Evaluation of Application Architecture Change Cases: Building Blocks Reusability Assessment Method

Rūta Pirta(✉) and Jānis Grabis

Department of Management Information Technology, Riga Technical University,
Kalku 1, Riga LV-1658, Latvia
{ruta.pirta,grabis}@rtu.lv

Abstract. The reuse of IT solutions plays an important role, as it enables organizations to develop services more quickly and at reduced cost, and promotes greater interoperability, standardization and cooperation. Organizations increasingly include the reuse principle in their Enterprise architecture (EA) development vision. However, they often lack methods for comprehensive evaluation of changes in Application Architecture (AA) including assessment of reuse of AA components. In this paper, we outline a method for AA change evaluation. The objective of the proposed method is to provide support for establishing a controlled environment for AA change implementation planning to meet defined EA principles. In this paper, we focus on the reuse principle, however the approach can be also adapted for change assessment with regards to other principles such as centralization and standardization.

Keywords: Enterprise architecture · Application Architecture · Architecture principles · IS changes · Reuse

1 Introduction

Enterprise architecture (EA) is a commonly accepted instrument to guide enterprise transformations. Its main goal is to enhance and maintain the mutual alignment of business and IT [2, 3]. EA vision is frequently led by architecture principles that support decision making across enterprise [4]. Nowadays large-scale organizations and public administration bodies more frequently include reusability principle in their EA development vision [5–7]. The reuse is not a goal by itself but is mainly motivates by such business goals as increase of productivity, reduction of development effort, time and cost, improvement of quality and increase of interoperability [8]. Thus, the principle supports enterprises in information technology (IT) related decisions making that can add value to the business and improve business and IT alignment.

However, the strategic level pursuit is often neglected as the operational level and individual units tend to prefer to build their own IT solution rather than reuse existing solutions. In many cases, that means greater autonomy but can also contribute to an expensive and fragmented EA, which often duplicates IT solutions and impedes the sharing and reuse of software and IS services [1]. The organizations with decentralized

W. Abramowicz (Ed.): BIS 2017 Workshops, LNBIP 303, pp. 150–162, 2017.
https://doi.org/10.1007/978-3-319-69023-0_14

IT governance often faces "Not-invented-here syndrome" - software engineers prefer to re-write components as they believe that they can improve the reusable component and writing original software is seen as more challenging than reusing existing one [8]. Not all Application Architecture (AA) components are reusable. Solutions might have dependencies or reuse constraints. These dependencies can either be of a technical nature (e.g. reliance on a specific third party product) or relative to specific legislation or business domain [6].

Empirical observations show that enterprises lack a comprehensive approach for evaluation of proposed changes in AA with regards to facilitating reusability. Decisions about changes in AA often are subjective and require extensive manual work (IT solutions technical documentation analysis, solutions audits etc.) [6].

In this paper we propose a method for evaluation of changes in AA according to EA development principles. This method generates recommendations for implementing changes in AA in a way to attain the best alignment with defined EA principles. In the paper we focus on the reuse principle, however the approach also can be adapted for assessment of compliance with other principles, for example, adoption of centralized IT management and preference for standardized solutions.

The proposed method is envisioned as a middle-ground between isolated evaluation of the IS change requests and comprehensive strategic level planning of EA evolution. It provides a tactical level tool helping organizations to understand implications of their IS change requests. The method is intended for large-scale organizations, including international corporations, concerns with several legal entities, as well as for public administration bodies. It focuses on significant changes in the AA that can be classified as incremental changes and re-architecting changes according to the TOGAF classification in [9].

The rest of the paper is organized as follows. Section 2 states the background and briefly outlines the related works. In the Sect. 3 the approach's outline is given. Section 4 describes an illustrative example. The paper closes in Sect. 5 with the conclusions.

2 Foundations

2.1 Problem Statement

Given EA, architecture principles and change case (CC) raised by user of enterprise information systems, the objective is to find the most suitable solution for implementing the requested architecturally significant changes without starting a new architecture development cycle. The most suitable solution is provided as a set of recommendations indicating which elements of the EA should be altered.

The EA consists of at least three views [9], namely, AA, information architecture (IA) and business architecture (BA). Each view consists of specific elements (Fig. 1). Logical application components represent application functionality independent of implementation. In this paper, logical application components are referred as Information Systems (IS). IS services represent automated elements of business service and are implemented using application components. Data entities encapsulate data that is recognized by a business domain expert as a discrete concept. They are processed by logical application components, are accessed and updated by business services and supported or consumed by actors. Business services support business capabilities through an explicitly defined interface and is explicitly governed by an organization. They are realized through business processes and consumed by actors.

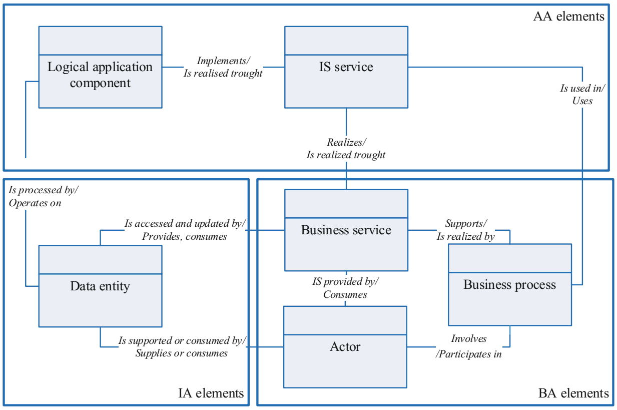

Fig. 1. Views and elements of EA used in evalaution of change cases

A CC defines a need for adjustments in enterprise business processes or information technologies in a semi-structured manner. It is assumed that the CC contains information allowing to identify IS used by an actor raising the CC as well as requested new or existing IS service can be implied by an expert from the description provided.

EA principles are statements of intent or purpose that support business needs and changing customer desires, they guide business & IT decisions and investments [10]. The principles in enterprises are chosen so as to ensure alignment of IT strategies with business strategies and visions [9], and well-formulated principles are measurable [7]. This paper focuses on the reuse principle and reuse efficiency is measured as total amount of reused EA components.

Evaluation of changes could be aided by consulting a domain specific reference architecture. The reference architecture encompasses domain specific knowledge and provides a template for accelerating and improving the EA development process [13].

2.2 Related Work

The need for more standardized products promotes an explicit (reusable) architecture and reuse of components [14]. Software reuse is: “the process whereby an organization defines a set of systematic operating procedures to specify, produce, classify, retrieve, and adapt software artefacts for the purpose of using them in its development activities.” [15].

The reuse principle is currently widely used in public administration bodies and large enterprises. The reuse enables organizations to develop services more quickly and at a reduced cost, and promotes greater interoperability, standardization and cooperation [1]. Two distinctive directions of research are exploration of reusability benefits [14–17] and software reusability assessment [6, 20, 21].

For example, a case study at a large telecom company provides evidence of quality benefits of large-scale reuse programs [14]. Other key benefits of software reuse are increased productivity, lower fault/defect-density, lower number of changes per module or per LOC, reduced development and maintenance effort, reduced complexity, and consistency of applications and the software architecture [16, 17]. These articles investigate benefits of reuse but they do not address planning for reusability.

A dependency model is used to evaluate reusability of open source components from both static and dynamic perspectives [21]; and three metrics based on the model to measure interaction behavior complexity between component and its context. For performance evaluation, the authors suggest using performance measurement model. [21] specifically addresses software source code reuse topic. The paper presents extraction and analysis methods for developers' source code reuse behavior. In [22], a method for aspect oriented software reusability assessment using inheritance metrics is proposed. The authors state that it is not possible directly measure reusability from the design of software, and propose to develop a quality model to quantitatively assess the reuse of components. These papers mainly focus on software elements (source code and others) technical characteristics evaluation and do not include architectural considerations.

A reuse reference grid is proposed as an assessment framework to help categorize and assess the cost/benefit of the current level of reuse as a prelude to considering future reuse opportunities [23]. It facilitates reuse assessment in three ways: (1) categorizing existing reuse, (2) assessing current reuse levels, and 3) considering future reuse strategy.

European Commission [6] proposed a template for a factsheet that would facilitate the assessment of solutions reuse by providing useful and detailed information that should be considered when evaluating reuse of a solution in a specific context. The template focuses on reusability of technical solutions both as a software component and as a service. However, the template does not support enterprise architects in the evaluation process. Template's usage requires architects to have a good understanding of the solution that would benefit from reuse.

3 Change Evaluation Method

The proposed method (Fig. 2) supports assessment of reuse potential of existing AA components, namely, architecture building blocks (ABB) and solutions building blocks (SBB) in the case of implementing architecturally significant change requests. A research method to be taken to address research method and practical challenges follows the nested design science problem solving approach [24].

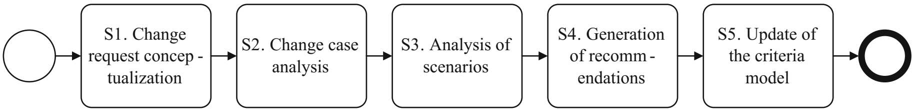
S1. Change request concep - tualization
S2. Change case analysis
S3. Analysis of scenarios
S4. Generation of recomm - endations
S5. Update of the criteria model

Fig. 2. Phases of the change evaluation method

The reuse principle is enshrined in the organization's EA vision. The method focuses on significant changes, which can be classified as incremental changes and re-architecting changes according to the TOGAF classification in [9]. Simplification changes are outside the scope, as they do not have a significant impact on EA and they can normally be handled via change management techniques. The types of reuse [11] considered in this paper are function reuse, component reuse and full model reuse.

3.1 Change Case Analysis (S1)

The evaluation process is initiated by receiving a CC containing information as defined in Sect. 2. The CC contains both structured and unstructured data and implies what IS services are pertinent to the request. These IS services are identified by an enterprise architect. The requested IS services can be delivered by introducing new or reusing existing EA components. The enterprise architect also links IS services to supported Business services (BS) and Business processes (BP), what facilitates identification of IS services in the reference model. The result of the CC analysis is a list with requested IS services and related BS and BP, which are referred as elements mentioned in the change request and are denoted as ISS*, BS* and BP*, respectively.

3.2 Development of Change Implementation Architectural Scenarios (S2)

A change implementation scenario defines one of alternatives for updating EA in response to the change request. It specifies ABB and SBB to be considered to implement the change. In this case, ABB are IS and SBB are IS services.

Change implementation scenarios are dependent of changes implementation type and level of components reuse (Table 1). Possible implementation types are new IS implementation or existing IS modification. Possible levels of IS services reuse are:

- The functionality provided by new IS services;
- The functionality provides new and reused IS services;
- The functionality provides reused IS services.

Table 1. Levels of reuse

Implementation type	Level of reuse		
	New IS services	New & reused IS services	Reused IS services
Existing IS modification	X	X	X
New IS implementation	X	X	

Combining implementation alternatives in ABB level and different levels of components reuse in SBB level are defined change implementation scenarios.

Change implementation scenarios are defined in two successive steps – initially ABB alternatives are set, then SBB alternatives are identified and combined with defined ABB, jointly creating change implementation architecture scenarios.

Identification of ABB

Analysis the change request, existing EA model and reference model is performed to identify ABB. Firstly, IS to be modified and/or IS to be implemented are inferred by analyzing the change request and existing EA model. The inference rules are aimed at narrowing a set of potential modification and implementation alternatives. Given ISS*, BS* and BP*, the inference rules analyze relationships in the EA model and selects candidate IS services for modification/implementation. The selection rules are: (1) select all IS mentioned in the change request; (2) select all IS supporting BP*; (3) select all IS supporting BS*; (4) select all IS maintaining data that is accessed or modified by BS*; and (5) select all IS maintaining data linked to the actor requesting changes. The selected IS are collectively denoted as IS'.

Secondly, existing EA and reference model gap analysis is performed. During the analysis common IS supporting BP* and BS* are identified and selected as potentially modifiable AA components. As well as gaps are identified (IS that exists in reference models but does not exists in existing EA model) and selected as potentially new AA components.

The analysis yields candidate ABB for implementation of each ISS*. The architect manually accepts or rejects each candidate ABB.

Identification of SBB

Analysis of the existing EA model and reference model is performed to identify candidate SBB what are linked to the candidate ABB. ISS* are mapped to the existing EA model and reference model as shown in Fig. 3. If an appropriate service is available its reuse is recommended. Otherwise mapping to the reference model is performed resulting in recommendations to implement a new IS service, to use a similar service and to review the service. The service review is recommended if similar services are not available in the reference architecture what suggests a need reconsider architecture design.

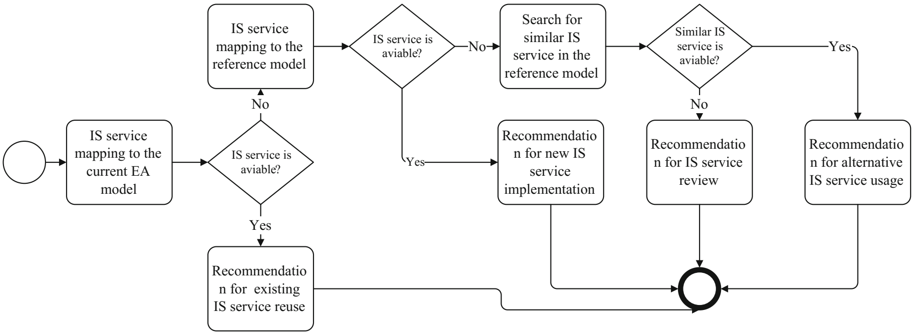
IS service mapping to the current EA model
IS service is aviable?
No
Yes
IS service mapping to the reference model
IS service is aviable?
No
Yes
Search for similar IS service in the reference model
Similar IS service is aviable?
No
Yes
Recommendation for existing IS service reuse
Recommendation for new IS service implementation
Recommendation for IS service review
Recommendation for alternative IS service usage

Fig. 3. SBB analysis process

3.3 Scenarios Analysis (S3)

Scenarios analysis consists of: (1) assessment of ABB alternatives; (2) assessment of SBB alternatives and (3) evaluation of ABB and SBB interoperability.

Technical criteria are used for assessment of candidate SBB and ABB, as well as for evaluation of ABB and SBB interoperability. These criteria are directly related to defined EA principles. They include specific ABB and SBB reusability evaluation criteria as proposed in [6]. The examples of ABB and SBB interoperability criteria are use of standards and availability of well-defined application programming interfaces. The standards used by a solution could either make it a good reuse candidate for a specific solution or exclude it due to conflicting choices already made.

The technical criteria are calculated from the EA model to analyze scenarios.

3.4 Recommendations Generation (S4) and Criteria Model Update (S5)

During the recommendation phase, the most appropriate implementation scenario is selected. Values of the selected technical criteria are calculated for every scenario. The tentative landscape having the best values of criteria is selected. It is recommended as the most appropriate way of implementing the change request.

To allocate resources efficiently, after the existing components evaluation, financial calculations must be performed (Total Cost of Ownership vs Total Cost of Reuse) to compare existing components reuse vs new components implementation alternatives.

The process ends with criteria model update. After several rounds of change implementation in the EA model, the enterprise architect updates the criteria model with empirical knowledges about changes implementation practice.

4 Illustrative Example

An example is based on the real-life case in a government body (further referred as GOVb). The organization runs centralized IT service that serves more than 13 partly autonomous departments. The organization has complex AA architecture that consists of more than 30 partly integrated IS supporting delivery of public services as well as internal administrative processes. Several IS are used for IT services management support, including, service desk system, budgeting IS, users management IS, documents management IS and a centralized network and IS management software. Collaborative tools are also used, e.g., e-mail, internal chat and documents sharing environment. GOVb uses intranet portal for internal information flow.

GOVb has started a transition from traditional ICT delivery model to service-oriented. To implement the change, GOVb started to design a "to-be" IT service delivery model. The model defines several new processes, including processes for centralized ICT change management. Currently the service desk system and the centralized network and IS management software support the "as-is" change management processes in a fragmented manner. The service desk system's functionality party duplicates the centralized network and IS management software, both IS include several similar IS services such as ICT tickets management (incidents, problems, changes). The service desk system manages user tickets, while the centralized network and IS management software are used to report work performed by ICT personnel and ICT items configuration management.

Several AA CC were identified during implementation of the centralized and standardized ICT change management process. The enterprise architect reviews the CC and identifies included IS services and links them to supported BS and BP (Table 2) as defined in Sect. 3.

Table 2. Identified IS services and linked BP (all BP are associated with ICT change management BS)

ID	IS service	BP
1.	ICT change request data automatic processing and importing	ICT change request registration
2.	ICT change request data fill	ICT change request registration
3.	ICT change request workflow management	ICT change request registration; ICT change request approval; ICT changes planning; ICT changes approval; ICT changes control; ICT changes review and closing
4.	ICT change request status changes management (registered, approved, closed etc.)	ICT change request registration; ICT change request approval; ICT changes planning; ICT changes approval; ICT changes control; ICT changes review and closing
5.	ICT change request classification	ICT change request approval
6.	Notifications management (notifications about status changes etc.)	ICT change request registration; ICT change request approval; ICT changes planning; ICT changes approval; ICT changes control; ICT changes review and closing
7.	ICT change project documentation management	ICT changes planning; ICT changes approval; ICT changes control; ICT changes review and closing
8.	ICT items data update	ICT changes review and closing

IS to be modified are identified by analyzing the existing EA model and reference model. The inference rules suggest that the change might relate to the following existing IS: (1) Intranet portal (because IS currently supports related BS); (2) Service desk system (because IS currently supports related BS); (3) Users management IS (IS maintaining data accessed or modified by BS); (4) Documents management IS (IS maintaining data accessed or modified by BS); (5) Centralized network and IS management software (IS that currently is used to support BS); and (6) Documents sharing environment (IS maintaining data that is accessed or modified by BS).

IS to be implemented are identified by performing gap analysis between the existing EA and reference model. For the change evaluation, the ITIL ARIS domain-specific reference model is chosen. IS that exist in the reference model but do not exist in the EA model are identified: (1) ICT asset management system (IS that currently is used to support BS); (2) ICT services catalogue (IS maintaining data that is accessed or modified by BS); (3) Configuration database (IS maintaining data that is accessed or modified by BS).

Table 3. Selected candidate ABB (a fragment)

ID	IS service	Candidate ABB	
		Existing IS modification	New IS implementation
1	ICT change request data automatic processing and importing to the change request	Intranet portal	–
		Service desk system	–
		Centralized network and IS management software	–
		Documents management system	–
2	ICT change request data fill	Intranet portal	–
		Service desk system	–
		Centralized network and IS management software	–
		Documents management system	–
3	ICT change request workflow management	Documents management system	–
		Service desk system	–
		Centralized network and IS management software	–
7	ICT change project documentation management	Documents management system	–
		Documents sharing environment	–
8	ICT items data update	Centralized network and IS management software	–
		–	ICT asset management system
		–	Configuration database

The enterprise architect reviews and accepts or rejects candidate ABB. The accepted candidate ABB are listed in Table 3.

Candidate SBB are identified by mapping the candidate IS services to existing EA. Candidate SBB for implementation of new IS services are identified by mapping of the candidate IS services to the ITIL ARIS reference model. They are ICT change request classification, ICT change project documentation management and ICT items data update.

The SBB level alternatives are combined with the ABB level alternatives and several changes implementation scenarios are proposed. These scenarios are evaluated according to GOVb's architecture principles, which include reusability and interoperability. For each scenario its rating is calculated, using measures appropriate for the candidate ABB and SBB. Examples of the criteria used to evaluate the candidate ABB are given Table 4. The criteria evaluation metrics is in the range between 0 and 5 with boundary values defined in the table.

Table 4. Criteria for evaluation of candidate ABB

#	Criteria	Criteria description [6]	Criteria evaluation metrics
1	Interfaces (interoperability principle)	Information on the architecture of the component and any expectations that the component would have of the overall solution (e.g., configurability, API)	0 – ABB is a stand-alone system 5 – well-defined open interfaces
2	Actual reuse (reuse principle)	Cases where the solution has already been reused, either as a service or as a software component. The extent to which a solution is already reused is a good indication of its maturity and reusability, both in technical terms but also potentially in terms of policy domains.	0 – ABB has not been reused 5 – maximum level of reuse across all ABB
3	Modularity (reuse principle)	Measures taken to ensure that this solution has been built as reusable from the ground up (e.g., non-proprietary technologies, modular software architecture, SOA)	0 – homogeneous ABB 5 – fully modularized ABB
4	Maturity (reuse principle)	The status of the solution in terms of maturity, to indicate its ongoing development and design stability. The solution's development status is expressed using the following values: (1) Plan, (2) Design, (3) Development, (4) Integration and testing, (5) Deployment, (6) Operation, (7) Disposal.	0 – disposed ABB 5 – implemented and used in BS and BP
5	Extensibility (reuse principle)	Information regarding the possibility to extend of modify the component to suit a specific solution's needs.	0 – ABB has not been extended 5 – maximum level of extension across all ABB
6	Use of standards (interoperability principle)	The standards that the solution uses and conforms to ranging from business and data standards to technical and communication standards.	0 – only proprietary solutions are used 5 – only standard solutions are used

Selected criteria are measured by analyzing the EA model. For example, for ABB actual reuse criteria links between ABB logical application components and different BS are analyzed – if ABB supports at least 2 unlinked BS that processes separate data entities, it is assumed that the component is reused. Based on the analysis results for each candidate ABB and scenario their ratings are calculate according to each criterion. An example of ratings for candidate ABB is shown in the Fig. 4.

As the result, one scenario (Table 5) is recommended for implementation in alignment with the AA evolution vison. The recommended scenario has the best overall rating, which is based on ratings in several measurement positions.

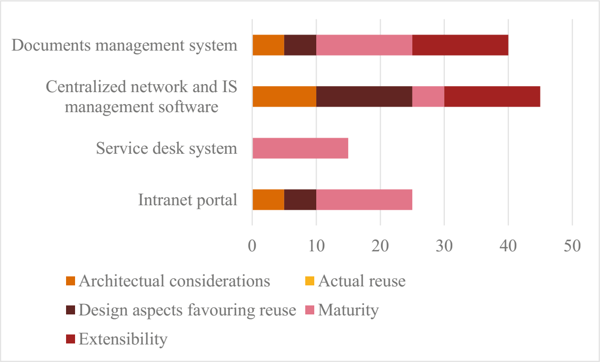

Fig. 4. Rating of candidate ABB

The implementation scenario suggests that the IS service for ICT change request data automatic processing and importing can be developed by modifying the existing Centralized network and IS management software (ABB level) where existing IS service can be reused (SBB level). The ICT change request classification IS service also can be implemented by modifying the same existing IS through development of a new IS service is required at the SBB level. The former case exemplifies high level of reuse while the latter case has a moderate level of reuse. There is no reuse in the case of IS service for ICT items data update that requires implementation of a completely new IS.

Table 5. Recommended change implementation scenario

ID	IS service	ABB		SBB	
		Existing IS modification	New IS implementation	Reused IS service	New IS service
1	ICT change request data automatic processing and importing to the change request	Centralized network and IS management software		X	
2	ICT change request data fill	Centralized network and IS management software		X	
3	ICT change request workflow management	Centralized network and IS management software		X	
7	ICT change project documentation management	Documents sharing environment			X
8	ICT items data update		ICT asset management system		X

5 Conclusions

This paper outlines a method for identification and evaluation of scenarios for modifying AA to accommodate CC raised by users of enterprise IS. The method focuses on promoting reuse of existing architectural and solution level building blocks. It combines expert judgment with quantitative and network analysis of EA models. It uses network analysis to identify candidate building blocks for implementing the change and multi-criteria quantitative analysis to evaluate the candidates. The paper presents preliminary rules for network analysis and measurable criteria for evaluation of the candidates. These set of rules and criteria are indented as extensible and are subject of further refinement and validation.

The main benefits of the method are the following: (1) the method helps aligning IS changes with the EA evolution strategy, thus facilitating business and IT alignment as well; (2) more transparent decision-taking process and (3) reduced need of expert involvment.

There are several limitations to be addressed on the future research: ensuring completeness of CCs; accounting for differences in the level of details in EA and reference models; and extending the analysis beyond evaluation of the AA components.

References

1. European Commission: The Sharing and Reuse IT Solutions Framework, Fostering collaboration among Public Administrations (2015). https://joinup.ec.europa.eu/sites/default/files/sharing_and_reuse_of_it_solutions_framework_final.pdf
2. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations. IBM Syst. J. **32**(1), 472–484 (1993)
3. Luftman, J.N., Lewis, P.R., Oldach, S.H.: Transforming the enterprise: the alignment of business and information technology strategies. IBM Syst. J. **32**(1), 198–221 (1993)
4. Urbaczewski, L., Mrdalj, S.: A comparison of enterprise architecture frameworks. Issues Inf. Syst. **7**(2), 18–23 (2006)
5. Guimarães, Th.S.M.: 21 principles of enterprise architecture for the financial sector (2012). http://www.ibm.com/developerworks/rational/library/enterprise-architecture-financial-sector/index.html?ca=drs
6. European Commission: Reusability Factsheet Template (2015). https://joinup.ec.europa.eu/sites/default/files/sc73_d02.03_reusability_factsheet_v4.00.pdf
7. Greefhorst, D., Proper, E.: Architecture Principles: The Cornerstones of Enterprise Architecture. Springer, Heidelberg (2011). doi:10.1007/978-3-642-20279-7
8. Sommerville, I.: Software Engineering, 9th edn. Addison-Wesley, Boston (2011). pp. 429–452
9. The Open Group: TOGAF, Version 9, Personal PDF Edition (2012). http://www.kingdee.com/news/subject/10togaf/pdf/TOGAF_Manual_G091.pdf
10. PricewaterhouseCoopers: Enterprise Architecture Training materials (2009)
11. Reese, R., Wyatt, D.L.: Software reuse and simulation. In: Proceedings of the 19th Conference on Winter Simulation. ACM, Atlanta, Georgia, United States, pp. 185–192 (1987)
12. Op’t Land, M., Proper, H.A.: Impact of Principles on Enterprise Engineering, In: ECIS 2007 Proceedings, p. 113 (2007)

13. Noran, O.: Using reference models in enterprise architecture: an example. In: Fettke, P., Loos, P. (eds), Reference Modeling for Business Systems Analysis, Idea Group, Hershey, USA, pp. 141–165 (2006)
14. Mohagheghi, P., Conradi, R.: An empirical investigation of software reuse benefits in a large telecom product. ACM Trans. Softw. Eng. Methodol. (TOSEM), 17(3), Article no. 13 (2008)
15. Mili, H., Mili, A., Yacoub, S., Addy, E.: Reuse Based Software Engineering Techniques Organizations and Measurement. Wiley, Hoboken (2001)
16. Selby, W.: Enabling reuse-based software development of large-scale systems. IEEE Trans. Softw. Eng. **31**(6), 495–510 (2005)
17. Mili, A., Chmiel, S.F., Gottumkkala, R., Zhang, L.: An integrated cost model for software reuse. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000), pp. 157–166 (2000)
18. Pirta, R., Grabis, J.: Integrated methodology for information systems (IS) change control based on enterprise architecture (EA) models. Inf. Technol. Manag. Sci. **18**(1), 103–108 (2015)
19. Zhang, W., Jarzabek, S.: Reuse without compromising performance: industrial experience from RPG software product line for mobile devices. In: Obbink, H., Pohl, K. (eds.) SPLC 2005. LNCS, vol. 3714, pp. 57–69. Springer, Heidelberg (2005). doi:10.1007/11554844_7
20. Wu, J., Liu, Y.P., Jia, X.X., Liu, C.: Mining open source component behavior and performance for reuse evaluation. In: The 9th International Conference for Young Computer Scientists, Hunan, pp. 1241–1247 (2008)
21. Ohta, T., Murakami, H., Igaki, H., Higo, Y., Kusumoto, S.: Source code reuse evaluation by using real/potential copy and paste. In: 2015 IEEE 9th International Workshop on Software Clones (IWSC), Montreal, QC, pp. 33–39 (2015)
22. Vinobha, A., Senthil Velan, S., Babu, C.: Evaluation of reusability in aspect oriented software using inheritance metrics. In: IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, pp. 1715–1722 (2014)
23. Waguespack, L.J., Schiano, W.T.: A reuse reference grid for strategic reuse goals assessment. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS 2006), pp. 228a (2006)
24. Wieringa, R.: Design science as nested problem solving. In: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology. Pennsylvania (2009)