SPECIFICS OF PROJECTS IN THE AREA OF ENTERPRISE ARCHITECTURE DEVELOPMENT

Dmitrij V. KOZNOV
Associate Professor, Department of Software Engineering, Saint Petersburg State University
Address: 28, Universitetsky Prospect, Stary Peterhof, Saint Petersburg, 198504, Russian Federation
E-mail: st008149@spbu.ru

Maxim Yu. ARZUMANYAN
Assistant Professor, Department of Information Technologies in Economics,
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications;
Smart Architects Ltd.
Address: 22, Prospect Bolshevikov, Saint Petersburg, 193232, Russian Federation
E-mail: maxim.arzumanyan@gmail.com

Yury V. ORLOV
General Director, Smart Architects Ltd.
Address: 61A, Moyka, Saint Petersburg, 191186, Russian Federation
E-mail: yury.orlov@smartarchitects.ru

Maxim A. DEREVYANKO
Engineer, Department of Information Technologies in Economics, The Bonch-Bruevich
Saint Petersburg State University of Telecommunications; Smart Architects Ltd.
Address: 22, Prospect Bolshevikov, St. Petersburg, 193232, Russian Federation
E-mail: maxim.derevyanko@gmail.com

Konstantin Yu. ROMANOVSKY
Senior Lecturer, Department of Software Engineering, Saint Petersburg State University
Address: 28, Universitetsky Prospect, Stary Peterhof, Saint Petersburg, 198504, Russian Federation
E-mail: k.romanovsky@spbu.ru

Anastasia A. SIDORINA
Student, Department of Computer Science, Saint Petersburg State University
Address: 28, Universitetsky Prospect, Stary Peterhof, Saint Petersburg, 198504, Russian Federation
E-mail: sidorina-nastia@mail.ru

Enterprise Architecture (EA) is a discipline for business and IT system management, describing the fundamental artifacts of business and IT. Introducing EA activities in a company is done via special projects (referred to as EAM projects). While much research focuses on EA as is, EAM projects are not yet considered properly. This paper deals with EAM projects as a special kind of IT project. The definition of an EAM project is suggested using the IT solution concept of the Microsoft Solution Framework. This is justified, since the EAM-software introduced in a company is the main result of a typical EAM project. Based on the definition of a work product in CMMI, this paper introduces the final delivery of an EAM project. In addition, domain specific modeling is used to describe EAM project delivery, since the most important part of EA is concepts and terms for describing business and IT in a company. To implement the language formed by these terms and concepts, an EAM tool is selected and customized (Mega, IBM System Architects, Aris, etc.), and additional software (i.e. Web-portal, integration scripts, etc.) is implemented. All of the EA principles and methods could be considered as guidelines for this software (EAM tool and additional software). Final delivery of an EAM project is divided into the following parts: method (domain specific language, EA method, EA process, integration, modeling results), technology (EAM tool, additional software, documentation), support and training. Using the concepts introduced in this paper, four EAM projects are analyzed. Lack of support of the EA method is identified as a common problem of these projects.

Enterprise Architecture (EA) is a discipline for business and IT system management, describing the fundamental artifacts of business and IT. Establishment of EA activities in a company is implemented in special projects (referred to as EAM projects). While much research focuses on EA as is, EAM projects are not yet considered properly. This paper deals with EAM projects as a special kind of IT projects. A definition of EAM project is suggested using IT solution concept of Microsoft Solution Framework. It is justified since EAM-software introduced in a company is the main result of a typical EAM project. Basing on definition of work product in CMMI, the paper introduces the final delivery of EAM project. In addition, domain specific modeling is used to describe EAM project delivery, since the most important part of EA are concepts and terms for describing business and IT in a company. To implement the language formed by these terms and concepts, an EAM tool is selected and customized (Mega, IBM System Architects, Aris, etc.), and additional software (i.e. Web-portal, integration scripts, etc.) is implemented. All of EA principals and methods could be considered as guidelines for this software (EAM tool and additional software). Final delivery of an EAM project is divided into the following parts: method (domain specific language, EA method, EA process, integration, modeling results), technology (EAM tool, additional software, documentation), support and training. Using concepts introduced in the paper, four EAM projects are analyzed. A lack of support of EA method is identified as a common problem of these projects.

Introduction

The modern market is mobile and volatile due to a high competition, changing technology, as well as external circumstances (political, geographical and so on.). Therefore, companies to be successful, have to change on a permanent basis. To ensure efficient management and renovation of companies, the scientific and practical area referred to as *Enterprise Architecture Management* (EAM) — analysis, design, planning and implementation of the company activity based on a systemic concept [7, 21] is currently rapidly growing. This activity is supported by special software products (hereinafter referred to as EAM-tools), such as ARIS [9], Mega [22] and some others. The introduction of architecture management progress is generally accompanied by creation of solutions based on EAM-tools, and in most cases implemented within special projects, which will be called EAM-projects. In implementation of EAM-projects, standard tools are adapted to the individual peculiarities of the customer company, namely methodologies and standards (TOGAF, DoDAF, Archimate et al. [12]), graphic languages and notations [16, 19, 20], as well as software tools [11].

The purpose of this paper is to define EAM-projects, considering them as specific IT projects. The resulting delivery of an EAM-project is also investigated, with the identification of its typical Work Products [13] that will make it possible to define the scope of implementation and scope of works of such projects. In addition to the above, methods of Software Engineering and Domain-Specific Modeling (DSM) are used [18].

1. Basic definitions

1.1. Enterprise Architecture Management

Enterprise Architecture[1] (EA) covers the basic principles of its existence and its development either as an isolated structure, or together with partners, suppliers and/or buyers. Specifically, the enterprise can be considered as a whole or partially (for example, any single business area or department) [24]. Enterprise architecture is generally divided into IT architecture and business architecture. The development of IT architecture involves systematizing and coordinated development of the company's IT infrastructure, i.e. information systems, platforms, networks, hardware, etc. The business architecture management consists

---

[1] The enterprise is taken to mean a business company or government agency.

in identification and description of the business tasks of the company, its functional-role structure, business processes, etc.

Visual modeling plays the key role in the implementation projects of architecture management: the company activity and infrastructure are described by a set of models that makes these descriptions available to a wide range of specialists. The rapidly growing market of EAM-tools [11] is largely a modeling tools market.

1.2. CMMI and Work Products

In the early 1990s, the CMM (Capability Maturity Model) standard for certification of military and federal contractors in the area of software development was developed in the United States. In 2000, a generalized standard version called CMMI (Capability Maturity Model Integrated) and combined development of software and non-software components of the complicated artificial systems was released. Currently this standard is widely used in global industry and, among other things, is a known glossary of IT terms. In particular, it provides a definition of a Work Product as a significant result of the development process which can be a software component, document, invoice, established process (for example, process of using software by end-users), etc. [13].

1.3. IT solution in MSF

Early in 2000, Microsoft created a software development methodology called the Microsoft Solution Framework (MSF). One of the basic MSF concepts was the IT solution — software that is not «packaged» and is created to meet the needs of a particular customer. An IT solution in MSF is defined as the coordinated delivery of a set of elements (work products in CMMI terms) required to meet some business need of a specific customer. The examples can include an application software code, documentation, communications, implementation process, training, support [23].

1.4. DSM-approach
in software engineering

Domain specific modeling is a policy of creating software that increases the level of abstract development compared to traditional means by providing applications specifications in terms of the data domain [18].

This paper [5] draws parallels between software engineering and enterprise architecture management. In particular, it is shown that the use of visual modeling has much in common in these areas: it is a marketable product, and mature software tools are widely used in these areas. Finally, domain specific modeling is applied in these areas, thereby making it possible to configure standard tools (languages, methods, software modeling tools) for a specific task or develop new tools. This paper [5] also generalizes a definition of DSM solutions oriented in both data domains. However, it happened to be too general and did not cover all work products included in EAM-project delivery.

2. EAM-project

Generally, the company architecture development activities are organized in the form of one or several projects: a company is able to focus on the next round of changes and unification by providing appropriate resources. It is reasonable to create methods and tools to solve company problems within such projects. However, final implementation of the proposed methods and tools in the management processes, as well as their full use, shall be carried out by the company on its own. Therefore, it is important that the architecture description can be added and changed, and that this can be done by company employees. To do this, in the course of the EAM-project an appropriate IT solution is implemented in the company.

There are many projects (especially in Russia) on development company architecture which are restricted to generation of static descriptions of company management processes (as is or to be models). There are also projects within which new regulations, standards and formal procedures are generated. This can also be an EA-activity if it is implemented based on unified principles. To accomplish this, both activities are often performed by external consulting organizations. However, we will call EAM-projects only those architecture management projects within which EAM-tools are implemented (often with modification), and which, therefore, have a significant IT component.

3. Definition
of EAM-solution

Let us call the EAM project result an EAM-solution. Its main difference from MSF IT solution is that in the first case, the software percentage in the project is substantially lower than in the second. As a platform for implementation of EAM-solutions, ready modeling tools are used, with significant efforts spent to study the

company work specifics, integrate solutions with various processes into companies, as well as develop documentation, coordinate and implement the solution.

Let us divide the EAM-solution into three parts, i.e. methodological unit, technology unit and maintenance and support unit. The methodological unit includes a modeling language, modeling procedure, process integration and results. The technological unit is composed of a basic tool and set of software tools and documentation. Finally, the maintenance and support unit comprises two elements for training and support. From the experience of our projects, we have concluded that such splitting is reasonable in organization and performance of EAM-projects: it seems to be more «operational» than various classifications of the composition of enterprise architecture [15, 17].

Let us turn our attention in more detail to the description of work products that are part of each unit.

A modeling language is created as part of the EAM-project, inasmuch as large companies generally have unique, specific characteristics, with the result that standard languages and notations proposed within EAM-tools need to be improved. In so doing, one uses extenders which are generally available for all EAM-tools [11]. Therefore, domain-specific modeling languages appear in this area [18]. In developing such a language, it is important to define the concepts that shall be used in modeling: with their help it is convenient for company specialists to describe the enterprise activities. For this purpose, reference models, industry standards and frameworks are often used and, in fact, explicitly or implicitly ontologies are created [1, 2]. The modeling language can be a simple slice/setting of a standard language, but a more complicated case is possible, when the mathematical model of the base language is changed, and new concepts and interconnections, new modeling viewpoints, types of charts, etc. are defined.

In order to determine the proper use of the language modeling a *procedure* is created. The main question to which it gives an answer is «how to model» («what to model» was defined when creating the language). The procedure contains modeling scenarios, detail levels and modeling focuses in development of various models, as well as sources of information for development / modification of the models.

An important work product is a running and efficient architecture management process. Among the main aspects to be covered by this process are:

✦ roles of the solution users including various rights to change information in the repository;

✦ end-to-end scenarios of using solutions – both basic and system (support of the integrity and correctness of the information repository, backup, version control, etc.);

✦ additional rules, i.e. naming objects, storage folder for charts and objects, rules to work with predefined objects, etc.

Such a process is started.

Integration is implementation of the interrelations of the architecture management process with other processes and company functions (requirements, project management and so on.). There is also a need to implement the EAM-solution integration with various company information systems to exchange data with them during architecture management. Various techniques and standards, specifically, TOGAF [24], give great attention to this work product.

An important part of the final delivery covers *modeling results* of the company architecture. This part of the delivery can be created by the project team and/ or company specialists. Meanwhile, «as is» models, i.e. a description of the current situation of the company, and «to be» model, i.e. a description of the final (desired) organization of the company are created. The creation of these models is a time-consuming activity, so you need to clearly define exactly what kind of models should be developed within the project and their details.

Selection of an appropriate *basic tool* is an important work product of the EAM-project. The EAM functionality is described many times and in detail (see, for example [11]). Many organizations, for example, Gartner [11] are involved in EAM-tools classification. These approaches are used by the majority of the business enterprises when selecting EAM-tools. However, the selection of the EAM-tool can in addition be affected by the following circumstances:

✧ if your company is already using a certain EAM-tool, then, it is reasonable to use it in the EAM-project;

✧ when selecting a foreign EAM-tool for Russian companies, it is important that the supplier has a mission/partner in the Russian Federation to provide support for the implementation and operation of the tool; otherwise, the use of such a tool has high risks.

The software environment is the most important set of work products of the EAM-solution. This environment is created by configuring/modification of the basic EAM-base tool. *Configuring* involves the use of EAM-tool settings (templates, palettes, etc.).

Modification is the creation of additional software using an open software EAM-tool interface. Let us select the following work products of the software environment:

✦ *user interfaces* intended for different user groups;

✦ *basic language support*, allowing users to work in EAM-tool in terms of the created modeling language;

✦ *integrity support* (*post-processing*) — packet auditors for diagram syntax (basically required, as the modeling language is usually hard to implement in the EAM-tool);

✦ *integrity support* (*administration*) — monitoring the correct location of individual model elements in the repository packs, rules of using predefined objects, cleaning the model from «garbage», etc.;

✦ *import/export of data* — data exchange between the EAM-tool and different information systems existing in the company;

✦ model *reports generators* — configuring the existing reports and creating new ones;

✦ *web-portal* — contains reference data on the modeling, often the modeling results. Although EAM-tools often support automatic generation of portals for the repository, many projects require the creation of a complicated and multi-functional portal (see the example in paper [10]);

✦ *repository structure (*in terms of ARIS [9] — the folders in terms of [6, 8] — classifiers); when designing the structure, not only the information structure, but also the areas of responsibility in the company are taken into account (that is, the repository shall have no folders, for which nobody is responsible);

✦ *scanning and setting up* the created EAM-solution.

Let us note that the methods and volumes of implementation of these work products are dependent on the properties of the basic EAM-tool used in the project.

All basic delivery elements must be provided with *documentation*. The main objective of the documentation is to facilitate as much as possible the involvement of company employees/contractors in the architecture management processes. The documentation may also be posted on the web-portal, contain a significant amount of visual materials, including video tutorials, quizzes to test knowledge, and more. The volume and complexity of this part of delivery can vary profoundly, becoming one of the most resource-intensive parts of the EAM-solution.

An essential EAM-solution component is *training*. The result of training is that the customer is able to use the solution all by himself.

Solution *maintenance* means assistance to users in working with the solution upon completion of the project, as well as correction of a number of errors and shortcomings. The scope of work that can be performed within the maintenance is normally stipulated in the contract.

4. Examples

Tables 1, 2 and 3 provide descriptions of EAM-projects in terms of the EAM-solution. Projects P1, P2, P3 were performed for large state-owned corporations of the Russian Federation (IT-architecture development), while project P4 was implemented for a large government organization of the Russian Federation (business architecture development). Of all the projects, only P2 was not completed successfully, although again its results (ontology and modeling language) were used by the company. The most complicated domain-specific language was created within project P1: it was developed almost «from scratch» and used the product language ARIS as an implementer. However, this language maintenance was inferior to similar maintenance in projects P2 and P3 (items 2.2 and 2.3 in *Table 2*). This resulted from the fact that the ARIS packet is considerably inferior to Mega and IBM System Architect in terms of functional capabilities of the extenders. The EAM-packet used in project P4 did not support visual modeling tools. Therefore, not only a domain-specific modeling language but an ontology (i.e., without diagrammatic notation) was created within the project, with the result that the integrity support information in this ontology was negligible. The user interface in project P3 was created based on Share Point and turned out to be the most multi-purpose of all projects (item 2.1 of *Table 2*).

Table 1.

Methodological unit

#	Final delivery elements	EAM projects			
		P1	P2	P3	P4
1	Modeling language	+ + +	+	+ +	+ +
2	Procedure	-	-	-	+ -
3	Process	+	+ -	+	+ -
4	Integration	+ -	-	+	-
5	Modeling results: «as is»	+ +	+	+	+
	Modeling results: «to be»	-	+ -	+	-

Technological unit

#	Final delivery elements	EAM projects			
		P1	P2	P3	P4
1	Basic EAM-tool	ARIS	Mega	IBM System Architect	ORG-Master (Russian EAM-package)
2	Software environment				
2.1	User interfaces	+ -	+ -	+	+
2.2	Basic support of the modeling language	+ -	+	+	+ -
2.3	Integrity support (online)	+ -	+	+	+ -
2.4	Integrity support (post-processing)	+	+	+	-
2.5	Integrity support (administration)	+	+	+	+ -
2.6	Import/export	+	+	+	-
2.7	Web-portal	+	+ +	+	+ +
2.8	Report generators	+ -	+ -	+	+ +
2.9	Repository structure	+	+	+	+
2.10	Solution scanning and setting up	+	+	+	+
3	Documentation	+	+	+	+ -

Maintenance and support unit

#	Final delivery elements	EAM projects			
		P1	P2	P3	P4
1	Training	+	-	+	+ -
2	Maintenance	+	-	+	+ -

Conclusion

This paper addresses a definition of the EAM-project and its result – an EAM-solution. Review of the EAM-project as a specific IT project made it possible to apply software engineering tools to its description. A precise identification of components of the final delivery of EAM-solution can help in development of requirements in EAM-projects and evaluation of resources necessary for their implementation. The model can be also used to assess the quality of EAM-project results. In addition, this model is applicable for further study of EAM-projects. Even a fairly superficial analysis of the four projects carried out with its help has revealed the following tendency, i.e. weak support of the modeling technique: specifically, customers do not understand the importance of this part of delivery combining the technique with the user documentation or even excluding it from the final delivery.

For a more efficient practical application of the EAM-solution final delivery model, numerical metrics should be created to evaluate its work products. Further work is also needed to deepen the parallels between the domain specific modeling in software engineering and architecture management. Generally speaking, the results of this work are a first step in application of software engineering techniques to studying projects in the area of enterprise architecture.

Furthermore, an important aspect of a more successful implementation of EAM-projects is the issue of high-quality training of company employees (within or beyond the framework of the EAM-project), the latest architecture management method, knowledge management, etc. [3, 4]. ∎

References

1. Gavrilova T.A. (2002) Logiko-lingvisticheskoe upravlenie kak vvedenie v upravlenie znanijami [Logic-linguistic methods as introduction in knowledge management]. *Novosti iskusstvennogo intellekta*, no. 6, pp. 22–27 (in Russian).

2. Gavrilova T.A., Kudrjavcev D.V., Gorovoj V.A. (2006) Modeli i metody formirovanija ontologij [Models and methods for ontology development]. *Nauchno-Tehnicheskie Vedomosti SPbGPU*, no. 46, pp. 21–28 (in Russian).

3. Gavrilova T.A., Leshcheva I.A., Kudrjavcev D.V. (2012) Ispol'zovanie modelej inzhenerii znanij dlja podgotovki specialistov v oblasti informacionnyh tehnologij [Using of knowledge engineering models for training of specialists in the field of information technologies]. *System Programming*, vol. 7, no. 1, pp. 90–105 (in Russian).

4. Gavrilova T.A., Leshcheva I.A., Strakhovich E.V. (2011) Ob ispol'zovanii vizual'nyh modelej v prepodavanii [On application of visual models in teaching]. *Herald of Saint Petersburg University. Series 8: Management*, no. 4, pp. 124–150 (in Russian).

5. Koznov D.V. (2014) *Vizual'noe modelirovanie informacionnyh e-servisov v publichnoj sfere* [Visual modeling in development of public information e-services]. St.Petersburg: SPbGU (in Russian).

6. Grigor'ev L.Ju., Koznov D.V., Kudrjavcev D.V. (2013) Obzor jazyka modelirovanija ORG-Master [OGR-Master modeling language]. *Sistemnoe programmirovanie*, vol. 8, no. 1, p. 5 (in Russian).

7. Kudrjavcev D.V., Arzumanjan M.Ju., Grigor'ev L.Ju. (2014) *Tehnologii biznes-inzhiniringa* [Business engineering technologies]. St.Petersburg: SPbGPU (in Russian).

8. Kudrjavcev D.V., Koznov D.V., Grigor'ev L.Ju. (2013) Sredstva tipizacii i pragmatika jazyka modelirovanija ORG-Master [Typing and pragmatics of ORG-Master modeling language]. *Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki*, vol. 88, no. 6, pp. 79–85 (in Russian).

9. Kamennova M., Gromov A., Ferapontov M. (2001) *Modelirovanie biznesa. Metodologija ARIS. Prakticheskoe rukovodstvo* [Business modeling: ARIS guidelines]. Moscow. Vest' – Meta Tehnologija (in Russian).

10. Kostyrko A., Kudrjavcev D., Grigor'ev L., Kislova V., Zhulin A., Sinyatullina L., Ermakov R. (2012) Modelirovanie kompleksov gorodskogo hozjajstva dlja sistemnogo razvitija IKT goroda [Modeling of municipal facilities for system development of municipal info-communication technologies]. Proceedings of the *Knowledge Engineering and Semantic Web Technologies Conference (KESW 2012), November 2-8, 2012, Saint Petersburg*, pp. 81–88 (in Russian).

11. Brand S. (2014) *Magic Quadrant for Enterprise Architecture Tools*. Gartner.

12. Buckl S., Schweda C.M. (2011) *On the state-of-the-art in Enterprise Architecture Management literature: Technical report*. Munchen: Technische Universitat Munchen.

13. *CMMI-SVC, CMMI for Services*, Version 1.3. (2010) Available at: http://www.sei.cmu.edu/reports/10tr034.pdf (accessed 27 January 2015).

14. Cox I.R. (2014) *Enterprise Architecture: How to get EA optimized*. Amazon Digital Services.

15. Federal CIO Council (2013) *Federal Enterprise Architecture Framework*. Version 2. Federal CIO Council, US.

16. Frank U. (2014) Enterprise modelling: The next steps. *Enterprise Modelling and Information Systems Architectures*, vol. 9, no. 1, pp. 22–37.

17. ISO (2000) *ISO 15704. Industrial automation systems – Requirements for enterprise-reference architectures and methodologies*. ISO.

18. Kelly S., Tolvanen J. (2008) *Domain-specific modeling: Enabling full code generation*. Wiley, IEEE Computer Society Press.

19. Kudryavtsev D., Gavrilova T. (2011) Diagrammatic knowledge modeling for managers: Ontology-based approach. Proceedings of the *International Conference on Knowledge Engineering and Ontology Development, (KEOD 2011), October 26, 2011, Paris, France*, pp. 386–389.

20. Kuhrmann M., Kalus G., Knapp A. (2013) Rapid prototyping for domain-specific languages – From stakeholder analyses to modelling Tools. *Enterprise Modelling and Information Systems Architectures*, vol. 8, no. 1, pp. 62–74.

21. Lankhorst M.M. (2005) *Enterprise Architecture at work: Modelling, communication and analysis*. Berlin, Heidelberg: Springer.

22. MEGA International (2011) *MEGA Studio User Guide*. 2nd Edition. MEGA International.

23. *Microsoft Solutions Framework, Process Model*, Version 3.1. (2002) Microsoft. Available at: http://www.microsoft.com/msf (accessed 28 January 2015).

24. TOGAF, Version 9.1. Enterprise Edition. (2011) Available at: http://www.opengroup.org/togaf/ (accessed 25 January 2015).

25. Franke U., Ekstedt M., Lagerstrom R., et al. (2010) Trends in Enterprise Architecture practice – A survey. Proceedings of *5th International Workshop (TEAR 2010), November 12, 2010, Delft, The Netherlands*, pp. 16–29.

ОСОБЕННОСТИ ПРОЕКТОВ В ОБЛАСТИ РАЗРАБОТКИ КОРПОРАТИВНОЙ АРХИТЕКТУРЫ ПРЕДПРИЯТИЙ

Д.В. КОЗНОВ
кандидат физико-математических наук, доцент кафедры системного программирования, Санкт-Петербургский государственный университет
Адрес: 198504, г. Санкт-Петербург, Старый Петергоф, Университетский проспект, д. 28
E-mail: st008149@spbu.ru

М.Ю. АРЗУМАНЯН
ассистент кафедры информационных технологий в экономике, Санкт-Петербургский государственный университет телекоммуникаций им. М.А.Бонч-Бруевича; ООО «Смарт Архитектс»
Адрес: 193232, Санкт-Петербург, проспект Большевиков д. 22, корп. 1
E-mail: maxim.arzumanyan@gmail.com

Ю.В. ОРЛОВ
генеральный директор ООО «Смарт Архитектс»
Адрес: 191186, г. Санкт-Петербург, наб. реки Мойки, 61А
E-mail: yury.orlov@smartarchitects.ru

М.А. ДЕРЕВЯНКО
инженер кафедры информационных технологий в экономике, Санкт-Петербургский государственный университет телекоммуникаций им. М.А.Бонч-Бруевича; ООО «Смарт Архитектс»
Адрес: 193232, Санкт-Петербург, проспект Большевиков д. 22, корп. 1
E-mail: maxim.derevyanko@gmail.com

К.Ю. РОМАНОВСКИЙ
кандидат физико-математических наук, старший преподаватель кафедры системного программирования, Санкт-Петербургский государственный университет
Адрес: 198504, г. Санкт-Петербург, Старый Петергоф, Университетский проспект, д. 28
E-mail: k.romanovsky@spbu.ru

А.А. СИДОРИНА
студентка кафедры информатики, Санкт-Петербургский государственный университет
Адрес: 198504, г. Санкт-Петербург, Старый Петергоф, Университетский проспект, д. 28
E-mail: sidorina-nastia@mail.ru

Управление архитектурой предприятия (Enterprise Architecture Management, EAM) является ключевым видом деятельности при повышении эффективности бизнеса в современных компаниях. Данная деятельность поддерживается специальными программными продуктами (так называемыми EAM-инструментами) — ARIS, Mega и др. Внедрение процесса управления архитектурой, как правило, происходит путем реализации соответствующих проектов. Данная статья посвящена исследованию таких проектов. Среди них выделяются так называемые EAM-проекты, которые являются специфическими ИТ-проектами и направлены на создание и развертку в компании ИТ-решения по управлению архитектурой на основе какого-либо EAM-инструмента. Для дальнейшего исследования EAM-проектов использованы методы программной инженерии (Software Engineering) — методология разработки ПО Microsoft Solution Framework (MSF) и терминология стандарта CMMI, — а также концепция предметно-ориентированного моделирования (Domain-Specific Modeling, DSM).

В работе детально описана финальная поставка типового ЕАМ-проекта, подразделяющаяся на методологический блок (язык моделирования, методика, процесс, интеграция, результаты моделирования), технологический блок (базовый инструмент, комплекс программных средств, документация), сопровождение и поддержку (обучение, сопровождение). Обсуждаются особенности разработки этих рабочих продуктов. При этом детально анализируется программная часть поставки — пользовательские интерфейсы, базовая поддержка языка моделирования, поддержка целостности (online), поддержка целостности (post-processing), поддержка целостности (administration), импорт/экспорт, Web-портал, генераторы отчетов, структура репозитория, развертка и настройка решения. В рамках введенных понятий обсуждаются четыре ЕАМ-проекта по реализации управления архитектурой, реализованные в российских компаниях: три проекта были выполнены для различных госкорпораций, один — для крупного органа государственной власти. Выявлена типовая сложность этих проектов — недостаточная поддержка методики моделирования.

Литература

1. Гаврилова Т.А. Логико-лингвистическое управление как введение в управление знаниями // Новости искусственного интеллекта. 2002. № 6. С. 22—27.
2. Гаврилова Т.А., Кудрявцев Д.В., Горовой В.А. Модели и методы формирования онтологий // Научно-технические ведомости СПбГПУ. 2006. № 6. С. 21—28.
3. Гаврилова Т.А., Лещева И.А., Кудрявцев Д.В. Использование моделей инженерии знаний для подготовки специалистов в области информационных технологий. Системное программирование. 2012. Т. 7, № 1. С. 90—105.
4. Гаврилова Т.А., Лещева И.А., Страхович Э.В. Об использовании визуальных моделей в преподавании. Вестник Санкт-Петербургского университета. Серия 8: Менеджмент. 2011. № 4. С. 124—150.
5. Кознов Д.В. Визуальное моделирование информационных е-сервисов в публичной сфере. СПб.: СПбГУ, 2014. 144 с.
6. Григорьев Л.Ю., Кознов Д.В., Кудрявцев Д.В. Обзор языка моделирования ОРГ-Мастер // Системное программирование. 2013. Т. 8, № 1. С. 5.
7. Кудрявцев Д.В., Арзуманян М.Ю., Григорьев Л.Ю. Технологии бизнес-инжиниринга. СПб.: СПбГПУ, 2014. 427 с.
8. Кудрявцев Д.В., Кознов Д.В., Григорьев Л.Ю. Средства типизации и прагматика языка моделирования ОРГ-Мастер // Научно-технический вестник информационных технологий, механики и оптики. 2013. Т. 88, № 6. С. 79—85.
9. Моделирование бизнеса. Методология ARIS. Практическое руководство / М.С. Каменнова и [др.]. М.: Весть — Мета Технология, 2001. 333 с.
10. Моделирование комплексов городского хозяйства для системного развития ИКТ города / А. Костырко и [др.] // Сб. трудов конференции «Инженерия знаний и технологии семантического веба» (KESW 2012), 2-8 ноября 2012 г., г. Санкт-Петербург. С. 81—88.
11. Brand S. Magic Quadrant for Enterprise Architecture Tools. G00262. Gartner, 2014. 38 p.
12. Buckl S., Schweda C.M. On the state-of-the-art in Enterprise Architecture Management literature: Technical report. Munchen: Technische Universitat Munchen, 2011. 118 p.
13. CMMI-SVC, CMMI for Services, Version 1.3. 2010. 520 p.
14. Cox I.R. Enterprise Architecture: How to get EA optimized. Amazon Digital Services, 2014. 48 p.
15. FEAF-II. Federal Enterprise Architecture Framework. Version 2. Federal CIO Council, US. 2013. 434 p.
16. Frank U. Enterprise modelling: The next steps // Enterprise Modelling and Information Systems Architectures. 2014. Vol. 9, No. 1. P. 22—37.
17. ISO 15704. Industrial automation systems — Requirements for enterprise-reference architectures and methodologies. 2000. 7 p.
18. Kelly S., Tolvanen J. Domain-specific modeling: Enabling full code generation. Wiley, IEEE Computer Society Press, 2008. 444 p.
19. Kudryavtsev D., Gavrilova T. Diagrammatic knowledge modeling for managers: Ontology-based approach // Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2011), October 26, 2011, Paris, France. P. 386—389.
20. Kuhrmann M., Kalus G., Knapp A. Rapid prototyping for domain-specific languages — From stakeholder analyses to modelling tools // Enterprise Modelling and Information Systems Architectures. 2013. Vol. 8, No. 1. P. 62—74.
21. Lankhorst M.M. Enterprise Architecture at work: Modelling, communication and analysis. Berlin, Heidelberg: Springer, 2005. 338 p.
22. MEGA Studio User Guide. 2nd Edition. MEGA International, 2011. 114 p.
23. Microsoft Solutions Framework, Process Model, Version 3.1 // Microsoft 2002. [Электронный ресурс]: http://www.microsoft.com/msf (дата обращения 28.01.2015).
24. TOGAF, Version 9.1. Enterprise Edition. 2011. 692 p.
25. Trends in Enterprise Architecture practice — A survey / U. Franke [et al.] // Trends in Enterprise Architecture Research. Proceedings of 5th International Workshop (TEAR 2010), November 12, 2010, Delft, The Netherlands. P. 16—29.