Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1221–1228ISSN 2300-5963 ACSIS, Vol. 11 DOI: 10.15439/2017F96

Enterprise Architecture Approach to SCRUM Processes, Sprint Retrospective Example

Jan Werewka, Anna Spiechowicz
Department of Applied Computer Science
Faculty of Electrical Eng., Automatics, Computer Sci. and Biomedical Eng.
AGH University of Science and Technology
al. Mickiewicza 30, 30-059 Kraków, Poland
werewka@agh.edu.pl

***Abstract*—Enterprise architecture supports a holistic approach used to optimize various activities of a company. Software development companies frequently use a popular agile approach, and the most popular agile methodology is Scrum. A sprint retrospective is a Scrum process which is supposed to enable self-development and improve communication among team members. Unfortunately, the reality is usually different. The aim of the paper is to identify problems with retrospectives and to use enterprise architecture models to help different stakeholders to understand the problems of agile approach and to find reasons why sometimes it does not meet its goals. Next, the authors try to find solutions for the identified problems on the basis of a persona concept.**

***Index Terms*—Scrum; agile; retrospective; team work; enterprise architecture; ArchiMate**

I. Introduction

ENTERPRISE architecture may be used for an efficient development of a company. To optimize the structure and behavior of a company, a holistic approach should be used in which the company architecture should be modelled in a uniform way. One of the reasons why it is worth pursuing this approach is the fact that it offers different company stakeholders a clear understanding of what is going on. In the literature these problems are closely related to integration and interoperability of enterprise architecture [1]. Other important ideas regarding enterprise architecture evolution are standardization and harmonization [2].

There are many examples of processes that do not work so efficiently as expected. A clear understanding is useful not only to people taking part in the process, but also to people observing the process. So, there is a need to describe the behavior of a company in a uniform way. The structure and behavior of a company can be modelled in ArchiMate [3] language, which supports strategy, motivation, business, technology and physical layers concepts. Developing such models is especially valuable for companies developing software (software houses).

Software houses usually apply a mix of different methodologies, e.g. agile or classic. In these companies service orientation and continuous improvement seem an important goal. To obtain a full picture of the company, it is important to distinguish services in the software production, and to show how these services are supported by processes.

In the paper an ArchiMate model is proposed for iterative software development based on Scrum. The model is based on a more general model of the Scrum framework for teams and a more detailed one for a sprint retrospective. Scrum Guide [4] is a widely accepted reference publication for Scrum.

A sprint retrospective is a meeting that takes place during the last stage of the sprint. During the meeting the participants discuss the finished sprint, focusing on the team and its problems. The main topics of the conversations during a retrospective focus on what went well, what difficulties occurred during the sprint, and how to take corrective actions to avoid similar problems in the future. In Scrum, a sprint retrospective is an integral part of control and adaptation processes, without which the team cannot develop and improve the efficiency of its work.

In many Scrum team members' opinion, a retrospective is a fragile process. To check this view, a survey was conducted by the authors of the paper in which they collected information on the most common problems raised up during a sprint retrospective. Unfortunately, its results do not lead to optimistic conclusions. The main research question was why most teams using an agile approach do not conduct such meetings at all or end them with negative results. Answering this question can be followed by additional questions: Are the authors of Scrum and the authors of books praising the advantages of retrospectives incurable optimists? Are they misled in their assertions? Or maybe the way of conducting a retrospective leads to mistakes?

There are many books, papers and blogs which describe how to conduct a retrospective in a proper way. Usually their authors are experts in coaching agile teams. However, it is difficult to find a common and uniform understanding of a retrospective process. That is why a more general approach is proposed based on an enterprise architecture model. To understand its idea better, a retrospective motivation model and a concept of UX (user experience) are proposed.

The structure of the paper is as follows. After the introduction in the first chapter, an enterprise architecture view of Scrum is proposed in the second chapter, and selected literature concerning a retrospective is analyzed in the third chapter. In the fourth chapter a base motivation model of a retrospective is proposed, while in chapter five common problems of a retrospective are investigated based on a survey, whose results are discussed in the next chapter. The success of a retrospective depends on people taking part in it, and that is why personas and retrospective roles are discussed respectively in chapters seven and eight, which are followed by chapter 9 discussing ways of improving a retrospective. Conclusions are devoted to future research concerning the integration of enterprise architecture and models of agile methodologies.

II. Enterprise Architecture Approach to Scrum

Enterprise architecture is used to obtain a holistic view of the company. From this point of view, the most suitable enterprise architecture definition is the one given by Lankhorst [5] as "a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure". The second important point is competitive development of a company, which is best defined by Gartner Group [6] "Enterprise architecture (EA) is the process of translating business vision and strategy into effective enterprise change by creating, communicating, and improving the key principles and models that describe the enterprise's future state and enable its evolution." In the paper both definitions of enterprise architecture are important: the one regarding the company's holistic view and the one regarding its competitive evolution.

The developers of agile methods try to differentiate them from other methods by different means. There is no essential reason why integrating Scrum into enterprise architecture could not be possible. Literature offers some examples of such proposals, e.g. [7], [8]. The development of enterprise architecture can be proposed for different fields [9]. IT enterprises running projects in different heterogeneous environments integrate classical and agile project management methodologies. Paper [10] discusses the problem of alignment of two project management methodologies based on two ontologies: a classical one represented by PMBOK [11] and an agile one represented by Scrum [4]. In [12] the problem of selecting a suitable agility framework is discussed, and the analysis ends with the conclusion that the investigated methodologies were inconsistent.

To describe project management methodologies or agile frameworks, a common meta-model may be proposed. In the area of software engineering, different meta-models for harmonizing different standards and solutions are offered. In the domain of software development, harmonization is proposed for ISO standards based on the Software Engineering Metamodel for Development Methodologies (SEMDM) described in ISO/IEC 24744 standard. For example, in [13] a proposal of such harmonization is based on Ontology Pattern Language.

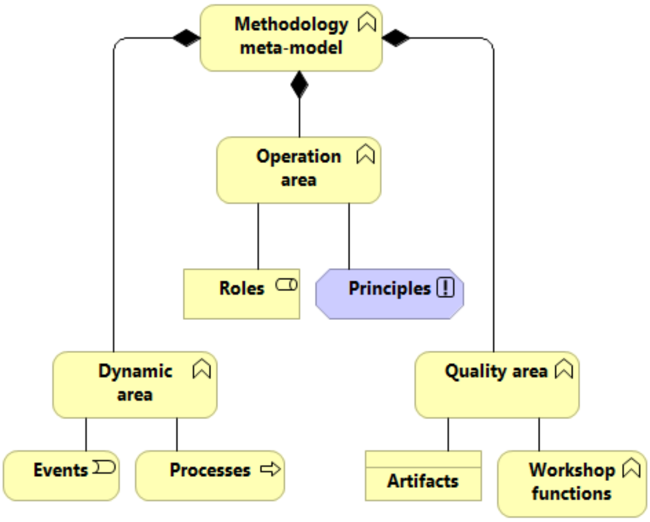

Figure 1. Meta-model of project management metodologies

The paper proposes an approach which harmonizes the project and product development methodologies using enterprise architecture concepts. The proposed meta-model is simple and distinguishes the following areas (Fig. 1): dynamic (describing activities), operation (describing active elements, like roles and principles governing the methodology), quality (depending artifacts, i.e. inputs and outputs of processes) and workshop functions (e.g. the application of selected tools, techniques, and practices).

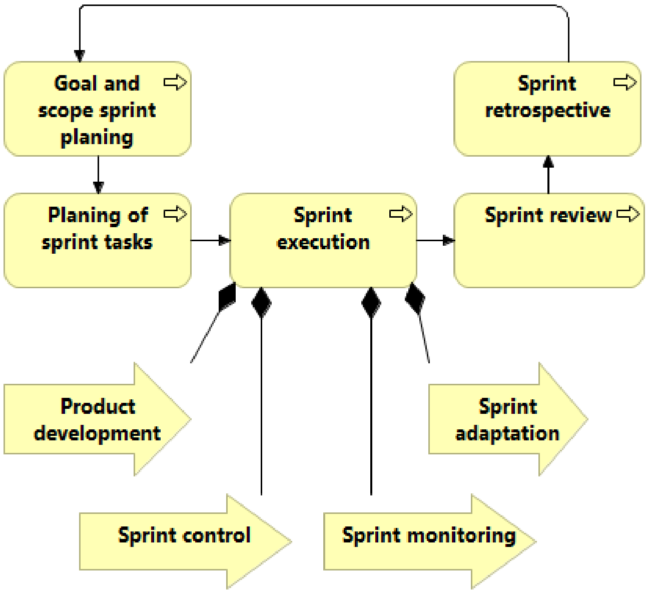

Figure 2. General Scrum process model

In building a Scrum model, Scrum processes should be considered at first. ArchiMate is one of the best known languages used for describing enterprise models. A simple example of a model of processes in Scrum with triggering and composition relations is given in Fig. 2. Sprint, the Scrum iteration, contains planning, execution, review and retrospective processes. The sprint execution process consists of composite sub-processes: product development, sprint control, monitoring and adaptation. These sub-processes are running parallel during sprint execution.

The full Scrum model based on the meta-model is presented in [14] from two viewpoints: the team's viewpoint and the business owner's viewpoint.

III. State of the Art on a Sprint Retrospective

A sprint retrospective is the last meeting during the sprint. Its main objective is to facilitate the development and self-improvement of the team. During a retrospective its participants discuss the events that have taken place, their impact on their work and how they may be able to deal with problems in the future. The retrospective process is discussed and described in detail in different publications. The Retrospective Handbook [15] shows ways of running more effective retrospectives by addressing certain practical challenges.

A retrospective should not be conducted in a chaotic way. It is very important to prepare, conduct and close a retrospective properly. It is assumed [7] that for a one month sprint, a retrospective should last up to 3 hours. The authors of [16] suggest dividing a sprint retrospective into five phases of varying lengths:

1) Setting the stage. The first phase involves familiarizing the group with the timetable and retrospective goals, and, additionally, doing exercises that will make it easier for shy people to express their opinions later.
2) Gather data. The data collection is intended to remind the group of events taking place during a sprint. During this phase the team should recall all events that occurred during a sprint, such as meetings, decisions, milestones, integration meetings, rotation of team members as well as adaptation of new technologies. The Timeline *[17]* exercise may be proposed here.
3) Generate Insights. Through brainstorming the team members try to notice the correlation between the events and the quality and effectiveness of their work. Through this analysis, it will be possible to identify which events help and which make it more difficult for the team to achieve the goal. Exercise that can help during brainstorming is called 5 Whys method, which may be used in software development to prevent recurrence of the same problems *[18]*.
4) Decide what to do. During this phase, while working in groups, team members create a detailed list of actions to be performed during the next sprint. The most important element of this stage is the selection of 2-3 most important factors that affected the last sprint issues, in which the "Planning Game" *[19]* exercise may be helpful.
5) Close the retrospective. In the end, the leader will gather feedback on the meeting. It is important to collect the results of the analysis, the list of decisions made and to create a common picture in the form of a poster.

On the basis of this proposal, phases and their goals can be described in ArchiMate model as shown in Fig. 3. The sprint retrospective process consists of sub-processes realizing different goals. The person preparing the meeting is obliged to prepare the place where a retrospective will take place. A good choice will be a location in which the team usually works or an isolated room allowing participants to arrange chairs in a semicircle so that everyone can see one another. The room should offer a possibility to post posters or draw graphs and timelines. This will create a proper atmosphere for the discussion.

Sometimes a retrospective always performed in the same way becomes boring for some team members or for the whole team, however, it is possible to revitalize retrospective meetings. In [20] seven principles serving this purpose are defined: (1) Rotate leadership. It seems natural that the Scrum Master should take leadership of a retrospective, but rotating leadership among team members can bring good results. (2) Change the question. By asking standard questions, we usually obtain the same answers. (3) Vary the process. Some tools may be proposed to structure the process. (4) Include different perspectives. Take into consideration the viewpoints of different stakeholders. (5) Change the focus. Change the focus to e.g. social communication, organization or engineering issues. (6) Try appreciative inquiry. Consider good results obtained in the past to explore how to use them in the future. (7) Analyze recurrent themes. If the same themes occur, try to influence the situation or change the plans.

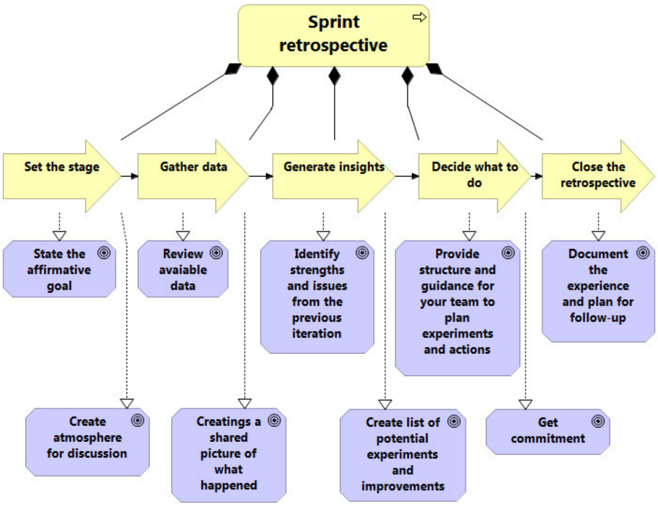
Sprint retrospective
Set the stage
Gather data
Generate insights
Decide what to do
Close the retrospective
State the affirmative goal
Review avaiable data
Identify strengths and issues from the previous iteration
Provide structure and guidance for your team to plan experiments and actions
Document the experience and plan for follow-up
Create atmosphere for discussion
Creatings a shared picture of what happened
Create list of potential experiments and improvements
Get commitment

Figure 3. Retrospective phases and their main goals

There are some works offering exercises facilitating a retrospective. Book [21] contains many practical exercises aimed at becoming more proficient in performing retrospectives. Similarly, [22] provides a tool set of activities to transform a group of people into an effective team by keeping the participants amused and providing a setting where they can reflect, discuss and have fun. Another issue is team networking, which is performed outside a retrospective, and for which collaborative games may be proposed, e.g. [23].

Identify applicable sponsor/s here. *(sponsors)*

IV. Enterprise Architecture Approach to a Sprint Retrospective

A case of a retrospective process will be investigated here in more detail. A sprint retrospective takes into account (Fig. 4) such inputs as: work progress during the sprint (represented by sprint burn down charts), delivered product at the end of the sprint, product development recommendation made during a sprint review, team capability and retrospective recommendations from previous sprints. A burn down chart, product development and retrospective recommendations are Archimate business objects. A sprint retrospective triggers a process of retrospective recommendation realization.

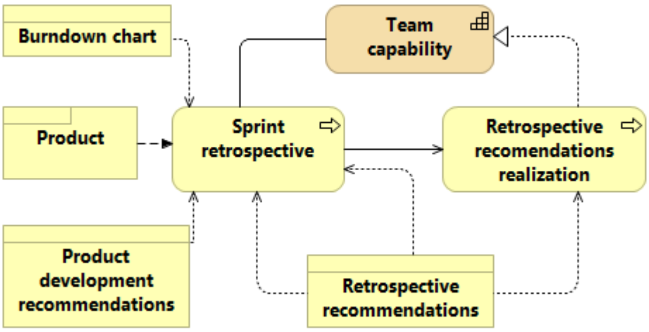
Burndown chart
Team capability
Product
Sprint retrospective
Retrospective recomendations realization
Product development recommendations
Retrospective recommendations

Figure 4. Model of a sprint retrospective

In building an enterprise model a motivation layer should be developed in the first place. Fig. 5 presents retrospective motivation using the following ArchiMate concepts: stakeholder, driver, assessment, principle, requirement, constraint, goal and value. The motivation model for this simplified detail level may be used for a quick check of what the reasons to perform a retrospective are.

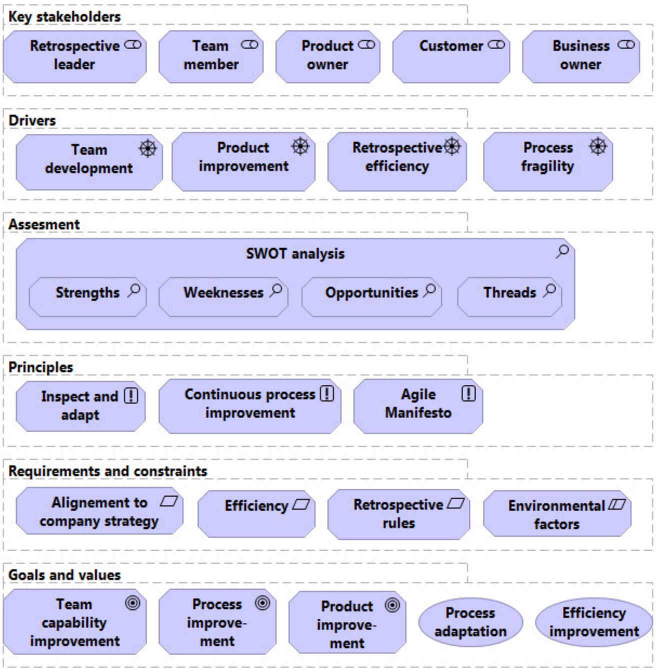

Figure 5. Motivation model of a sprint retrospective

In a full Scrum model the motivation elements are tied to other kinds of motivation, a business layer, and strategy elements.

V. Common Issues During a Sprint Retrospective

Obtaining and maintaining a holistic view is an important issue in developing enterprise architecture, but in architecture evolution harmonization and efficiency become the most important goals. The problem can be defined as follow: is a retrospective the best way to develop and improve efficiency of team work?

In order to determine the problems related to a sprint retrospective, a written questionnaire was distributed among people working in Scrum projects. The survey was conducted between May and June 2016 in Cracow and its neighboring areas, which are the largest outsourcing center in Europe [24]. 32 survey participants selected for the survey were members of different Scrum teams from different companies. Their job positions were as follow: 59% were software developers, 38% quality assurance engineers and 3% business analysts. Their work experience ranged between 0-5 years (66%), 6-10 years (21%), 11-15 years (105) and 16-20 years (3 %).

The first part of the survey concerned the position and the work experience of the respondents. In the second part general questions regarding retrospective were as follow: Does the team in which you are working conduct sprint retrospectives? How long does an average sprint take in your project? What is the average duration of a retrospective in your project? Do you prepare for a retrospective? How many exercises does your team perform during a retrospective?

Then, in the third part of the survey the respondents were asked to what extent they agreed with the following statements: (1) I know why a retrospective is carried out. (2) I understand the meaning of a retrospective. (3) A retrospective is a necessary meeting during the sprint. (4) During a retrospective we discuss important things from the team's and project’s points of view. (5) A retrospective brings a lot of changes to the team. (6) During a retrospective I make many observations. (7) I want to share my observations during a sprint retrospective. (8) I feel that my participation in a retrospective is important. (9) A retrospective takes the appropriate length of time. (10) A retrospective is interesting. (11) A retrospective drives the team to action. (12) A retrospective is not used to indicate the person responsible for the success / failure of the sprint. (13) A retrospective improves communication between team members. (14) A retrospective improves the team's work organization. (15) A retrospective motivates for better work. (16) A retrospective improves team efficiency. (17) A retrospective reveals strengths and weaknesses of the team. (18) A retrospective reveals the problems that occur in the project. (19) A retrospective allows for the development of team members. (20) A retrospective is carried out correctly.

The survey participants chose a decimal value assigned to each answer ranging from 1 (strongly disagree) to 5 (strongly agree). Finally, the respondents were asked for other suggestions or opinions related to a retrospective, which were not included in the questionnaire.

VI. Survey Results

The survey revealed that only 67% of surveyed people working in Scrum projects participate in a retrospective. Thus, only the answers of the respondents who have actually taken part in retrospectives are analysed.

Most people declared that sprints in their projects last for 2 weeks (55%), 3 weeks (30%) or 4 weeks (15%). Regarding the duration of a retrospective, it takes up to 1 hour in 40% of cases, 2 hours in 55% of cases and 3 hours in 5% of cases. During a retrospective, one exercise (75% cases) is performed as a standard, and its goal is to determine the problems that Scrum teams face. It is regrettable to say that only 35% of the respondents prepare for a sprint retrospective.

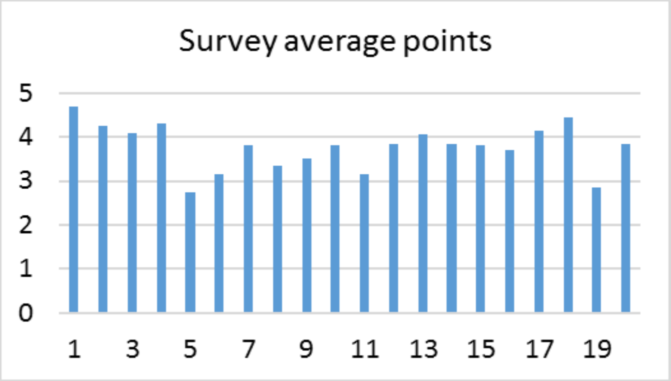

Figure 6. Average points for 20 questions of the survey

Twenty questions from the third part of the questionnaire make it possible to determine the satisfaction index of a sprint retrospective. For the purpose of this study, satisfaction was measured by assigning points between 1 (strongly disagree) to 5 (strongly agree) as the answers to the questions. The average values are presented in Fig. 6.

The article presents in detail only some of the most surprising answers to the questions. Only 26% of the respondents (Fig. 7) agreed or partially agreed with the assertion that a retrospective leads to many changes in the team's work. At the same time, only 45% (Fig. 8) of them thought that a retrospective motivates them to work better. In addition, only 30% said (Fig. 9) that a retrospective drives the team to action. Even worse, only 20% partially agreed (Fig. 10) that a retrospective leads to the development of the team members. No one was completely convinced of the validity of this statement. When it comes to identifying the problems faced by the participants of the meeting themselves, it was not better, either. Only 40% of the respondents (Fig. 11) made some observations during the meeting. A better situation occurred with sharing ones insights with others - 85% of the respondents (Fig. 12) declared such a desire.

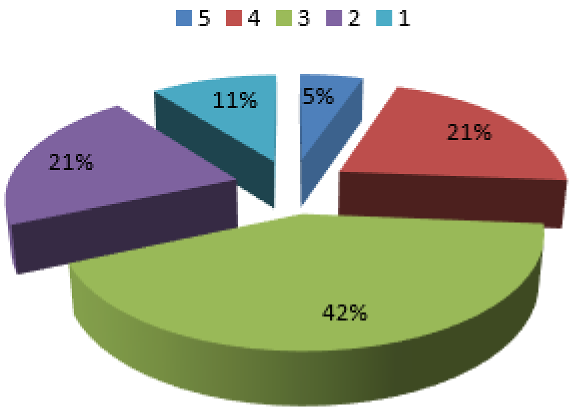

Figure 7. A retrospective introduces many changes to team work

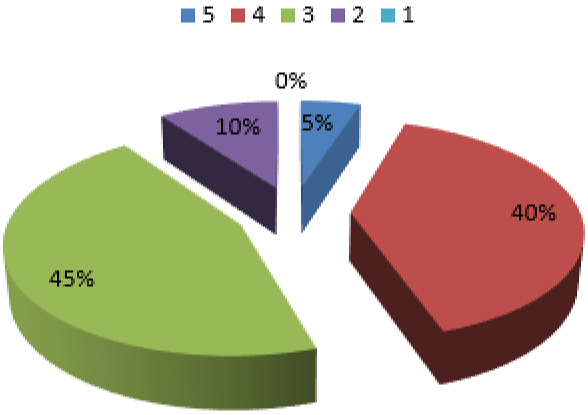

Figure 8. A retrospective motivates for better work

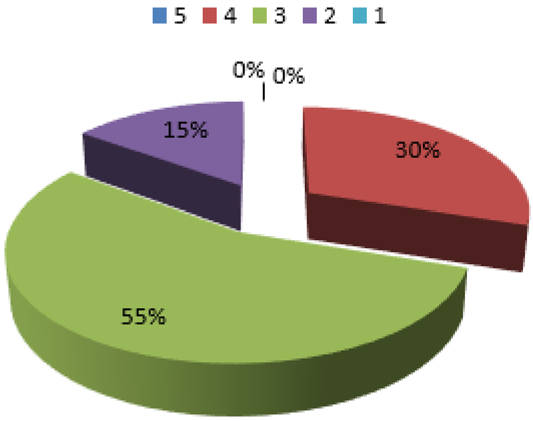

Figure 9. A retrospective drives the team to action

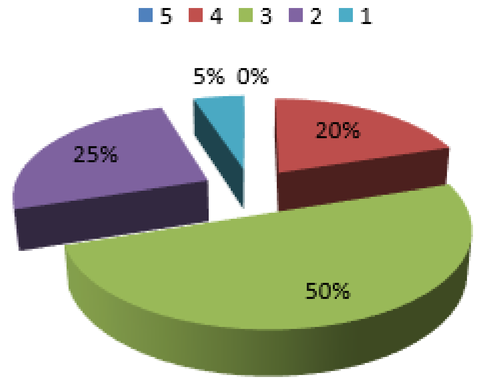

Figure 10. A retrospection enables the development of team members

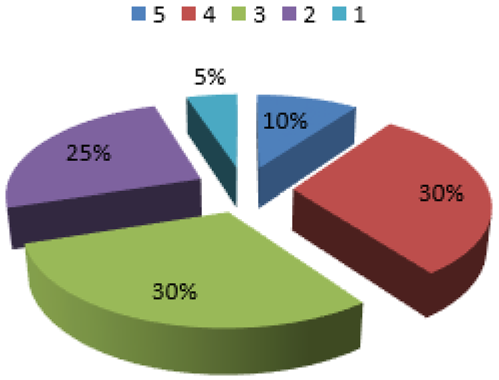

Figure 11. During a retrospective I have many insights

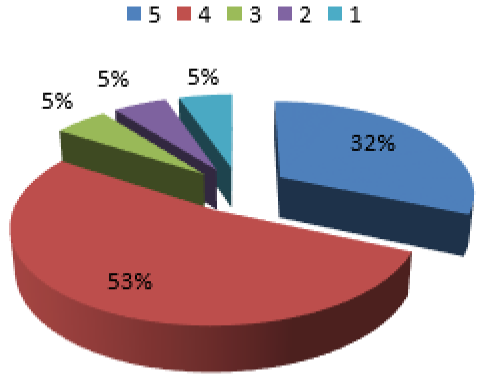

Figure 12. I want to share my observations during a retrospective

These responses indicate that a retrospective created to give the team the opportunity to develop, motivate and make changes to the process, in most cases does not fulfill its core functions.

Nevertheless, the respondents also noticed a more positive side of a retrospective. They claimed that such meetings verify the team's integration and enable the exchange of experiences between the participants. In addition, a retrospective reveals the problems that have not been noticed during the sprint.

VII. Applying a Persona Concept to Retrospective Roles

An important concept in enterprise architecture modelling is service realized by processes. In the area of service development many solutions are proposed, such as service design thinking or user and customer experience. The question is why not use these solutions to improve agile processes. In the service design a persona is an important concept. The idea of understanding customer segments was proposed by Angus Jenkinson [25] based on creating imagined or fictional characters with certain behaviors and attitudes which represent customer segments or communities. Success of a retrospective depends on the retrospective leader and team members. For that reasons it would be interesting to develop a persona model for the retrospective leader and team members. Fig. 9 presents the persona model expressed in ArchiMate language.

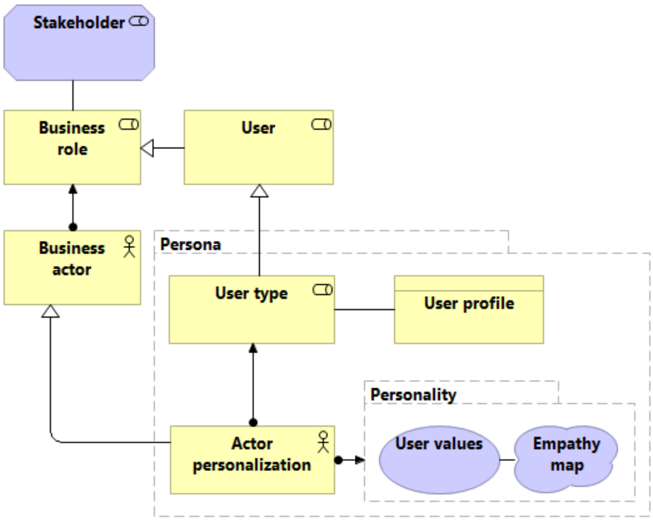

Figure 13. Persona concept

In building a persona model of a team member, at the beginning his profile should be determined, which can be characterized by: experience, job seniority, age, education, family background, and so on. An important issue is to uncover his personality, e.g. motivation, attitude, feelings and the way of thinking. The definition of a persona for a retrospective should be followed by customer journey canvas designed for each persona. An important task while developing a persona and customer journey canvas is to build an empathy map.

The same considerations are valid for the retrospective leader. This role is of great importance for realizing goals of a retrospective. That is why the feelings of the surveyed persons regarding this role seem interesting.

VIII. Sprint Retrospective Leader Roles

Success of a retrospective depends on the retrospective leader and team members. The behavior of the retrospective leader and team members depends on the retrospective phase. A retrospective consists of three phases: preparing for a retrospective, conducting a retrospective, and post-retrospective actions.

Preparing for a retrospective. Survey results confirmed that most participants of a retrospective are not prepared for it. The leader should investigate what occurred during the last sprint, exceptional events that took place, the feelings of the team members, the artifacts and the attractors. These should help to outline and understand the problems the team was facing, and, ultimately, to set a retrospective goal. Outlining the purpose of a retrospective makes the team members see the reason why they are going to spend their time attending the meeting. Retrospective duration depends on many factors, including, e.g. the sprint length, the complexity of the project, technologies used, the size of the team, the level of conflict in the team and the issues that arouse controversy. The next step in the preparation of a retrospective is setting up a schedule. The meeting should be typically divided into phases. For each of these phases, special activities must be prepared that allow the group to mobilize to work together. These activities are designed to encourage team members to actively participate in the meeting, increase their creativity and focus on the topic. It is important not to use only one exercise during retrospective meetings, because doing it repeatedly becomes boring and does not promote creative thinking.

Conducting a retrospective. The leader should focus on the process and the structure of a retrospective. Adapting to the needs and dynamics of the team should help the team members to reach the goal while staying neutral in a discussion, even if they have their own insights. In addition, the leader, as the person who knows the schedule of a retrospective, has to present each activity before starting it. This will make the team aware of what they can learn from exercises. Every activity should end with a discussion on the obtained effects and conclusions drawn from it. An important task of the leader is to observe the activities of the participants of a retrospective. The leader should draw attention to these who do not participate at all, encouraging them to take part in the discussion and express their opinions, as well as to these who dominate the conversation. During a retrospective, the retrospective leader listens attentively to participants' speeches, trying to capture the signals of the blame or clutter of the team members. Seeing what the conversation is about, the leader should try to change its course. Moreover, it may happen that during a meeting the team loses the sense of time and here emerges another role of the leader connected with managing the time. The leader should give signals so that the meeting proceeds according to the schedule.

Post-retrospective actions. The basic principle of a retrospective is - after inspecting - making adaptations in the next sprints. If only a small part of the proposed changes are implemented, the team may become frustrated and not willing to take part in retrospectives in the future.

IX. Sprint Retrospective Improvement Considerations

A retrospective is a part of Scrum, which, unfortunately, is often ignored, depreciated and neglected. As a result, the team cannot obtain desired results from such meetings. But it is assumed that a retrospective influences team development, improves the efficiency of its work and communication between team members. Without a good retrospective, it is not possible for the team to improve their performance.

Unfortunately, the results of the survey reveal that in most cases retrospectives do not meet the assumptions. A large group of the survey participants failed to notice whether such meetings make proper changes to the team and, in particular, whether they allow them to develop. Does this mean that the belief in the power of a retrospective is only a myth?

In the survey several people noticed certain additional issues during a sprint retrospective. It happened that, despite the improvement resolutions, the team members lacked the consistency in implementing them. In addition, during the meeting itself, the participants often lacked the discipline, which was often the leader's fault.

At the same time, the survey showed that satisfaction with retrospectives was strongly linked with the number of exercises done during meetings and almost did not depend on earlier preparation for them. The more activities were done during the meeting, the more positive effects of the action were noticed by the team members. In addition, a retrospective was better assessed by people who prepared for it.

The analysis of the survey results allowed us to formulate tips helping to fix a retrospective and make it work as expected. One of the most important factors influencing success of a retrospective is the choice of the leader. The leader should be a good observer who can encourage shy people to actively participate in the meeting, while diminishing the behavior of those overactive. In addition, the leader should be a person who knows the schedule of the meeting best and possesses thorough knowledge of how each exercise should be performed during the meeting.

This is directly related to the preparation of a person to lead a retrospective, which consists not only of the preparation of a proper place together with the needed materials, but also of learning about the course of the sprint and the situation in the project, and to determine the purpose of a retrospective. It is also a good habit to divide the meeting into five phases of varying lengths, each with different duration and aim. The direct result of this division is the activity performed during each of the phases aimed at achieving the goal of each phase. Taking these tips into account when preparing for the next retrospective will certainly make the team's attitude to a retrospective meeting more positive and will enable them to notice its positive impact on their work.

Sometimes the reasons why a sprint retrospective does not work well are known, and the time needed for solving problems is much longer then the sprint length. In this case alternative measures should be considered, like e.g. forwarding problems to an issue log system to solve impediments and perform team networking exercises to enhance cooperation.

X. Conclusions

Enterprise architecture is a means of improving and understanding activities in a company in a uniform and holistic way. Such an approach may be proposed for agile and classic software products development processes. The presented enterprise architecture models may be suitable for different stakeholders, because, as shown above, they can model different methodologies with sufficient details and in a way that is understandable to different stakeholders.

Enterprise architecture models are good measures of the improvement of company processes. The main driver of the paper were survey results concerning a sprint retrospective, which showed numerous weaknesses of a retrospective. The main aim was to look at a retrospective from the perspective of enterprise architecture and to find suitable solutions from the company's perspective. The proposed model may be used to compare different approaches. Developing a detailed model of a retrospective will make it possible to asses it from the outside.

The obtained results are a good starting point for developing a more comprehensive and consistent model with sub-models related to project and product management in software houses.

References

[1] D. Chen, G. Doumeingts, and F. Vernadat, "Architectures for enterprise integration and interoperability: Past, present and future," *Comput. Ind.*, vol. 59, no. 7, pp. 647–659, Sep. 2008.

[2] C. Pardo, F. J. Pino, F. García, M. Piattini, and M. T. Baldassarre, "An ontology for the harmonization of multiple standards and models," *Comput. Stand. Interfaces*, vol. 34, no. 1, pp. 48–59, Jan. 2012.

[3] The Open Group, "ArchiMate® 3.0 Specification," 2016. [Online]. Available: http://pubs.opengroup.org/architecture/archimate3-doc/toc.html. [Accessed: 27-Jan-2017].

[4] K. Schwaber and K. Sutherland, *Scrum Guide*. 2016.

[5] M. Lankhorst, *Enterprise Architecture at Work*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.

[6] A. Lapkin *et al.*, "Gartner clarifies the definition of the term'enterprise architecture'," 2008.

[7] S. Hanschke, J. Ernsting, and H. Kuchen, "Integrating Agile Software Development and Enterprise Architecture Management," 2015, pp. 4099–4108.

[8] J. Werewka, K. Jamróz, and D. Pitulej, "Developing Lean Architecture Governance at a Software Developing Company Applying ArchiMate Motivation and Business Layers," in *Beyond Databases, Architectures, and Structures*, vol. 424, S. Kozielski, D. Mrozek, P. Kasprowski, B. Małysiak-Mrozek, and D. Kostrzewa, Eds. Cham: Springer International Publishing, 2014, pp. 492–503.

[9] M. Lankhorst, *Enterprise Architecture at Work*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[10] P. Szwed, J. Werewka, and G. Rogus, "Ontology based alignment of classic and agile project managment for an IT enterprise," *Zesz. Nauk. Wydzialu ETI Politech. Gdań.*, vol. 19, pp. 345–350.

[11] PMI, *A Guide to the Project Management Body of Knowledge: PMBOK(R) Guide*, 5 edition. Newtown Square, Pennsylvania: Project Management Institute, 2013.

[12] R. Wendler, "The Structure of Agility from Different Perspectives," in *2013 Federated Conference on Computer Science and Information Systems*, 2013, pp. 1177–1184.

[13] F. B. Ruy, R. A. Falbo, M. P. Barcellos, and G. Guizzardi, "Towards an ontology pattern language for harmonizing software process related ISO standards," 2015, pp. 388–395.

[14] J. Werewka, G. J. Nalepa, M. Turek, T. Włodarek, S. Bobek, and K. Kaczor, *Project management in IT company. Vol. 3, Project and software development processes management*, vol. 3. Kraków: AGH Press, 2012.

[15] M. P. Kua, *The Retrospective Handbook: A guide for agile teams*. CreateSpace Independent Publishing Platform, 2013.

[16] E. Derby, D. Larsen, and K. Schwaber, *Agile Retrospectives: Making Good Teams Great*, 1 edition. Raleigh, NC: Pragmatic Bookshelf, 2006.

[17] "Agile wallboards," *Projects' Little Helper*, 29-Oct-2010. [Online]. Available: http://www.projectslittlehelper.com/2010/10/29/being-agile/collaboration-and-communication/agile-wallboards/. [Accessed: 14-Apr-2017].

[18] T. KATAOKA, K. FURUTO, and T. MATSUMOTO, "The Analyzing Method of Root Causes for Software Problems," *SEI Tech. Rev.*, no. 73, p. 81, 2011.

[19] M. Sutton, "How to Make an App - an AAC app comes to life in a week," *Tactus Therapy Solutions*, 02-Feb-2015. .

[20] E. Derby, "Seven Ways to Revitalize Your Sprint Retrospectives | esther derby associates, inc.," 2010. [Online]. Available: http://www.estherderby.com/2010/06/seven-ways-to-revitalize-your-sprint-retrospectives.html. [Accessed: 14-Apr-2017].

[21] L. Gonçalves and B. Linders, *Getting Value out of Agile Retrospectives - A Toolbox of Retrospective Exercises*. lulu.com, 2014.

[22] "eBook: Fun retrospectives," *Caroli.org*. [Online]. Available: http://www.caroli.org/book-fun-retrospectives/. [Accessed: 14-Apr-2017].

[23] A. Przybylek and M. K. Olszewski, "Adopting collaborative games into Open Kanban," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2016, pp. 1539–1543.

[24] Tholons, "2015 Top 100 Outsourcing Destinations." www.THOLONS.com, 2015.

[25] A. Jenkinson, "Beyond segmentation," *J. Target. Meas. Anal. Mark.*, vol. 3, no. 1, pp. 60–72, 1994.