Agile Software Engineering Methodology for Information Systems' Integration Projects

Kuldar Taveter[1], Alexander Norta[1]

[1] Department of Software Science, Tallinn University of Technology, Akadeemia tee 15a, 12618 Tallinn, Estonia
kuldar.taveter@ttu.ee, alex.norta.phd@ieee.org

Abstract. In this paper, first the notions are defined that are important for running integration projects – system of systems and sociotechnical system – and it is then argued that integrated systems should be treated as sociotechnical systems of systems. This is followed by defining the conceptual framework – viewpoint framework – required for agile engineering of such systems. Based on the viewpoint framework, the agile software engineering methodology for engineering sociotechnical systems of systems is then defined, proceeding by different viewpoint aspects.

Keywords: System of systems, sociotechnical system, agile software engineering, integration project

1 Introduction

Many software engineering projects require the integration of existing independent information systems (IS) rather than developing new systems from scratch. Integration of different IS results in systems of systems. *System of systems* is a collection of task-oriented or dedicated systems that pool their resources and capabilities together to create a new, more complex system that offers more functionality and performance than simply the sum of the constituent systems [2]. System of systems are large-scale concurrent and distributed systems the components of which are complex systems themselves [3]. Engineering of systems of systems is a complex task because [11]: (i) system of systems has no single owner or controller; (ii) decisions within organisations running the integrated IS are driven by political motives rather than technical considerations; (iii) there is no clearly defined and fixed system boundary. Therefore, according to Sommerville et al [11], one of the research questions that still defines the agenda of software engineering is "How can we support the agile engineering of systems of systems?" This question is also a research question for this paper.

To reflect the fact that stakeholders of IS – their owners, designers, and users – form evolving and interacting communities that include technical, human and organisational elements, IS should be treated as sociotechnical systems [11]. A *sociotechnical system* is a software-intensive system that has defined operational processes followed by hu-

man operators and that operates within an organization [1]. Alternatively, a sociotechnical system may be defined as a system that contains both a social aspect, which may be a subsystem, and a technical aspect [1].

In the past, we have addressed information systems' integration projects in [12], where we proposed a methodology for modelling inter-enterprise business processes. Later, we have elaborated the methodology in [13], [6], and [14], using case studies from the area of business-to-business electronic commerce.

In this paper, we use as the case study the "Once-Only" Principle Project (TOOP) funded by the Horizon 2020 program of the European Union (http://www.toop.eu/). The TOOP project has the ambition to connect 60 registries and information systems from 22 countries. This new large-scale integration project is concerned with the interchange of company data between the registries and information systems. We view a registry as a special kind of information system. Normally, information about a company is stored in the business registry of the country where the company is registered. However, the same information or a part of it is also stored in registries managed by public administrations and other stakeholders in other countries. Keeping this information up to date is a real challenge, especially when it is concerned with the companies that have daughter companies in other countries.

In this paper, we describe the agile software engineering methodology for information systems' integration projects that has been devised for the TOOP project and other similar large-scale integration projects. The Agile Integration Methodology (AIM) views different information systems and registries as sociotechnical systems to be integrated, resulting in a complex *sociotechnical system of systems*. No off-the-shelf agile software engineering methodology for large-scale integration projects exists. The closest methodology that can be utilized for this purpose is the Architecture Development Method of The Open Group Architecture Framework (TOGAF) [15]. The process of the Architecture Development Method is iterative and cyclic, consisting of the steps of Architecture Vision, Business Architecture, Information System Architectures, Technology Architecture, Opportunities and Solutions, Migration Planning, Implementation Governance, and Architecture Change Management. Each step of the cycle interacts with the Requirements stated for the architecture. However, the Architecture Development Method is concerned only with the architecture, while we are interested in agile integration of sociotechnical systems. Noteworthy in the context of information systems' integration projects is also The Viable Software Engineering Life Cycle proposed in [19], which we will utilize in the further enhancement of the AIM.

The rest of this paper is structured as follows. In Chapter 2, we introduce the conceptual framework – the Viewpoint Framework – and populate it with the kinds of models required for engineering sociotechnical systems of systems. Chapter 3 describes the AIM methodology, proceeding by different stages of the methodology. Conclusions are drawn in Chapter 4.

2 The Viewpoint Framework

Because of the complexity of sociotechnical systems of systems, any methodology devised for engineering systems of systems should be based on a coherent and holistic conceptual framework. One of the earliest and most extensive conceptual frameworks is the Information Systems Architecture (ISA) framework, also known by its author as the Zachman framework [4]. The ISA framework [4] introduces six abstraction layers: the system's scope, enterprise or business model, system model, technology model, models of components, and the functioning system. These layers reflect the stages of systems engineering. In addition to the abstraction layers, the ISA framework defines six orthogonal aspects of a target system. These aspects are as follows. The concepts or data aspect represents the relevant conceptual objects and relationships between them. The function aspect describes the activities performed within the problem domain. The network aspect is concerned with the geographical distribution of the activities and interactions between them. The actors' or agents' aspect describes what human or manmade actors or agents perform which activities. The time aspect describes events significant to the problem domain. The motivation aspect describes the goals of the organization owning the system to be created and is also concerned with their translation into specific ends.

Other conceptual frameworks have also been proposed. The ArchiMate® conceptual framework [16] includes the abstraction layers of Overview, Coherence, and Details, and the architecture views of Informing, Deciding, and Designing. In the book [6], in addition to the ISA Framework, two other conceptual frameworks – The Enterprise Model [17] and RM-ODP [18] – have been described and compared. Based on this comparison and analysis, in [6] a simplified conceptual framework – viewpoint framework – has been proposed. The viewpoint framework is more suitable for modern agile development as compared to the ISA framework because the latter has far too many abstraction layers and orthogonal to them vertical aspects to be considered in modern agile software development [5]. The *viewpoint framework* [6] is depicted in Table I. It consists of a matrix with three rows representing different abstraction layers and three columns representing different vertical perspectives. The abstraction layers of the viewpoint framework are analysis, design, and implementation. The abstraction layers of the viewpoint framework are comparable to the respective abstraction layers of the ArchiMate® conceptual framework [16] Overview, Coherence, and Details. The vertical perspectives of the viewpoint framework are organisation, information, and behaviour. Each cell in this matrix represents a specific viewpoint, such as ''interaction analysis,'' ''information design,'' or ''behaviour implementation.'' The cells of the viewpoint framework represent artefacts – tabular models, graphical models, documents, and program code – that are produced by the AIM methodology. We will now provide an overview of artefacts of the AIM methodology, proceeding by viewpoint aspects.

From the viewpoint of *behaviour analysis*, the main artefact of the viewpoint framework is motivational scenario. A motivational scenario represents in an informal and loosely narrative manner the behaviours of actors of the given problem domain aimed at achieving a goal set in the problem domain. For example, a motivational scenario can represent the activities required for obtaining a license or permission for cross-

border service provision. Describing motivational scenarios is addressed in subchapter 3.2. Motivational scenarios can be visualised by goal models. A goal model represents in a hierarchical manner functional goals of the sociotechnical system to be designed, and attached to the functional goals quality goals applying to them and roles required for their attainment. Functional goals roughly correspond to functional requirements and quality goals – to non-functional requirements. Goal models are further described in subchapter 3.3.

From the viewpoints of *interaction and information analysis*, the main artefact is role model. A role model defines a generalised role performed by actors of the given problem domain in terms of responsibilities exercised by the actors and the information consumed, updated, and/or created thereupon. For example, a role model can define the responsibilities and informational needs of the role License Issuer played by a competent authority responsible for issuing licences for an economic activity of some kind undertaken in the destination country. Role models are treated in subchapter 3.4.

From the viewpoint of *information design*, the information to be consumed, updated, or generated is mapped to the information and communication technology (ICT) systems: information systems and registries. For example, basic company information and information about the mandates to represent the company can be mapped to the electronic business registries of the company's destination country and country of origin. The mapping of the information to the information systems and registries is described in subchapter 3.5. It is worthwhile to note here that in information systems' integration projects, the viewpoints of information analysis and design are of crucial importance.

From the perspective of *interaction design*, the main artefact is business process model. Business process models introduces temporal sequence into goal models and shows the interactions between different roles needed for the achievement of goals from the goal model. Business process models are described in subchapter 3.6.

From the viewpoint of *behaviour design*, the main artefact is user story. A user story represents how an actor of some type enacting some role can perform an activity to fulfil one or more responsibilities defined by the model of the corresponding role. For example, a user story can specify how the representative of a legal person can authorize retrieval of the basic company information and information about the mandates by means of the registration service of the destination country and electronic business registry of the country of origin. Defining user stories is described in subchapter 3.7.

Table 1. The Viewpoint Framework.

	Viewpoint aspect		
Abstraction layer	Interaction	Information	Behaviour
Analysis	Role models	Role models	Motivational scenarios, goal models
Design	Business process models	Mappings to ICT systems	User stories
Implementation	Implementation models		

3 The Methodology

3.1 The Iteration Cycle and Agility

The iteration cycle of the AIM methodology is represented in Figure 1. The iteration cycle begins with defining motivational scenarios, which describe in a loosely and narrative manner the activities to be performed in any given integration project and lists the types of actors involved in them. Creating motivational scenarios is followed by turning them into goal models and modelling the roles for actors of the motivational scenarios. Role models define roles in terms of their responsibilities to be exercised when performing motivational scenarios. Role models also define in abstract terms the information to be used, updated, and created for fulfilling the responsibilities defined for the corresponding roles. This information is used at the next step of the AIM methodology for mapping roles to the relevant information systems and registries. Subsequently, business processes are modelled for the respective motivational scenarios. Business processes include interactions with the relevant information systems and registries. Thereafter user requirements are defined in the form of user stories. Finally, user requirements are elaborated into tasks for implementing building blocks and integration and finally tasks are managed when conducting the integration projects. The last two steps represented in Figure 1 in turquoise are managed by the Trello tool (https://trello.com/).

In compliance with the principles of agile development [8], the product backlog and sprint backlog of the AIM methodology consists of user stories, which have been presented according to the format exemplified by Table 6 in subchapter 3.7. As a novelty proposed in [9], for ensuring the consistency of user stories, they have been related to the goal model of the AIM methodology described in subchapter 3.3. User stories correspond to leaf-level goals of a goal tree. A sprint length varies between 2 and 12 weeks and sprint retrospectives are held between the sprints.

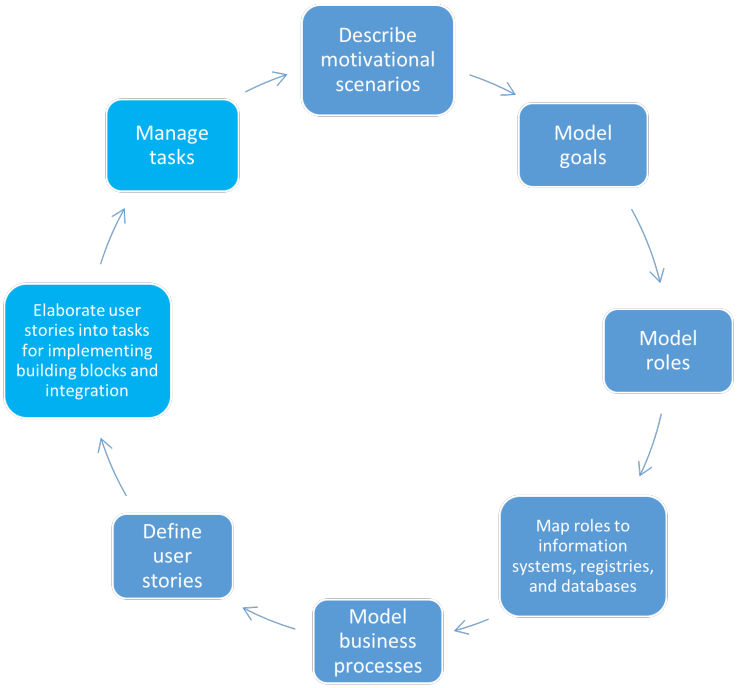


Fig. 1. The AIM methodology.

3.2 Describing Motivational Scenarios

A motivational scenario [6] is a tabular model that specifies in an informal and loosely narrative manner the activities required for achieving a goal, which is defined as a state-of-affairs. The activities of a motivational scenario are performed by actors of the corresponding types. A *motivational scenario* describes how actors of the problem domain at hand should perform their everyday work and is ***not*** concerned with ICT. A motivational scenario consists of the scenario name, types of actors involved in the scenario, goal of the scenario, steps of the scenario, and quality goals and key performance indicators. A scenario name is a freely chosen name characterizing the scenario. Types of actors involved in the scenario are types of human actors acting on behalf of companies or economic operators or representing themselves, such as Legal Person's Representative or Professional, types of human actors acting on behalf of Public Authorities, such as Civil Servant, and types of organisational actors, such as Legal Person, Public Authority, or an organisation responsible for keeping a Business Registry – Business Registry Keeper. The goal of the scenario describes the state of affairs to be achieved by taking the steps of the scenario. Steps of the scenario describe activities to be performed by actors of the scenario, in a rough chronological order, without going into details about conditions, loops, and branching. Steps of the scenario should clearly indicate what types of actors perform which steps of the scenario. For example, steps of a scenario can express that a Legal Person's Representative identifies himself/herself, and the Public Authority of the destination country communicates with the Public Authority

of the Legal Person's country of origin to request the required information about the company, as well as the mandate information authorising the Legal Person's Representative to represent the Legal Person. Finally, quality goals (also known as "soft" goals or non-functional goals) of the scenario should indicate as concisely as possible what qualities should be considered when performing the scenario, such as the reduction of administrative burden, operational costs or time, providing secure and reliable information, and automatic verifiability of information. Motivational scenarios should also show key performance indicators used for measuring the attainment of the corresponding quality goals, such as the extent of a decrease in the number of paper-based documents submitted, the sum of money or amount of time saved, and the number of fraud cases decreased. A simplified motivational scenario of retrieving basic company information and mandates for representing a company is modelled in Table 2.

Table 2. Motivational scenario of retrieving basic company information and mandates.

Scenario name	Retrieving basic company information and mandates
Types of actors involved in the scenario	Legal Person's (LP) Representative (LPR) Public Authority (PA) in the destination country PA in the country of origin
Goal of the scenario	PA in the destination country has retrieved basic company information and mandates from the PA of the country of origin
Steps of the scenario	PA of the destination country identifies LPR LPR applies to the PA of the destination country for service provision by the LP PA in the destination country requests basic company information and mandates from the PA of the country of origin LPR authorizes the PA of the country of origin to retrieve the information PA in the destination country retrieves basic company information and mandates from the PA of the country of origin PA authorizes service provision by the Legal Person in the destination country
Quality goals and performance indicators	Reduction of administrative burden for the LPR by 20% Reduction of operational costs for the PAs by 10%

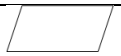
3.3 Modelling Goals

Motivational scenarios can be visualised by goal models. A *goal model* can be considered as a container of three components: goals, quality goals, and roles [6]. A goal is a representation of a functional requirement of the socio-technical system. A quality goal, as its name implies, is a non-functional or quality requirement of the system. Goals and quality goals can be further decomposed into smaller related sub-goals and sub-quality goals. The hierarchical structure is to show that the sub-goal is an aspect of the top-level goal that is required for achieving its parent goal. In other words, sub-goals model different aspects of achieving their parent goal. Goal models also show roles that are capacities or positions that agents playing the roles need to contribute to achieving the goals. In goal models, quality goals and roles are assigned to functional goals. A quality goal characterizes a quality aspect of how the corresponding functional goal (and the goals below it in the goal hierarchy) should be achieved. A role is a capacity or position that is required for achieving the corresponding functional goal (and the goals below it in the goal hierarchy). Roles are modelled in detail in the viewpoint of interaction analysis.

Attaching quality goals and roles to functional goals has two major advantages. First, it enables to see which quality goals apply to which (groups of) functional goals. Second, they bring clearly out what are the needs and intentions by different stakeholders because roles are likewise attached to functional goals. The notation for representing goal models is shown in Table 3. Because of their graphical and concise nature, goal models easily enable to see commonalities of different motivational scenarios, which are represented as sub-trees of a goal model. This is demonstrated by Figure 2, which shows one of the goal models of a company lifecycle – retrieving basic company information by the Public Authority in the destination country where a Legal Person intends to provide its services.

As has been demonstrated in [9], user stories [8], which in the AIM methodology are employed for representing requirements as is described in in subchapter 3.7, correspond to leaf-level goals of a goal tree.

Table 3. Notation for goal models.

Symbol	Meaning
	Goal
	Quality goal
	Role
	Relationship between goals
	Relationship between goals and quality goals

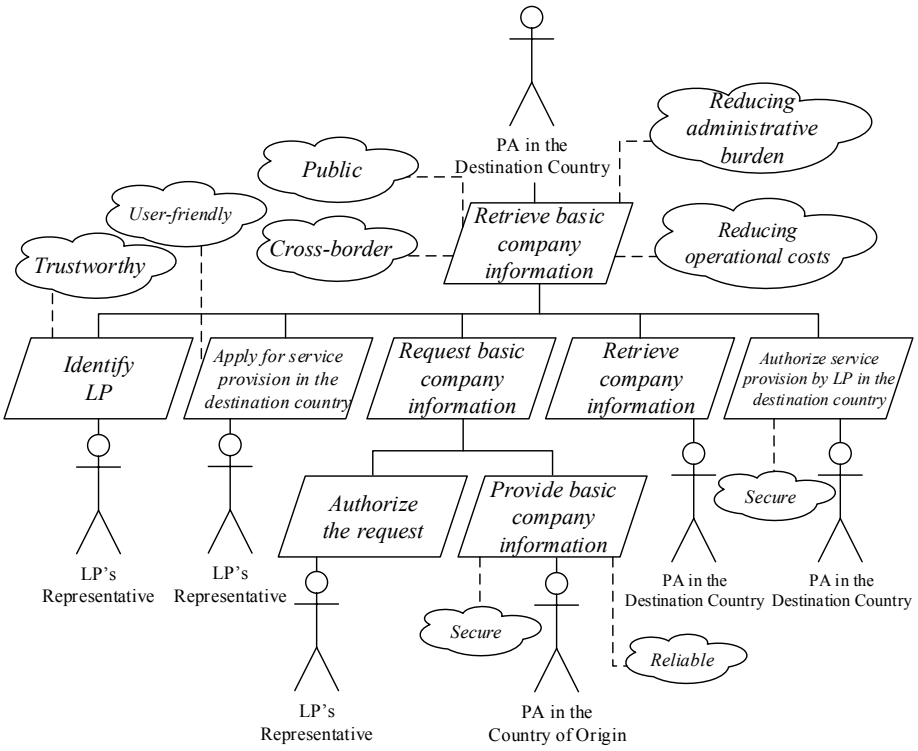


Fig. 2. Goal model for retrieving basic company information.

3.4 Modelling Roles

After having defined in motivational scenarios the types of actors and their activities, and in goal models the goals to be achieved by the sociotechnical system of systems and the roles required for achieving them, the roles need to be modelled in a detailed way. A *role* is understood as some capacity or position that is played by a human or an organisational actor [6]. For example, depending on the motivational scenario and the corresponding goal model, a human actor of the type Legal Person's Representative or Professional can perform the role Participant in the Tendering Procedure, Registree for Cross-Border Service Provision, or Service Provider. As another example, an organisational actor of the type Public Authority can perform the role License Issuer, Authorizer, Information Aggregator, or Business Registry Keeper. The difference between actor types and roles lies in their *rigidity* [6] – an actor can change its role more easily than type. Roles are defined in terms of responsibilities. A responsibility is a duty held by some actor playing the corresponding role to achieve, maintain or avoid some given state, subject to conformance with quality aspects representing legal, organisational, social and cultural norms [7]. A *role model* describes how actors of the given problem domain should exercise their responsibilities and is ***not*** concerned with ICT. Role models are orthogonal to the corresponding motivational scenario and goal model because

they show for each actor type the roles played by actors of the given type and represent responsibilities for each role. For example, the responsibilities of the role Service Provider performed by actors of the type Legal Person's Representative include identifying Legal Person and authorizing retrieval of the basic company information and information about the mandates from the Public Authority of the country of origin. As another example, the responsibilities of the role Authorizer played by a Public Authority include requesting and receiving basic company information and information about the mandates from the Public Authority of the country of origin, and authorizing service provision of the Legal Person in the country of destination.

The final aspect of role models is representing the information consumed, updated, or generated when exercising responsibilities by actors performing the corresponding roles. For example, actors of the types Legal Person's Representative performing the role Service Provider and a Public Authority performing the role Authorizer both are concerned with basic company information and information about the mandates originating in the Public Authority of the country of origin. Table 4 represents role models for the simplified motivational scenario of retrieving basic company information and mandates.

Table 4. Role models for the motivational scenario of retrieving basic company information and mandates.

Actor type	**Role**	**Responsibilities**	**Information consumed, updated, and/or created**
Legal Person's Representative (LPR)	Service Provider	Identify LPR Apply for service provision by the Legal Person in the destination country Authorize retrieval of the basic company information and information about the mandates from the PA of the country of origin	Identification information Basic company information Information about the mandates to represent the Legal Person
Public Authority (PA) in the destination country	Public Service Provider	Receive the application for service provision in the destination country Request and receive basic company information and mandates' information from the PA of the country of origin Authorize service provision by the Legal Person in the country of destination	Basic company information Information about the mandates to represent the Legal Person Authorisation for service provision in the destination country

Public Authority (PA) in the country of origin	Business Registry Keeper	Provide basic company information and mandates' information about a Legal Person in the country of origin	Basic company information Information about the mandates to represent the Legal Person

3.5 Mapping roles to information systems and registries

Motivational scenarios and goal and role models are descriptions at the level of problem domain analysis. Therefore, both modelling artefacts are included by the problem domain analysis row of the viewpoint framework represented in Table 1. Next, we are moving to the system design layer of the viewpoint framework, where we will first map roles to ICT systems – information systems and registries. These ICT systems support actors fulfilling the responsibilities included by their roles and represent information consumed, updated, and/or generated thereupon. For example, the role Business Registry Keeper, whose responsibilities deal with basic company information and information about the mandates to represent a company, is mapped to the electronic Business Registry of the company's country of origin. Similarly, the role License Issuer dealing with applications and licenses for cross-border service provision in the destination country is mapped to the competent Public Authority's information system in the destination country. Likewise, the role Tendering Participant is mapped to the destination country's e-tendering system. Table 5 represents the mapping from role models to the ICT systems for a simplified motivational scenario of retrieving basic company information.

Table 5. Mapping from role models to the ICT systems for the motivational scenario of retrieving basic company information and mandates.

Role	**Information consumed, updated, and/or created**	**Information system or registry**
Authorizer	Basic company information Information about the mandates to represent the Legal Person Authorisation for service provision in the destination country	Public Authority's information system in the destination country
Business Registry keeper	Basic company information Information about the mandates to represent the Legal Person	Business Registry of the country of origin

3.6 Modelling Business Processes

After having defined the models at the level of problem domain analysis and mapping the roles defined in problem domain analysis to ICT systems, it is time to model interactions between actors performing different roles and between the actors and the ICT systems. This is done by business process models, which are created according to the format of Business Process Modelling Notation (BPMN) [10]. The notation for BPMN is represented in Figure 3. Using this notation, the business process of retrieving basic company information is modelled in Figure 4. According to the type of business process modelled in Figure 4, Legal Person's Representative first gets identified through the corresponding service offered by the corresponding Public Service Provider of the destination country. After that, the Public Service Provider receives an application for service provision from the Legal Person's Representative and sends a message to the Business Registry Keeper of the registry containing the information about the Legal Person in its country of origin. After the Legal Person's Representative has authorised the request, the Business Registry Keeper of the country of origin provides the Public Service Provider with the requested basic company information and the Public Service Provider authorises service provision by the Legal Person in the destination country.

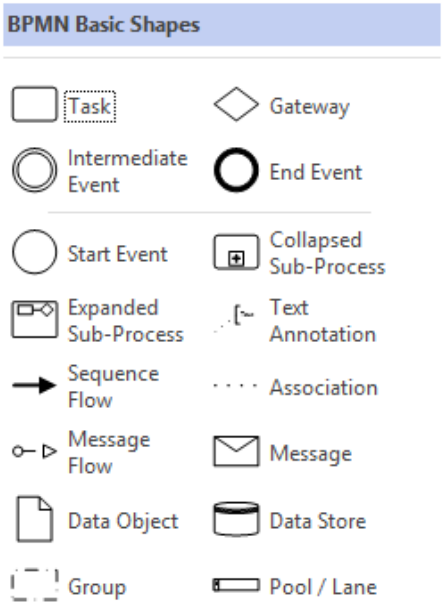


Fig. 3. Notation for BPMN.

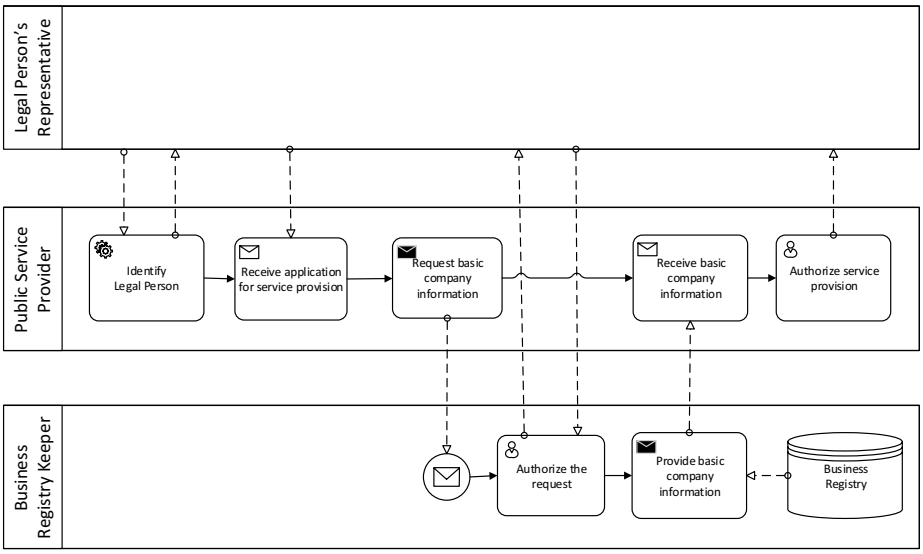


Fig. 4. Business process model of retrieving basic company information.

3.7 Defining User Requirements

By this stage, ICT systems – information systems and registries – have been determined that support actors in exercising responsibilities defined by their roles. In addition, business processes involving the roles have been modelled. Next it should be specified what the integrated system can do for actors performing their corresponding roles. This will be done by defining requirements for the integrated sociotechnical system of systems in the form of user stories. A user story [8] is a short, simple description of a feature described from the perspective of the actor who desires the new capability, usually a user or customer of the system. User stories are the most popular artefacts for representing requirements in agile software engineering. In the integration projects, a user story specifies what activities an actor of some type can perform with the help of the system integrating the relevant information systems and registries to fulfil a responsibility or responsibilities defined by the relevant role model. For example, a user story may specify that a Legal Person's Representative can authorize retrieval of the basic company information and information about the mandates by utilizing the integration of a competent Public Authority's information system in the destination country and electronic Business Registry of the country of origin. The same user story also specifies that the relevant responsibility exercised by an actor of the type Legal Person's Representative is authorizing the retrieval of the basic company data and information about the mandates and that the relevant quality goal is "Secure". Another user story can specify that a Public Authority responsible for issuing licences and permissions in the destination country can issue a license for cross-border service provision in the destination country by utilizing the integration of the Public Authority's information system in the destination country, the electronic Business Registry in the country of origin, and

the competent Public Authority's information system in the country of origin. Table 6 represents user stories for the simplified motivational scenario of retrieving basic company information and mandates.

3.8 Elaborating user stories into tasks for implementing building blocks and integration and managing tasks

At this stage, user stories are elaborated into tasks, which are programming, integration, or organisational tasks required for implementing a user story. User stories can be elaborated into tasks by means of several software project management tools, such as Trello (https://trello.com/). As an example, Figure 5 shows the management of different design and implementation and integration tasks by the Trello tool.

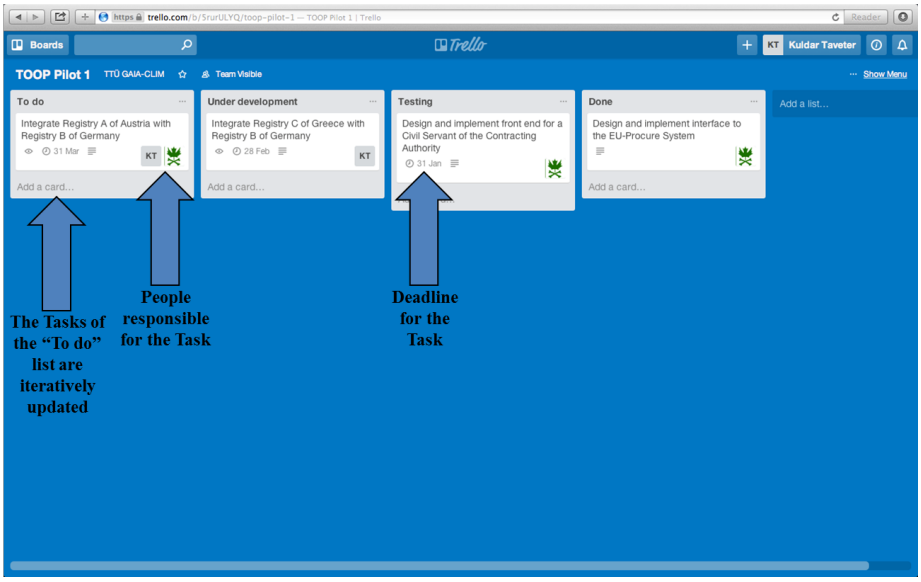


Fig. 5. Managing tasks by the Trello tool.

Table 6. User stories for the motivational scenario of retrieving basic company information and mandates.

equire- ent ID	Requirement (specifies what the user in some role can do to achieve a particular business requirement)	Information system or registry	Business requirements (from the Responsibilities)	Non-functional (quality) requirements
EQ-1	Legal Person's Representative can be identified	Public Authority's information system in the destination country	Identify Legal Person's Representative	Trustworthy
EQ-2	Legal Person's Representative can apply for service provision by the Legal Person in the destination country	Public Authority's information system in the destination country	Apply for service provision by the Legal Person in the destination country	User-friendly
EQ-3	Public Authority in the destination country can request and receive basic company information and information about the mandates from the Public Authority of the country of origin	Public Authority's information system in the destination country Business Registry of the country of origin	Request and receive basic company information and mandates' information from the PA of the country of origin	Secure, Reliable
EQ-4	Legal Person's Representative can authorize retrieval of the basic company information and information about the mandates	Public Authority's information system in the destination country Business Registry of the country of origin	Authorize retrieval of the basic company information and information about the mandates from the Public Administration of the country of origin	Secure

4 Conclusions

A major trigger for the research work reported in this paper was a large-scale integration project that aims to connect 60 registries and information systems from 22 countries. No off-the-shelf agile software engineering methodology exists for this project and other projects of a similar kind and magnitude. Therefore, a new agile software engineering methodology for running large-scale information systems' integration projects has been defined and reported in this paper. The methodology is currently used in the TOOP project and will also be utilized in another large-scale integration project that is currently being prepared.

References

1. Sommerville, I. 2010. *Software engineering*. London, England: Pearson Education.
2. Popper, S., Bankes, S., Callaway, R., and DeLaurentis, D. 2004. *System-of-Systems Symposium: Report on a Summer Conversation*, July 21–22. Arlington, VA: Potomac Institute for Policy Studies.
3. Kotov, V. 1997. *Systems-of-Systems as Communicating Structures*. Hewlett Packard Computer Systems Laboratory Paper HPL-97-124, pp. 1–15.
4. Sowa, J. F., and J. A. Zachman. 1992. Extending and formalizing the framework for information systems architecture. *IBM Systems Journal, 31*(3), 590–616.
5. Rajlich, V. 2006. Changing the paradigm of software engineering. *Communications of the ACM*, *49*(8), pp. 67-70.
6. Sterling, L., and Taveter, K. 2009. *The art of agent-oriented modelling*. Cambridge, MA and London, England: MIT Press.
7. Sommerville, I., Lock, R., Storer, T., and Dobson, J. 2009. Deriving information requirements from responsibility models. In *International Conference on Advanced Information Systems Engineering* (pp. 515-529). Berlin and Heidelberg, Germany: Springer.
8. Cohn, M. 2004. *User stories applied: For agile software development*. Boston, MA: Addison-Wesley.
9. Tenso, T., Norta, A., and Vorontsova, I. 2016. Evaluating a novel agile requirements engineering method: A case study. *ENASE 2016 - Proceedings of the 11th International Conference on Evaluation of Novel Approaches to Software Engineering, Rome, 27-28 April* (pp. 156−163). SciTePress.
10. Chinosi, M., and Trombetta, A. 2012. BPMN: An introduction to the standard. *Computer Standards & Interfaces*, *34*(1), 124-134.
11. Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., ..., and Paige, R. 2012. Large-scale complex IT systems. *Communications of the ACM*, *55*(7), 71-77.
12. Taveter, K., and Wagner, G. 2002. A multi-perspective methodology for modelling inter-enterprise business processes. In: *Conceptual Modelling for New Information Systems Technologies, ER 2001 Workshops HUMACS, DASWIS, ECOMO, and DAMA. Revised Papers. Ed. H. Arisawa; Y. Kambayashi; V. Kumar; H. C. Mayr; I. Hunt*. Springer, 403-416.
13. Taveter, K. 2004. *A multi-perspective methodology for agent-oriented business modelling and simulation*. PhD thesis, Tallinn University of Technology, Estonia. Tallinn University of Technology Press (ISBN 9985-59-439-8).

14. Norta, A., Dua, Y., Ma, L., Rull, A., Kolvart, M., and Taveter, K. 2015. eContractual Choreography-Language Properties Towards Cross-Organizational Business Collaboration. *Journal of Internet Services and Applications, 6* (8), 1-23.
15. TOGAF. 2011. *Open Group Standard, TOGAF Version 9, Part II*. Document Number: G116 (ISBN: 9789087536794). The Open Group. Last accessed on 11 September 2017 from http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html.
16. ArchiMate. 2013. *ArchiMate® 2.1 Specification, Part 8*. Document Number: C13 (ISBN: 1-937218-43-L). The Open Group. Last accessed on 11 September 2017 from http://pubs.opengroup.org/architecture/archimate2-doc/toc.html.
17. Bubenko, J. A. 1993. Extending the scope of information modelling. In A. Olivé (ed.), *Fourth International Workshop on the Deductive Approach to Information Systems and Databases, September 20-22, Lloret de Mar, Catalonia, Proceedings* (DAISD 1993). Report de recerca, LSI/93-25-R, Departament de Llenguatges i Sistemes Informatics (73-97). Barcelona, Spain: Universitat Politecnica de Catalunya (UPC).
18. Putnam, J. R. 2001. *Architecting with RM-ODP*. Upper Saddle River, NJ: Prentice Hall.
19. Atkinson, C., and Draheim, D., 2013. Cloud-Aided Software Engineering: Evolving Viable Software Systems Through a Web of Views. In: *Software engineering frameworks for the cloud computing paradigm* (pp. 255-281). Springer.