Manifesto [5] and results of previous research "Adopting to agile software development" [6].

Organizational domain is responsible for overall continuous improvement of the organization.

Subdomains:

- Communication – communication type, communication evaluation attributes, communication frequency attributes.
- Experience.
- Size.
- Learning – time for learning, financing for learning and certification.
- Team building process.
- Process change management − availability, responsibility, change implementation, evaluation.
- Goals.

Productivity domain is responsible for building increments of the product with a focus on the development team.

Subdomains:

- Communication – communication type, communication evaluation attributes, communication frequency attributes.
- Knowledge management – knowledge storage, knowledge sharing, knowledge sharing type, knowledge sharing frequency.
- Learning.
- Environment.
- Collaboration.
- Motivation.
- Team – team type, team size, team work organization;
- Experience – programming experience, experience in project, experience in technology.
- Practice change management – availability, responsibility, change implementation, evaluation, practices and practice set size, practice usage time.

Process domain is responsible for the effective use of Scrum [1] and its artefacts.

Subdomains:

- Time boxed activities − iteration planning, iteration, daily Scrum, retrospective, grooming, review meeting, release management.
- Roles − ScrumMaster, product owner, team.
- Artefacts − product backlog, release plan, release burndown chart, sprint backlog, product burndown chart.

Quality domain is responsible for the quality of the items delivered in Productivity domain.

Subdomains:

- Guidelines – availability, responsibility, evaluation, change implementation.
- Agreements – availability, evaluation.
- Standards – availability, responsibility, evaluation, change implementation.
- Tools – evaluation, financing for employees' tools, financing for organizations tools, time to learn the tool, tool implementation.
- Roles − DB architects, framework developers, UI developers, infrastructure developers, architects.
- Practice change management − responsibility, evaluation, availability, change implementation, practices, practice set size, practice usage time.

Value domain is responsible for increasing the value of the product releases.

Subdomains:

- Release management.
- Product management.
- Portfolio management – software portfolio, infrastructure portfolio, project portfolio.
- Metric change management – availability, responsibility, evaluation, change implementation, metric set size, metric usage time, metrics.

III. Agility Influence Index (AII)

Agility Influence Index is metric to evaluate each domain, subdomain and attribute (DSA). AII determines how a particular item (domain, subdomain or attribute) influences overall agility level of the organization. AII is measured in scale from 1 to 10 and is determined by the group of agile experts using method Delphi [2] or similar evaluation method where group of experts is involved. Sample AII values DSA are indicated in Fig. 2.

Such tree view of DSA is helpful; the organization is classified and its agility determined. Also tree view helps to identify the specific part of the organization where some adjustments are needed. To this point there are five domains, 130 subdomains and 548 attributes. As the purpose of this particular paper is not to list all AII values they will be published in a separate paper. Only two examples of attributes are presented in this paper for clarity.

One of the examples is "Amount of money for certification exams per person (EUR/Year)" and this attribute is described by values: $x = 0$; $1 < x < 100$; $100 < x < 200$; $200 < x < 300$; $300 < x < 400$; $x > 400$; Not known. This attribute describes the subdomain "Learning" from the domain "Productivity".

The second example is "Length of the Sprint in weeks" and this attribute is described by values: $x = 1$; $x = 2$; $x = 3$; $x = 4$; $x > 4$ [1]. This attribute describes subdomain "Time boxed activities" from the "Process" domain.

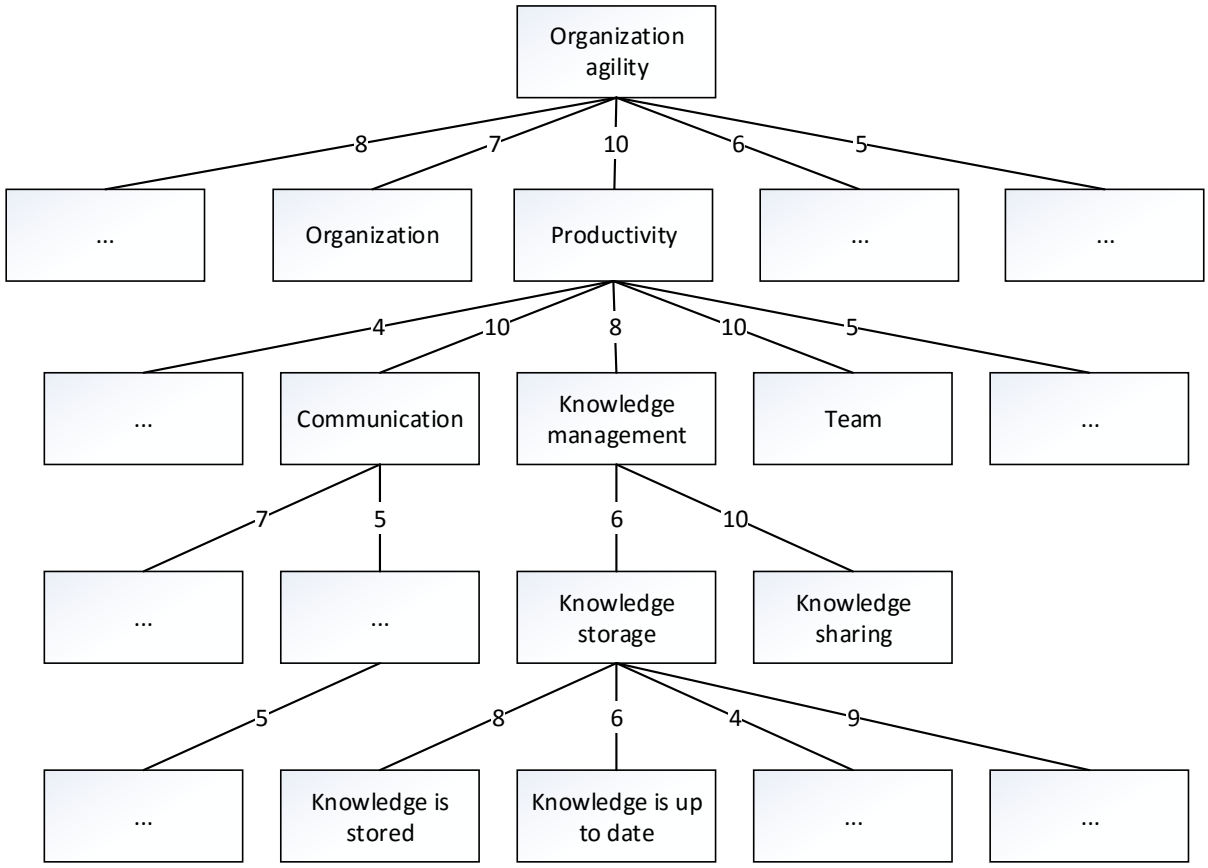

Fig. 2. Agility Influence Index sample evaluation values.

IV. ODA Method of Agility Evaluation

The purpose of the ODA method is to determine organizations agility level. If the organization is aware at which level of agility it stands, it can create "Change" or "Improvement" plan to bring the organization level up.

Mostly the problem is, that an organization does not or is not aware at which parts and levels of the organization changes should be implemented. This is the place for the ODA method to step in as it can help to find the problematic areas (Fig. 3.).

The process of the ODA method can be separated into two sub processes. The first process is responsible for getting AII for the DSA and this is the part of the complete research on how to adapt to agile software development. The AII values of the DSA will be published in a separate research paper, but the ODA method implies the possibility to reevaluate the AII values and modify DSA to improve organization agility level determination. As mentioned in the previous chapter AII is determined by Delphi method [2] or some alternative method. The first process where AII is determined must be done at least once, but can be repeated if necessary and mostly the reason for it is the change of DSA or outdated AII values. The second process of the ODA method is the assessment of the organizations agility on regular basis.

The organization or team decides when or at what intervals they would like to reevaluate organizations agility. It is important to remember that each team should be evaluated separately. It must be kept in mind that each team can be at a different level of agility and in case of tree teams only one of them can be agile. The results of each team's evaluation compose the result of the organization.

At the beginning of each evaluation iteration desired agility level is set. The next step is question generation. In the context of this research paper, the question generation is described very briefly just to indicate the idea about how questions are generated. Detailed question generation is described in a separate research paper. The main idea of question generation is that there are too many questions to be answered during one evaluation process. The purpose of the algorithm is to get only a set of questions. Previously saved AII values help to prioritize and obtain most important questions first. The amount of questions is configurable and depends on how many questions the organization is ready to ask their employees.

After the question generation is complete, question sets are sent to specific users. From the sent surveys information is gathered and then classification of organization takes place. In context of this research for data classification the FOIL algorithm [7] is used, but any other classification algorithm could also work, but needs to be tested separately. Based on the classification results reports are generated. Reports indicate the agility levels of DSA and help to create "Change" plan and improve parts and overall agility of the organization in case when the desired agility level is not achieved. The evaluation process can be repeated after some period of time.

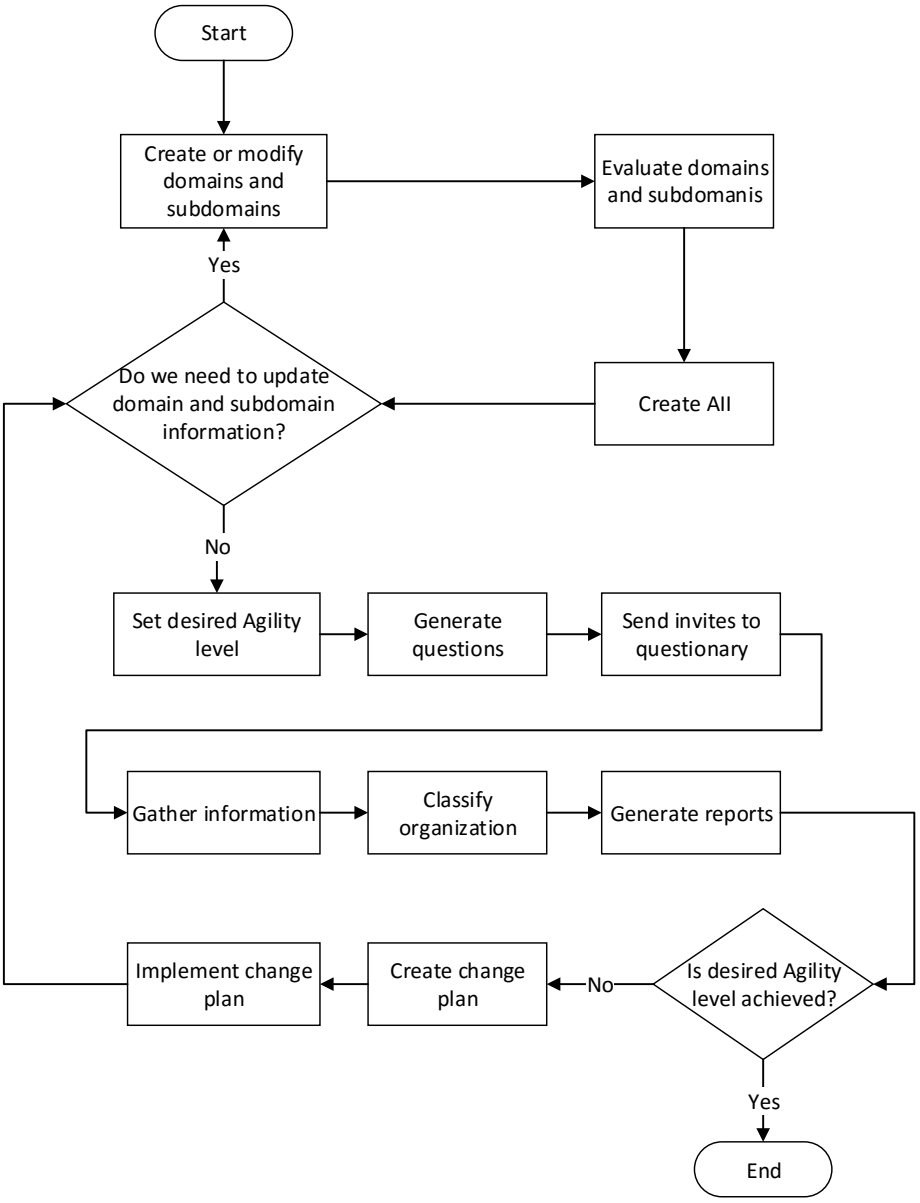

Fig. 3. Process of the ODA method.

V. Proposed Architecture for the Evaluation Tool

In the previous chapter the process of the ODA method was described. ODA must be used in conjunction with the supporting tool or tool set. In this chapter the proposed architecture of the tool needed for the method is described (Fig. 4.). The ODA method for the process domain is using Scrum, but processes from some other agile method could also be used. The Scrum tends to be one of the most popular agile methods today [6]. As the ODA method is using Scrum as agile method it includes some Scrum terminology which is briefly described below:

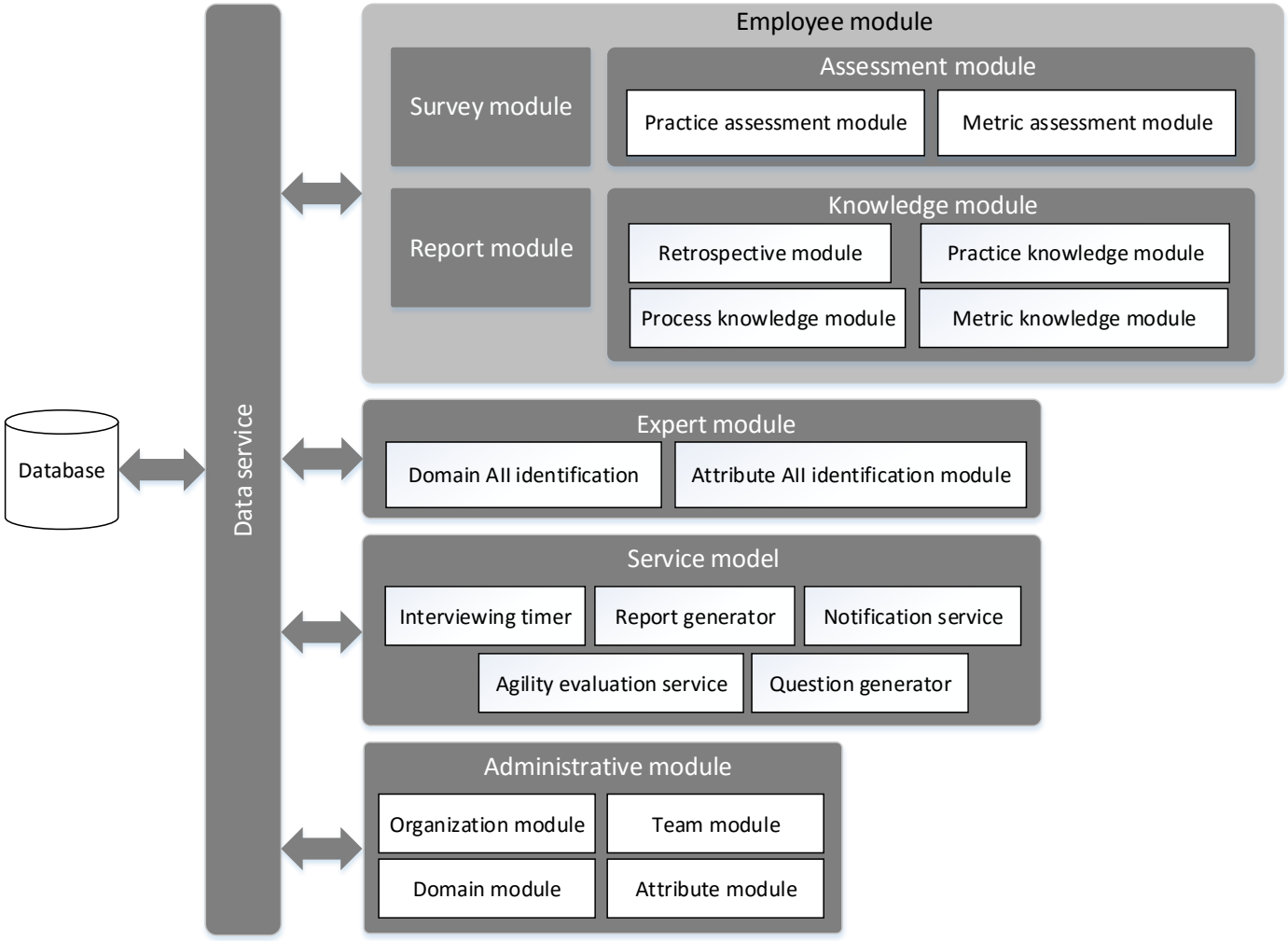

Fig. 4. Architecture of the evaluation tool.

- Retrospective – during the meeting after each iteration problems and successes are discussed among the team.
- Practice – repetition of an activity to improve skills, for example, Test Driven Development, Pair Programming etc.
- Process –a set of actions in particular order to achieve results.

The architecture consists of four main modules:

- Employee module –focused on employee submodules and functions:
  - Survey module – generated questions are accessible in this module.
  - Report module – contains functionality for viewing and initiating report generation.
  - Knowledge module – used to store user knowledge and consists of 4 submodules:
    - Retrospective module – gathers information discussed during retrospectives (positive and negative feedback);
    - Practice knowledge module – information about the used or recommended practices is saved and reviewed.
    - Process knowledge module – information about the used or recommended processes is saved and reviewed.
    - Metric knowledge module – information about the used or recommended metrics is saved and reviewed.
  - Assessment module – focused on assessing agile practices and metrics:
    - Practice assessment module.
    - Metric assessment module.

- Service module – services give opportunity to run the jobs in particular scheduled intervals:
  - Interviewing timer – responsible for the initiation of the sending process of the generated question to the employee.
  - Report generator – as report generation can be time consuming process it is recommended to generate information for the reports in the background, so no long waiting is involved, then requesting the report.
  - Notification service – the responsibility of the notification service is to send notifications to users and experts.
  - Question generator – the service is responsible for the generation of the set of questions and is executed manually or according to a particular schedule.
  - Agility evaluation service.

- Expert module – responsible for AII value identification for DSA:
  - Domain AII identification module.
  - Attribute AII identification module.

- Administrative module – responsible for working with globally scoped data and consists of 4 submodules:
  - Organization module – handles organizations level data.
  - Team module – handling of the team level data, including the list of team members.
  - Domain module – handling of domain and subdomain related data. There is a default domain and subdomain list proposed by the ODA method, but the user can also modify the configuration of the domains if necessary. In case of changes in domain and/or subdomain structure it has to be taken into account that revaluation of domain and subdomain AII will be required and expert network should be created for this purpose.
  - Attribute module – attributes are connected with subdomains and depend on particular subdomain configuration. The ODA method offers a list of attributes for each subdomain, but in the case of specific organization structure and processes the list of attributes can be modified.

- Data service – data access layer to save and retrieve data from the storage.

VI. Conclusion

The evaluation of SDC agility is not simple as software development process itself is quite complex. It is reasonable to split the problem in smaller components to handle each component separately. In this research decomposition of SDC in Organization, Quality, Value, Productivity, and Process domains is proposed. As each domain is also quite complex it is decomposed further to subdomains and in some cases it is decomposed further on. The subdomains are described by the sets of the attributes. All domains, subdomains and attributes (DSA) are evaluated by the group of experts.

From expert evaluations Agility Influence Index (AII) is created. AII defines how a particular element of DSA influences the agility of the organization. In context of this research evaluation method Delphi is used [2], but also some other methods could be used to identify AII. To this point there are five domains, 130 subdomains and 548 attributes and their AII values will be published in a separate research paper.

For the agility evaluation some systematic approach is needed. The problem is approached by using the ODA method. The ODA method helps to improve organizations agility by evaluating DSA of particular organization. The process of the ODA is divided into two sub processes. The first process is the evaluation of AII values of the DSA and this process needs to be done at least once. The second process is the iterative assessment of the organization domains by the employees from different teams and management. One of the main components in the process is the question generator, which generates questions by taking into account AII of DSA. This approach helps to decrease the amount of questions to be answered during each assessment iteration. The details of the question generator are a part of separate research and will be published together with AII values of DSA as purpose of this research is to focus on the process of the ODA method. AII values of DSA in conjunction with the information gathered from the employee surveys will be used by the FOIL method [7] to generate the rules and determine organizations agility level according to the agility classification.

The architecture developed for the tool consists of four main modules and the database: Employee module, Expert module, Service Module and Administrative module. The architecture concept is used to build the evaluation tool and test the effectiveness of the ODA method and "Organization Agility Model".

This research is a part of broader research and as the next step it is planned to build the tool to satisfy the requirements of the ODA method and to perform initial evaluation of DSA involving agile expert network [2].

References

[1] K. S. Rubin, *Essential Scrum: A Practical Guide to Most Popular Agile Process*. Boston, MA: Addision-Wesley Professional, 2012.

[2] G. J. Skulmoski, F. T. Hartman. J. Krahn, "*The Delphi Method for Graduate Research," in Journal of Information Technology Education*. Santa Rosa, CA: Informing Science Institute, 2007.

[3] M. Poppendieck and T. Poppendieck, *Implementing Lean Software Development: From Concept to Cash*. Boston, MA: Addision-Wesley Professional, 2006.

[4] A. Stellman and J. Greene, *Understanding Scrum, XP, Lean, and Kanban*. Boston, MA: Addision-Wesley Professional, 2014.

[5] K. Beck, M. Beedle, R. C. Martin et al. (2001, February) *Manifesto for Agile Software Development*, [Online]. Available: http://agilemanifesto.org/

[6] G. Linkevics, "Adopting to agile software development," in *Scientific J. of Applied Computer Systems.* Riga, LV: Riga Technical University, 2014, pp. 64–70.

[7] T. M. Mitchell, "Learning Sets of Rules," in *Machine Learning.* New York: McGraw-Hill, 1997, pp. 283–291.

Gusts Linkevics received the Engineer Degree in Computer Sciences, in 2003 and the Master Degree in Business Administration, from Riga Technical University in 2005.

He is a Senior Developer with ZZ Dats Ltd. (Elizabetes street 41/43), If P&C Insurance Company Ltd., Nexum Insurance Technologies, and a Lead Developer with Tilde Ltd.

He is a member of Latvian .NET user group, a member of Microsoft Student Partners. He is a Microsoft Certified Solution Developer, Microsoft Certified Application Developer, Microsoft Certified Professional.

E-mail: gusts.linkevics@rtu.lv, Phone: +371 29417442

Uldis Sukovskis is professor of Riga Technical University (Kalku Street 1). He is corresponding member of Latvian Academy of Science since 2008. He is Vice-Rector for Academic Affairs and Head of the Department of Applied Computer Science at Riga Technical University. He has work experience in company Exigen Services Latvia in the position of Director for IT Consulting and Audit. Professor Sukovskis is member of Information Systems Audit and Control Association (ISACA) and is Certified Information Systems Auditor (CISA).

E-mail: uldis.sukovskis@rtu.lv, Phone: +371 67089303