are made in Sections IV and V.

II. The Survey

The survey is based on the questionnaire involving software organizations in Latvia, whose responses were analyzed and used to relate the characteristics of organizations with their reuse experience.

The questionnaire was originally conducted by researchers from the Federal University of Pernambuco. The original version from [8], which was developed after an extensive review of the literature and based on industrial evidence

experienced by RiSE (Reuse in Software Engineering) research group, was refined and adapted for the Baltic market. The adapted revision was further reviewed and accomplished by a group of experts, both academic and software development industry professionals.

The questions could be divided into three groups:

a) general information about the organization;
b) information about its reuse experience;
c) information indirectly related to reuse (such as development team experience and software development approaches among the others).

Questions of the groups "a)" and "c)" are aimed at identifying possible factors that could have some influence on reuse success. Within the framework of the remaining survey twenty factors, adapted from [8] and supplemented, are considered important. The factors are organized into four categories as outlined in Table 1.

Answers to the questions from the group "b)" were used to assess whether organizations were successful in reuse by means of productivity or quality gains. It can not be treated as formal evidence, because claims on reuse success are mostly based on subjective analysis. However, as correctly noted in [8], it allows to capture the industry professionals' impressions on software reuse and to determine if some kind of benefit has been obtained.

The questionnaire was distributed among several software development organizations in Latvia, and the completed response forms were submitted from eighteen organizations. Both quantitative and qualitative data analysis was performed, and assessment was done based on the percentage of claimed success related to each characteristic, answers additionally included to the survey and our own experience.

In order to perform further comparison with related studies, four different levels of influence that a factor may have on reuse success are used:

1. strong, when the observed data clearly indicates a relationship between that factor and reuse success;
2. weak, when some relationship exists;
3. none, when no relationship exists;
4. no data.

III. Results

The findings derived from the questionnaire results suggest some interesting trends. In total about twenty-five software development organizations in Latvia were contacted and responses from eighteen were received.

No special restrictions were imposed for the survey participants. As a result, organizations of different size and development profile answered the questionnaire. In general, participants can be divided into the following groups:

- software development companies;
- IT outsourcing service providers;
- IT departments of financial or service institutions;
- educational institutions.

As outlined in Fig. 1 thirteen (72%) participants claimed to succeed in projects by the means of software reuse in their organization. The distribution of roles occupied by respondents in Fig. 2 indicates that both management and technical points of view were taken into consideration during the survey.

TABLE 1
Reuse Success Factors

Category	Factors
Organizational	1. organization and team size 2. project team experience 3. software reuse education 4. rewards and incentives 5. independent reusable asset development team
Business	6. product family approach 7. kind of software developed 8. application domain
Technological	9. software development approach 10. programming language 11. usage of software models 12. repository systems usage 13. CASE tools usage
Processes	14. quality models usage 15. systematic reuse process 16. kind of reused assets 17. specific function in the software reuse process 18. software reuse measurement 19. software certification process 20. configuration management of the reusable assets

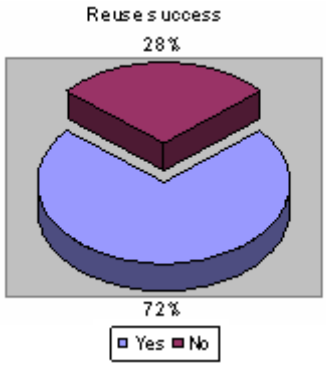

Fig. 1. Reuse success

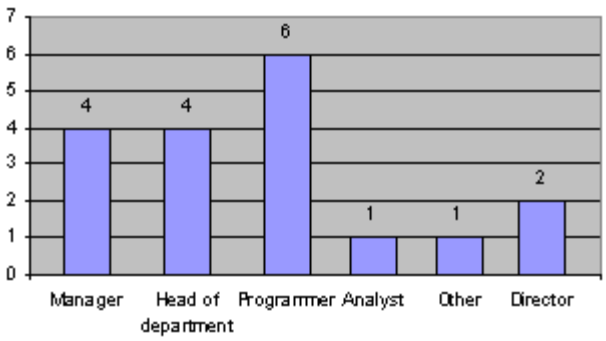

Fig. 2. Participants of the survey

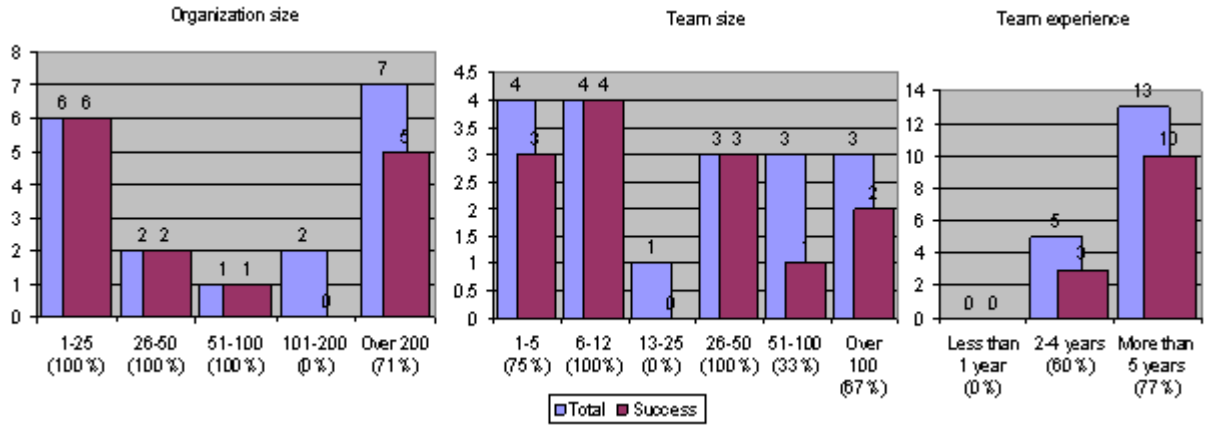

Fig. 3. Influence of organization size, team size and team experience on reuse success

A. Organizational Factors

Several characteristics of the organizations were observed for possible influence on the claimed success in reuse. The results of Fig. 3 show not statistically evident relation to organization's size, where small and medium software organizations with 1–100 employees presented higher reuse success rates than organizations of a large size. Similar phenomenon observed in [8] is explained by the following factors: a) more restricted domain scope and limited resources, that makes smaller organizations to have software reuse as their goal; b) better communication flows, and project managers being closer to the development team in smaller organizations; c) difficulty to introduce changes related to software reuse in large organizations because of the need in long-term investments and time to deploy.

Similar pattern could be observed in terms of software development team sizes (see Fig. 3), that could be explained by more opportunistic nature of reuse in smaller teams, which becomes hard to achieve in the large teams, where more controllable process is usually established. Since only few respondents reported about the use of any systematic reuse process, this fact complied with importance of reuse process as an effective way to obtain the benefits of software reuse for large software organizations emphasized in [7], [9], [10]. There is no reasonable correlation between experience and reuse success from the questionnaire results. In this context the related study [6] generally assumes, that software engineering experience has no effect on reuse of lifecycle objects, that maybe caused by historical lack of training in the reuse.

The survey results show that the reuse education is poor in Latvia, and only one respondent (5%) reported about existing training programs related to software reuse, such as: reuse-oriented approaches, component-based software engineering and design patterns. Similarly, corporate reuse education is also rare in US (19%) as previously observed in [6], and only few (15%) of the organizations interviewed in Brazil in [8] implement some policies related to reuse education. Interesting enough, organization confirmed existence of reuse education, has not reported success by means of reuse. Similar results were obtained concerning rewards and incentives as a stimulus to effective introduction of reuse inside organization – only one respondent (5%) answered this question affirmatively, considering motivation in form of recognition.

It is commonly assumed, that creating an independent team can influence good reuse results, because this team will be specialized in reuse, building more robust and well elaborated assets, minimizing the effort to reuse them [8]. However, the practice when one team develops the reusable assets, while another team develops the project, was confirmed by one organization (5%) participating in the survey. Therefore, we considered not enough data to conclude influence of software reuse education, rewards and incentives, and independent reusable assets development team on software reuse.

B. Business Factors

Although particular type of software is not known to have influence on reuse success, related questions were included to the survey and organizations were offered to classify themselves into six categories of developed software: embedded, games, information systems, intelligent systems, real-time systems and other. As outlined in Fig. 4 information systems are the most common, while games and embedded software are not being intensively developed by the interviewed organizations in Latvia (thus not included). Additionally, possible application domains were suggested for participants to choose. It should be noted, that some respondents have selected multiple options answering these questions. In general, the percentages of reuse success for observed kinds of software are similar and results for application domains are not conclusive enough, that could indicate such factors not being important for reuse.

Factor related to software production process, which fall into category of business factors, was considered to be important for analysis. Respondents were asked if the organization produces specific software for each project (isolated products), or develops a product that evolves over time, being adapted for each new project in some way (product families). A software product family also referred to as product line is a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a prescribed

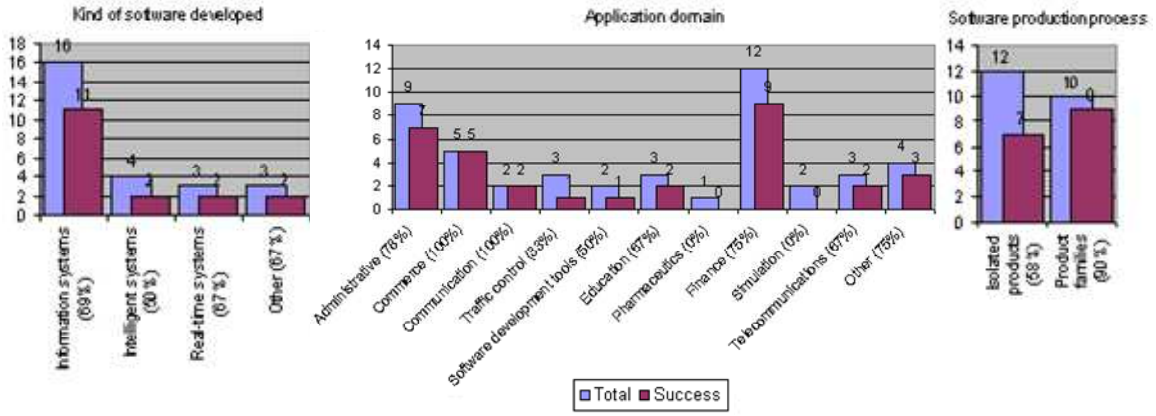

Fig. 4. Influence of kind of software, application domain and software production process on reuse success

way [11]. Comparing to conventional "monolithic" set of tasks software organizations are responsible for providing and maintaining software systems, product family approach introduces a differentiation between the tasks related to the production of reusable assets and the tasks related to the production of end-user applications. According to [6], [7], [12] this can bring benefits to software reuse, mainly because product families have functionalities that can be shared and the scope of the reusable assets should be defined in the early stages of development. 90% of the organizations being investigated that adapted a product family approach claimed to obtain success in reuse, while 58% have succeeded from others. Three organizations claiming success in reuse reported both isolated product and product family approaches being adapted. Therefore, product family approach was considered to have a strong influence on reuse success.

C. Technological Factors

From the technical perspective software reuse is often understood as advantage of the object-oriented methods, and component engineering discipline is usually treated as the most popular way to reuse software [4]. However, this contradicts with [7], where both development paradigm and the programming language are not considered influencing reuse success. During the survey the following results (see Fig. 5) were obtained about the influence of development approach on reuse success in software development organizations in Latvia: component-based development approach (80%), followed by the object-oriented approach (79%). Surprisingly, procedural development approach has very high reuse success percentage (80%). However, this phenomenon can be explained by the fact, that procedural development is specified as complimentary to the object-oriented approach in responses from several organizations claiming to obtain success in reuse. During analysis of the impact of the programming language on reuse it was observed that the software organizations that use Ruby, Java and Object Pascal have a greater reuse success level. However, results do not vary strongly across the different languages as could be seen on Fig. 5. Based on these observations, we conclude that both development approach and programming language have a weak influence on reuse.

It is concluded in [6] and [7]: a) the use of repositories as searchable collections of reusable assets is important, but should not be the primary focus for organizations trying to improve systematic reuse; b) many organizations regard CASE tools as a way to improve reuse, however, those are yet not effective in promoting reuse. In this context, the survey participants were asked, if there is a repository for storing and retrieving reusable assets, and if CASE tools are used in the development of software projects in their organization. The collected data shows that 90% (9 out of 10) of the organizations that claimed to have repository reported reuse

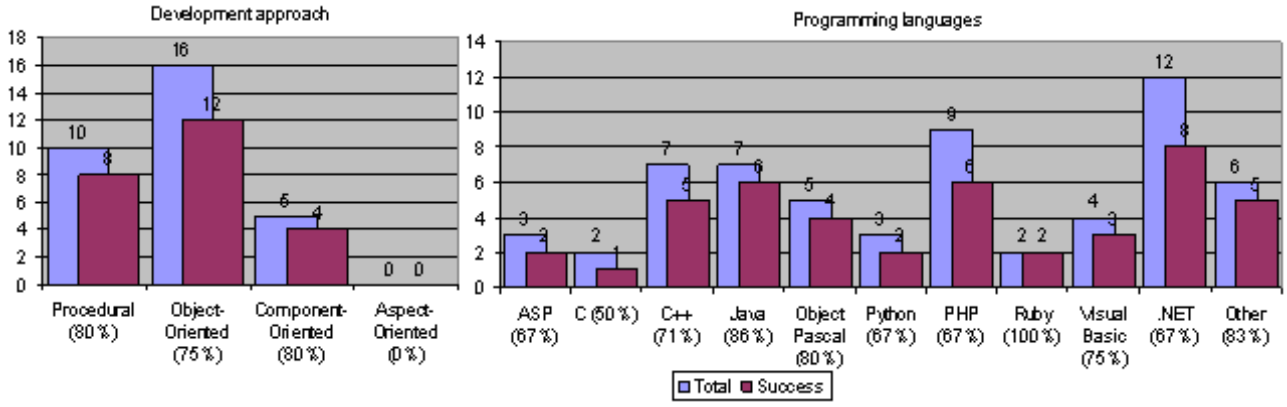

Fig. 5. Influence of development approach and programming languages on reuse success

success, against 50% (4 out of 8) of success for the organizations that do not. Additionally, 75% (6 out of 8) of the respondents, who use CASE tools in their projects, reported reuse success, against 70% (7 out of 10) for the organizations that do not. Therefore, we conclude that using repository for storing and retrieving reusable assets has a strong influence on reuse success, while usage of CASE tools has not. In fact, we observed the opposite to results in [8], where the percentage of success in organizations without a repository is superior and reuse repository is considered to have no influence on software reuse success, while CASE to have a strong influence.

Additional question included into the questionnaire is about the role assigned to software models in development process of the organization. The gains of the successful application of model-driven development approach include several benefits in the reuse area apart from gains in productivity, interoperability, portability and maintenance, as models may be used to generate a code for different platforms. From respondents answered this question 100% (4 out of 4) of the organizations, who use models as first-class development artifacts and have adapted MDA (Model-Drive Architecture) approach, claimed to achieve success by means of reuse. 80% (4 out of 5) of the organizations, who use models mostly for documentation purposes, also reported reuse success, against 43% (3 out of 7) of success for the organizations, where models are not considered explicitly. Therefore, the usage of software models in the development process is considered as factor with a strong influence on reuse success.

D. Process Factors

Importance of a systematic reuse process as an effective way to obtain the benefits of software reuse is emphasized in [7, 9, 10]. Being process-driven means that the software development is done in accordance with well defined processes that are enforced through management policies, and it is important in order to ensure efficiency, reproducibility, homogeneity, and predictable time and effort constraints. Software processes refer to all the tasks necessary to produce and manage software, whereas reuse processes are the subset of tasks necessary for successful software reuse within a company [7]. Only two participants confirmed the adaption of any systematic reuse process in their organization, whereof we considered that there is not enough data to identify its influence in our survey.

The central concept in reuse area is a reusable asset or artifact that could be either reusable software or software knowledge, potentially made up of many software life cycle products including requirements and architecture definitions, analysis and design models, code, test scenarios and reports [4]. Fig. 6. summarizes the answers about the type and the origin of asset systematically reused in respondents' organizations.

Quality and maturity models are used to define and evaluate the process that is used by an organization, in order to assure the quality of the final product [8]. The survey results show that 82% (9 out of 11) of the software organizations, who established a quality model, obtained success in software reuse as well, while 57% (4 out of 7) of the respondents, who have not established a quality model, managed to achieve success in reuse. So, a strong influence of quality models usage on reuse success is evident.

Configuration management process is fundamental to help in maintaining the assets' quality during their evolution. The survey tried to identify how organizations are using it for reusable assets, as well as its impact on reuse success. The obtained data shows, that 27% (5 out of 18) of the organizations use configuration management, and 80% (4 out of 5) of the participants who introduced it obtained success in reuse, while only 69% (9 out of 13) of organizations that do not introduced it obtained success in reuse. Based on these observations, we consider configuration management of the reusable assets to be a factor with strong influence on reuse success.

Additionally, several questions were included into the questionnaire with the aim to verify if some of the following factors often mentioned in reuse related researches can influence the success of reuse programs: specific position responsible for the software reuse process, measurement of reuse level and presence of asset certification process to assure its reusability. However, we received no data enough to identify the influence of those factors on reuse success.

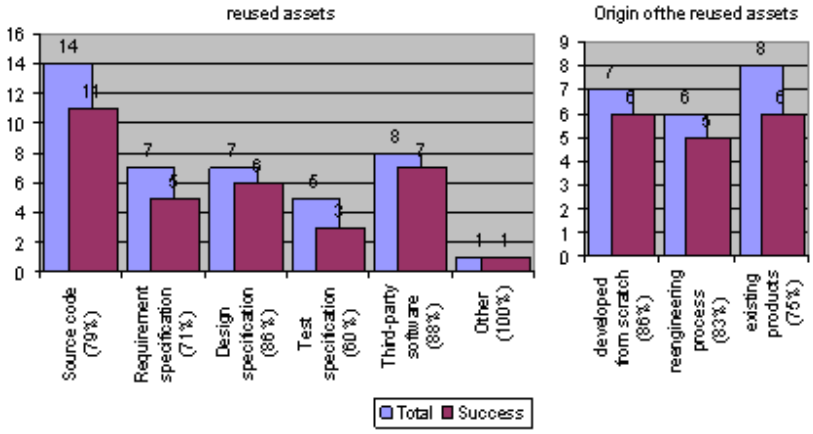

Fig. 6. Influence of kind and origin of reused assets on reuse success

IV. Related Researches

There are a few similar studies that aim to analyze factors related to a successful introduction of reuse. One of the first analysis of the organizational experience in [6] used the data obtained from the responses of twenty-nine respondents to answer sixteen questions commonly asked by organizations attempting to implement systematic reuse. The study highlighted a perceived economic feasibility, reuse education, type of industry, common software process and high quality assets as the factors to concentrate on to improve systematic reuse.

In [7] some of the factors in establishing a company-wide software reuse program were derived from empirical evidence of reuse practices from a survey of twenty-four projects elaborated for the introduction of reuse in European companies. The study concluded that despite the potential for success, about one-third part of the projects failed due to not introducing reuse processes, not modifying non-reuse processes, not considering human factors and a lack of commitment by top management, which could be considered as important factors for a reuse program.

Based on the research results in [12], the leading indicators of software reuse capability are the following: product-line approach, architecture that standardizes interfaces and data formats, common software architecture across the product-line, design for manufacturing approach, domain engineering, management which understands reuse issues, software reuse advocate(s) in senior management, state-of-the-art tools and methods, precedence of reusing high level software artifacts such as requirements and design versus just code reuse, and trace end-user requirements to the components that support them.

Recent survey performed in [8] that is used as a basis for the present study, analyzed practical experience reported by industry professionals, through a survey involving fifty-seven Brazilian small, medium and large software organizations. As a result, the following most influencing reuse success factors were marked out: independent reusable assets development team, product family approach, CASE tools usage, quality models usage, systematic reuse process, kind of reused assets, previous development of reusable assets and configuration management of the reusable assets. Besides, some other factors, for instance, an application domain, software development approach, programming language and origin of the reused assets also influence the reuse success in a smaller degree according to [8]. In addition, the study presented a guide pointing out the factors that should be more strongly considered by small, medium and large organizations attempting to establish a reuse program. The detailed comparison of the results of the present study and the results presented in [8] is provided in Table 2.

TABLE 2
Comparison of Survey Results with Related Works

Factor	Influence on reuse success	
	Survey results	Results from [8]
Organizational factors		
1. Organization and team size	Weak	None
2. Project team experience	None	Weak
3. Software reuse education	No data	None
4. Rewards and incentives	No data	No data
5. Independent reusable asset development team	No data	Strong
Business factors		
6. Product family approach	Strong	Strong
7. Kind of software developed	None	None
8. Application domain	None	Weak
Technological factors		
9. Software development approach	Weak	Weak
10. Programming language	Weak	Weak
11. Usage of software models	Strong	No data
12. Repository systems usage	Strong	None
13. CASE tools usage	None	Strong
Processes factors		
14. Quality models usage	Strong	Strong
15. Systematic reuse process	No data	Strong
16. Kind of reused assets	Strong	Strong
17. Specific function in the software reuse process	No data	No data
18. Software reuse measurement	No data	No data
19. Software certification process	No data	No data
20. Configuration management of the reusable assets	Strong	Strong

V. CONCLUSIONS

Software reuse is recognized as important mechanism to improve software quality and development productivity when properly understood and systematically deployed.

The paper presents the results of the survey, which aims to investigate the field of reuse in software development organizations in Latvia with the objective to identify the key factors to be considered by organizations in order to introduce and succeed from a reuse program.

The influence of the selected factors on the success of the reuse is summarized in Table 2. The following factors are considered to be the most influencing:

a) Processes factors – 14, 16, 20; b) Technological factors – 11, 12; c) Business factors – 6. However, there was not enough data received during the survey to analyze some factors and properly relate them to the success of the reuse.

Latvian software development organizations are different in size and development profile, and the amount of empirical evidence from the survey is too small for pure statistical approach. Nevertheless, it is possible to make the following suggestions for the improvement of the reuse on the basis of the survey:

- Organizations should focus on the development of product families if applicable in the business area they are operating in.
- Whenever possible, reusable assets should be previously developed, and a configuration management process should guarantee proper evolution of these assets. Organizations should consider using repository for storing and retrieving reusable assets.
- Organizations are suggested to look towards model-driven development approach when possible as it provides several benefits in the reuse area apart from gains in productivity, interoperability, portability and maintenance.
- More attention should be spent on the quality models and systematic reuse processes in the organization.

REFERENCES

[1] C. W. Krueger, "Software reuse," *ACM Computing Surveys*, vol. 24, New York: ACM, 1992, pp. 131-183.

[2] H. Mili, A. Mili, S. Yacoub, and E. Addy, *Reuse-Based Software Engineering: Techniques, Organization, and Measurement*. John Wiley & Sons, Inc., 2002.

[3] M. Ezran, M. Morisio, and C. Tully, *Practical Software Reuse*. Springer, 2002.

[4] V. Kotov, "Forty years of software reuse," in *Scientific Proceedings of Riga Technical University*, Series: Computer Science, Vol. 38. Riga: RTU Publishing House, 2009, pp. 153-160.

[5] "Software reuse and domain engineering," ReNews blog. [Online]. Available: http://wfrakes.wordpress.com/category/introduction. [Accessed: September 2009].

[6] W. B. Frakes and C. J. Fox, "Sixteen questions about software reuse," *Communications of the ACM*, vol. 38, no. 6. New York: ACM, 1995, pp. 75-87.

[7] M. Morisio, M. Ezran, and C. Tully, "Success and failure factors in software reuse," *IEEE Transactions on Software Engineering*, vol. 28, no. 4, 2002, pp. 340-357.

[8] D. Lucrédio, K. dos Santos Brito, A. Alvaro, et al. "Software reuse: the Brazilian industry scenario," *Journal of Systems & Software*. vol. 81, no. 6, 2008, pp. 996-1013.

[9] M. L. Griss and M. Wosser, "Making software reuse work at HP," HP Labs technical reports, Hewlett-Packard Company. [Online]. Available: http://www.hpl.hp.com/techreports/95/HPL-95-04.html. [Accessed: September 2009].

[10] B. Frakes, Kyo Kang. "Software reuse research status and future," *IEEE Transactions on Software Engineering*, vol. 31, no. 7, 2005, pp. 529-536.

[11] P. C. Clements, "FAQs: an introduction to software product lines," Software Engineering Institute. [Online]. Available: http://www.sei.cmu.edu/library/abstracts/news-at-sei/productlines20053.cfm. [Accessed: September 2009].

[12] D. Rine and R. Sonnemann, "Investment in reusable software. A study on software reuse investment success factors," *The Journal of Systems & Software*, vol. 41, 1998, pp. 17-32.

Vladimir Kotov has received the M.Sc.ing. degree in Computer science from Riga Technical University in 2007. Since 2007, he is a PhD student in Riga Technical University, Faculty of Computer Science and Information Technology, Institute of Applied Computer Systems.

Nowadays, he works in SIA Baltic Technology Group as Software Developer.

Vladimirs Kotovs. Atkārtotā lietošana programmatūras izstrādes organizācijās Latvijā
Rakstā ir sniegts pārskats par aptaujas rezultātiem, kuras mērķis ir izpētīt atkārtoto lietošanu programmatūras izstrādes organizācijās Latvijā. Programmatūras atkārtotā lietošana var nodrošināt vairākus ekonomiskos labumus, samazinot programmatūras izstrādes un ekspluatācijas izmaksas, kā arī efektīvi atkārtoti izmantojot izstrādātāju zināšanas un pieredzi. Pētījuma mērķis ir identificēt galvenos faktorus, kuri jāņem vērā kompānijām ieinteresētām atkārtotās lietošanas programmas ieviešanā organizācijas līmenī. Tiek uzskatīts, ka atkārtotās lietošanas ieviešana programmatūras izstrādes procesā ļauj uzlabot programmatūras produkta piegādes laiku līdz tirgum, izstrādes izmaksas un kvalitāti. Divdesmit faktori organizēti četrās kategorijās tika uzskatīti par svarīgiem pētījuma ietvaros. Rezultāti tika iegūti, veicot anketēšanu Latvijas programmatūras organizāciju vidū, kuru atbildes tika analizētas un izmantotas, lai sasaistītu organizāciju pazīmes ar to atkārtotās lietošanas pieredzi. Rakstā tiek arī iekļauts saistīto pētījumu pārskats un rezultātu salīdzināšana. Noslēgumā autors sniedz ierosinājumus atkārtotās lietošanas uzlabošanai programmatūras izstrādes organizācijās Latvijā balstoties uz aptaujas rezultātiem.

Владимир Котов. Повторное использование в организациях, занимающихся разработкой программного обеспечения в Латвии
В статье представлены результаты исследования, направленного на изучение повторного использования в организациях, занимающихся производством программного обеспечения в Латвии. Особое значение повторного использования в области разработки программного обеспечения обуславливается возможностью получения определенной экономической выгоды за счет сокращения расходов на разработку программных продуктов, а также эффективного повторного использования знаний и опыта разработчиков. Целью исследования является выявление основных факторов, которые необходимо иметь в виду компаниям, заинтересованным в интеграции повторного использования в процесс разработки программного обеспечения на уровне организации с целью снижения времени выхода продукта на рынок, улучшения стоимости и качества разработки. Результаты были получены путём анкетирования организаций, занимающихся разработкой программного обеспечению в Латвии, чьи ответы были проанализированы и использованы для установления зависимости между характеристиками организаций с их опытом повторного использования. Автор также дает предложения по улучшению повторного использования в организациях, занимающихся производством программного обеспечения в Латвии.