Application of Methodology Evaluation System on Current IS Development Methodologies

Alena Buchalcevova
Prague University of Economics, Czech Republic

ABSTRACT

This paper focuses on IS development methodologies, especially on new agile scaling frameworks. A significant need for large-scale agile is presented together with the evidence of its usage in companies. The aim of this paper is to examine selected agile scaling methodologies and frameworks, and compare them to each other and with other IS development methodologies. To do so, the Methodology Evaluation System METES is utilized for the assessment and comparison. Assessed IS development methodologies can be then used by companies for the selection of the appropriate methodology. Furthermore, presenting the results of the assessment especially in the graphical form supports better understanding of these methodologies.

Keywords: IS Development, Methodology, Scaled Agile, Evaluation, Selection, System, Criteria, Comparison

INTRODUCTION

Although software systems play a key role in today's society, the process of their development cannot be regarded as satisfactory. According to the recent Standish Group's CHAOS Report (The Standish Group, 2015), only 29% of all application development projects did satisfy the criteria of successfulness (on time, on budget and with satisfactory results). Even though the success rate of agile IT projects (IT project managed by an agile methodology) is higher (39%), numerous projects are facing challenges. The Standish Group analyses also the factors that are key to a project success. Among 10 key success factors are also the Standard Architectural Management Environment (practices, services, and products for developing, implementing, and operating software applications) and Agile Process. Confirming that software processes are an integral part of a software project success with a significant need for an improvement. This fact is demonstrated through the Software Process Improvement (SPI) initiative. We recognize two different SPI approaches. The first one is represented by the usage of international standards like ISO/IEC 12207, ISO/IEC 15504, ISO/IEC 29110 or CMMI (Capability Maturity Model Integration) whereas the other one promotes an advancement of IS development (ISD) methodologies, especially agile methodologies and a development of brand new scaled agile methodologies and frameworks. As the number of existing IS development methodologies and approaches increases, selecting the appropriate methodology becomes a challenge.

To support companies in their methodology selection and customization process, the Methodology Evaluation System METES was defined in 2009 (Buchalcevova, 2009), based on an analysis of existing systems for ISD methodologies selection. In a comparison to other systems for ISD

methodology evaluation and selection, the METES system enriches the methodology selection process by introducing the Support criteria group. This group evaluates the availability and accessibility of the methodology and other factors influencing the implementation of the methodology.

The METES was validated in 2009 by its utilization as an assessment system for selected ISD methodologies. For the assessment, the most utilized methodologies at that time, both prescriptive and agile, were selected, i.e. Rational Unified Process (RUP), OpenUP, Feature Driven Development (FDD), Scrum, Extreme programming (XP) and MSF for CMMI development. Detailed results of this assessment were published in (Buchalcevova, 2009) and selected results then in (Buchalcevova, 2011).

The results of conducted assessments of ISD methodologies are stored and are available to companies for the selection of an appropriate methodology for a concrete project. Moreover, the METES has an educational potential as it represents a conceptual tool for understanding and mutual comparison of ISD methodologies. In this way it is used within Software Engineering and Software Process Improvement university courses at the Prague University of Economics, both at the undergraduate as well as at the graduate level.

With the aim to reflect the evolution of ISD methodologies over time, a brand new assessment was conducted in 2017. Selected ISD methodologies were reassessed with the aim to address their evolution and changes. In addition, newly emerged methodologies such as Kanban, Discipline Agile Delivery (DAD), Large-scale Scrum (LeSS) and Scaled Agile Framework (SAFe) were assessed.

The aim of this paper is to present the results of the assessment of new scaled agile methodologies and frameworks. The rest of the paper is organized as follows. First, the research methodology is presented followed by the evolution of agile methodologies. Then, the Methodology Evaluation System METES is introduced and all the evaluation criteria are presented. Next section describes selected results of the assessment of ISD methodologies conducted in 2017 with a focus on scaled agile methodologies and frameworks. Finally, concluding remarks are discussed.

RESEARCH METHODOLOGY

The objective of this research is to examine newly emerged ISD methodologies, especially scaled agile methodologies and frameworks.

The research question is defined as follows: What are the differences in characteristics and implementation support of individual scaled agile methodologies and how these characteristics differ from other ISD methodologies?

To answer the research question, the Methodology Evaluation System METES was utilized. The METES was developed by the author in 2009 using Design Science Research (Hevner et al., 2004; Gregor & Hevner, 2013), specifically Design Science Research methodology (Peffers et al.,2008). The validation of the METES system was performed in 2009 by the assessment of the most utilized methodologies at that time. To address the evolution of ISD methodologies and emergence of new methodologies, further assessment was needed, performed then in 2017. The sample of ISD methodologies selected for this assessment was defined according to the following rules:

- All previously assessed methodologies, except of MSF for CMMI development that is of low utilisation, were selected.

- Newly emerged agile methodologies were selected based on a level of their usage reported in the State of Agile Survey (VersionOne, 2016).

Based on these rules, the final sample of ISD methodologies was as follows: Rational Unified Process (RUP), OpenUP, Feature Driven Development (FDD), Scrum, Extreme programming (XP), Kanban, Discipline Agile Delivery (DAD), Large-scale Scrum (LeSS) and Scaled Agile Framework (SAFe).

The assessment was performed in the following way. Groups of 3 to 4 students of the graduate course Software Process Improvement within the Information Systems curriculum were established and asked to choose an ISD methodology (from the sample) for an assessment. The groups followed a detailed description of the METES including all criteria and scale definition as well as all previous assessments. In the timeframe of 6 weeks, each group conducted the assessment according to the procedure defined in the METES. Criteria values were assigned based on thorough research of current publication sources examining the methodology. Students focused on studying the methodology and performed an extensive research on teaching, training, certifying and supporting the implementation of the methodology. Each group prepared a detailed documentation explaining the rationale of each criteria value assigned including graphs visualizing the assessment results. Several workshops followed arranged with all students of the course and the author of this paper, lecturer of the course and an expert with long-lasting experience with ISD methodologies. Each group presented the results of their assessment and their rationale behind it. In case of an ISD methodology reassessment, the differences from previous assessment were highlighted and explained. During the discussion, assigned criteria values were either confirmed or marked for further exploration. Results of it were again confirmed by the lecturer.

Once ISD methodologies within the sample were assessed (i.e. all criteria values were determined), a comparison focused on scaled agile methodologies was performed by the author based on the METES criteria. The results of the comparison are presented in a graphical representation prepared using MS Excel.

EVOLUTION OF AGILE DEVELOPMENT

Today's digital adapt-or-die environment demands rapid changes in the way of creating and delivering value to customers. Both an effective utilization of software systems and their development are key factors for mastering this ability. That is why the development methodologies must enable innovation, collaboration, and speed. The traditional waterfall methodologies do not scale to such challenges, and thus more responsive development methods are needed. An agile approach is a major step in that direction. However, agile methods were originally designed for a usage in small, single team projects (Boehm & Turner, 2005). Nevertheless, their benefits have made them attractive also for larger projects and larger companies (Dikert, Paasivaara & Lassenius, 2016) even despite a more difficult implementation within larger projects (Dybå & Dingsøyr, 2009). Compared to small projects, being ideal for agile development, larger ones are characterized by the need for an additional coordination. Large-scale agile involves additional concerns in handling an inter-team coordination and interfacing with other organizational units, such as human resources, marketing and sales, and product management. In addition, large scale may cause users and other stakeholders to become distant from the development teams (Dikert, Paasivaara & Lassenius, 2016). Despite such known problems related to large-scale agile, there is an industry trend towards adopting agile methodologies in-the-large (VersionOne, Inc, 2016; Dingsøyr & Moe, 2014).

The State of Agile Survey conducted annually since 2007 by Version One has recently researched large-scale agile as well. The results indicate that there exist a significant number of companies that have already taken or are taking agile into use in large-scale settings (Dingsøyr & Moe, 2014).

A number of scaled agile methodologies and frameworks are in place like the Discipline Agile Delivery (DAD), Large-scale Scrum (LeSS), Scaled Agile Framework (SAFe), Scrum@Scale, and Nexus. In the following sections, the SAFe, LeSS and DAD methodologies are described in more detail as they are further evaluated within the METES system.

Scaled Agile Framework

The Scaled Agile Framework (SAFe) is a freely revealed knowledge base of proven, integrated patterns for enterprise-scale Lean-Agile development (Scaled Agile, 2016). The SAFe was created by Dean Leffingwell in 2012 and since then it has continually evolved to a current 4.0 version. The SAFe website (SAFe, 2017) provides a guidance for scaling agile development across the Portfolio, Value Stream, Program, and Team levels that are part of the Big Picture, i.e. a visual overview of the Framework. The Framework is scalable and modular, allowing each organization to adapt it to its own business model. The Framework has four core values that help to make the SAFe effective: Alignment, Built-in Quality, Transparency, and Program Execution. The SAFe's practices are grounded on nine fundamental principles that have evolved from agile principles and methods, Lean product development, systems thinking, and observation of successful enterprises. The heart of the SAFe is the Program level, which revolves around an organization called the Agile Release Train (ART). Each ART aligns teams to a common mission and vision via a single program backlog and produces valuable and evaluable system-level solutions every two weeks. The Agile teams in an ART have the following choice of methods: Scrum, Kanban, and XP. They also use built-in quality practices. Each SAFe portfolio has the value streams, people, and processes necessary to provide Lean-Agile funding and governance for the products, services, and solutions required to fulfil its business strategy (Scaled Agile, 2016).

Large-scale Scrum

The Large-scale Scrum (LeSS) framework was created by Bas Vodde and Craig Larman in 2013 based on their experiences working with large-scale product development. As both authors state in (Larman &Vodde, 2013) *scaling Scrum starts with understanding and being able to adopt standard real one-team Scrum*. Large-scale Scrum requires examining the purpose of single-team Scrum elements and figuring out how to reach the same purpose while staying within the constraints of the standard Scrum rules. The LeSS provides two different large-scale Scrum frameworks (LeSS framework, 2017), i.e. the basic LeSS applicable up to eight teams (of eight people each) and the LeSS Huge that introduces additional scaling elements for development up to hundreds of developers.

Disciplined Agile Delivery

The Disciplined Agile Delivery (DAD) framework is a hybrid of existing methods such as Scrum, Kanban, Agile Modelling, SAFe, Extreme Programming, Agile Data, Unified Process and many others. The DAD provides the flexibility to use various approaches and plugs the gaps not addressed by mainstream agile methods (Ambler & Lines, 2011). The main characteristics of this framework are that it: is a people first, learning oriented hybrid agile/lean approach; has a risk value delivery lifecycle; is goal-driven; is enterprise aware; is tactically scalable at the team level; and strategically scalable across all of the enterprise (Disciplined Agile 2.X, 2017).

Although a significant number of studies have analysed the impact of implementing agile techniques in smaller team settings, there is a very little reporting done on how agile development methods can be

implemented at the team level and scaled up at the program/portfolio level in large software organizations (Kataria, 2016).

In (Kataria, 2016), the results of an empirical study focused on the penetration of agile development at Pearson Education, a large enterprise with hundreds of employees, are presented. Pearson Education used the Product Creation Framework (Pearson Education, 2016) which is based on the Scaled Agile Framework (SAFe) (Scaled Agile, 2016) for implementing agile practices at the program and portfolio level. The study showed that about 90% of the respondents used agile development and 87% of them worked at the team level. The analysis also indicated that among the respondents using the non-agile methods, 83% wished to switch to agile methods. Surprisingly, agile practices were followed more rigorously in larger teams. The respondents experienced only in working with agile methods practiced agile techniques more rigorously and perceived them more positively. Also, when received training in agile methods, the respondents were significantly more inclined to adhere to the process and had an overwhelmingly positive opinion. The study also showed that there is a way of scaling up agile methods successfully from the team/project level to the program level by following a disciplined approach. Teams and programs have dependencies, thus a better synchronization and coordination can be achieved if the agile methods are implemented across all the teams and programs. Training human resources, defining and rigorously practicing agile techniques at the program and project level and reducing dependencies are key factors in the success of scaling agile methodologies (Kataria, 2016).

Further study (Amro, 2014) identified challenges faced upon using agile in a global distributed environment project. Among these challenges, a wrong selection of an appropriate agile method is described as one of the most common issues. To support a right and effective selection of the appropriate methodology, the Methodology Evaluation System METES presented in the next section can be utilized as a problem solver.

METHODOLOGY EVALUATION SYSTEM METES

The Methodology Evaluation System METES was developed by the author in 2009 and is described in (Buchalcevova, 2009; 2011). Prior to developing the METES, existing systems focused on ISD methodology selection had been analysed, i.e. the Methodology Framework for IS/ICT – MeFIS (Buchalcevova, 2004), the System and Method for Software Methodology Evaluation and Selection (Hecksel, 2007), Boehm and Turner's system (Boehm & Turner, 2004) and its widening proposed by Taylor (Taylor, 2006). As none of the analysed systems met all the requirements, especially not taking into consideration the availability and accessibility of the ISD methodology, its implementation support, training, certification and localization, a new methodology evaluation system, the METES, was developed to fulfil such gaps.

The METES comprises both an assessment system and a basis of assessed ISD methodologies to choose from. Figure 1 shows the conceptual model of the METES in the form of UML 2.0 class diagram.

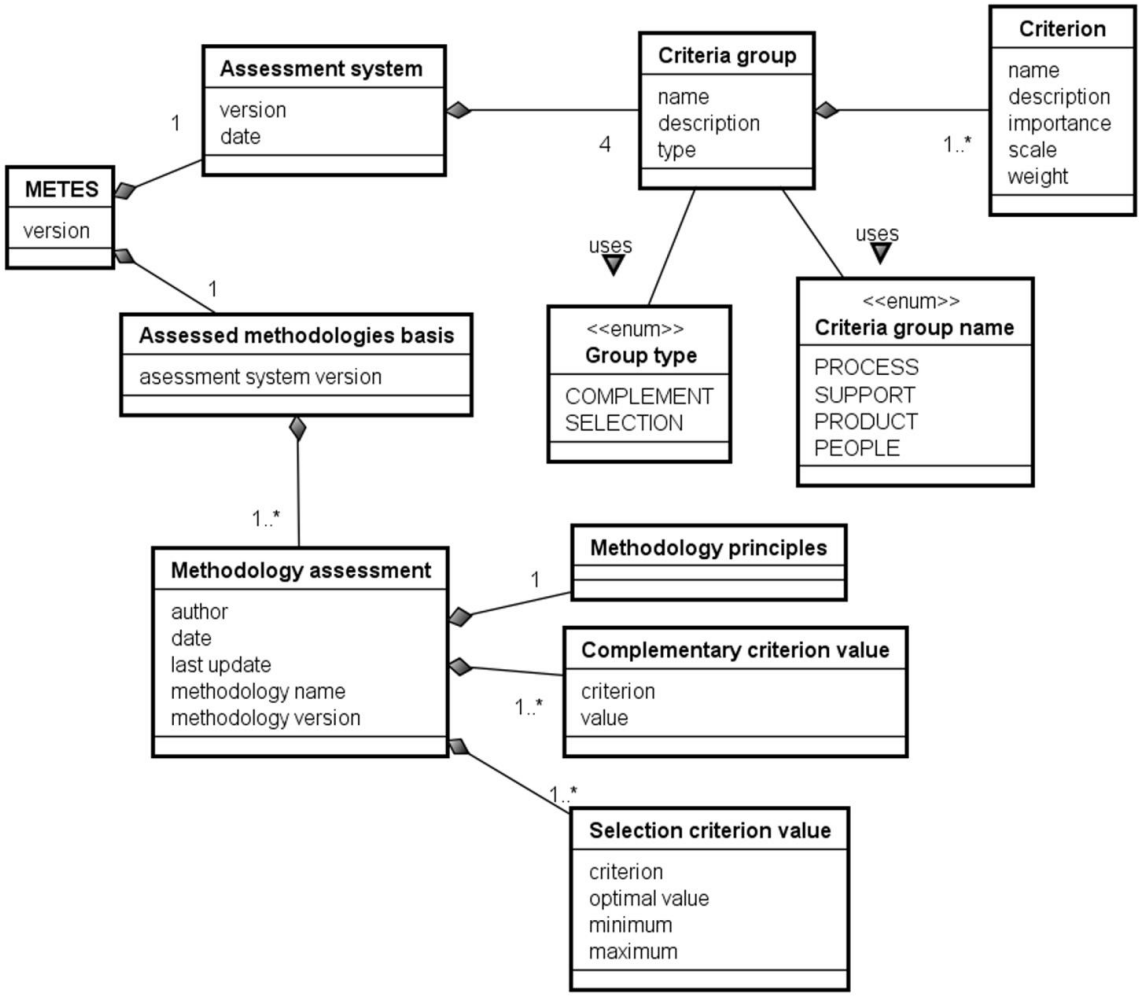

Figure 1 Conceptual model of METES system

The structure of the assessment system is captured in Figure 2. For each ISD methodology, the evaluation criteria are assessed clustered into 4 groups – Process, Support, Product and People. The criteria in the Process group represent process features of the methodology, e.g. scope of software life cycle processes, life cycle process model, roles, metrics, type of development, etc. The criteria included in the Support group assess availability of the methodology, support of the implementation and customization of the methodology, availability of skilled people, etc. The Product group criteria evaluate the built solution whereas the People group criteria describe features of the development team.

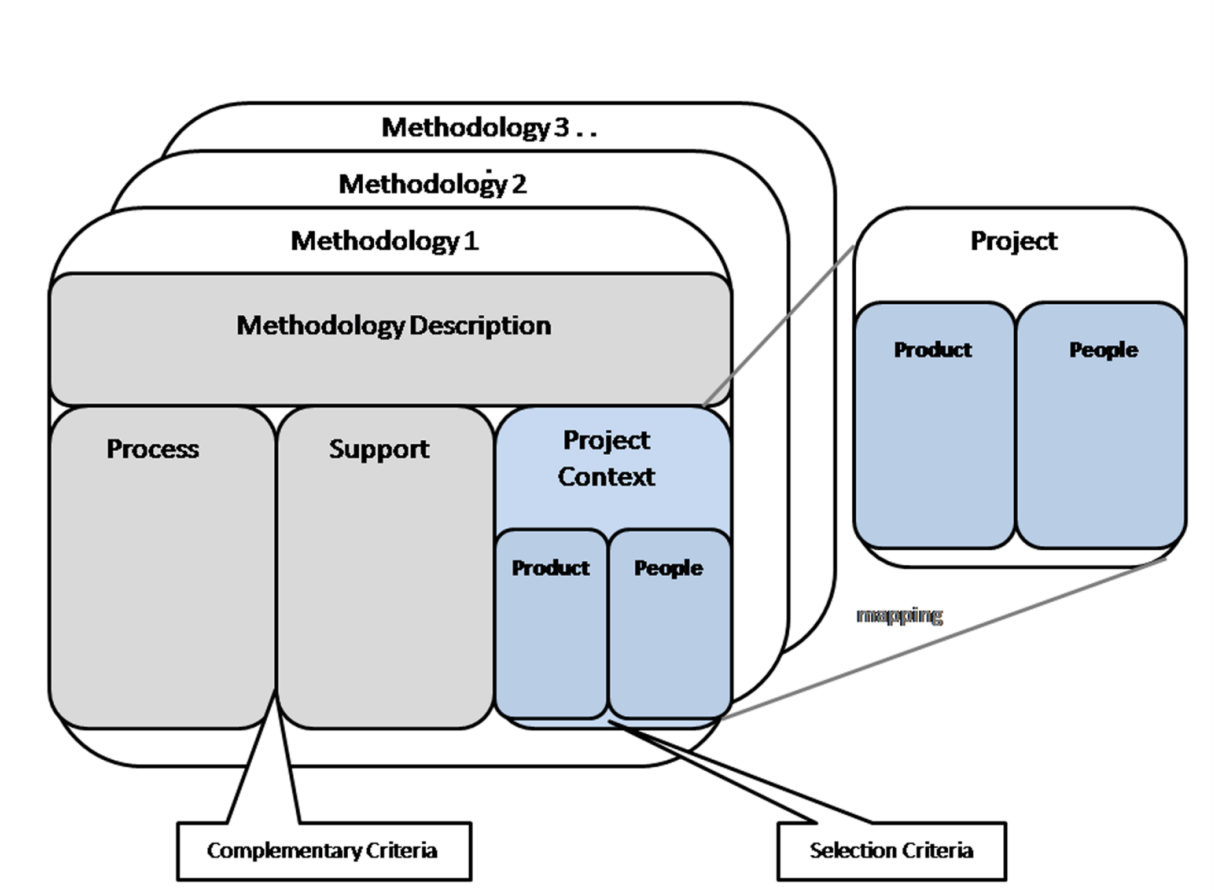

Figure 2 Structure of METES system

The Product and People criteria groups represent the project context. In the methodology selection process their values are assessed for the project and then compared to the criteria values (minimal and maximal) of individual ISD methodologies. These criteria are therefore named as selection criteria. Among the selection criteria, 5 key criteria being the most critical for the methodology selection were identified. The key selection criteria, described in Table 1, are similar to those defined by Boehm & Turner (2004). However, the Project duration criterion is inserted in addition. The criteria in the Process and Support groups compose the complementary criteria that are used as an additional tool in the selection process.

For each criterion, a scale from 0 to 5 is defined along with a detailed meaning of the values on the scale. A detailed description of all criteria and their role in the METES is included in (Buchalcevova, 2009). As it exceeds the scope of this paper, only a short explanation is presented in Tables 1 - 4. Table 1 depicts key selection criteria. The remaining criteria in the Product and People groups represent other selection criteria and are described in Table 2. The criteria in the Process (see Table 3) and Support (see Table 4) groups are used just as a supporting tool for the ISD methodology selection.

Table 1. Key Selection Criteria

Key Selection Criteria			
Group	**Criterion**	**Description**	**Scale**
Product	Product criticality	Criticality of the designed system	0: pilot project 1: system just for entertainment 2: mission supported system

			3: mission critical system at the national level 4: mission critical system at the global level 5: life critical system
Product	Project duration	Duration of the project measured in months	0: less than month 1: less than 3 months 2: less than 6 months 3: less than 12 months 4: less than 24 months 5: more than 24 months
People	User accessibility	How accessible the user is	0: user is a part of the team and is responsible for requirements specification 1: user is accessible on daily basis 2: user is accessible at any time when asked 3: user is accessible at the beginning, at the end of the project and at defined milestones 4: user is accessible at the beginning and at the end of the project 5: user is not accessible during the project
People	Team size	Size of the team measured by number of its members	0: 1 - 4 1: 5 - 10 2: 11 - 20 3: 21 - 50 4: 51 - 100 5: more than 100
People	Distribution	Distribution of the team	0: located in one room 1: located in one building 2: more locations in one city 3: two locations in one country 4: more locations in one country 5: more countries

Table 2. Other Selection Criteria

Other Selection Criteria			
Group	**Criterion**	**Description**	**Scale**
Product	Requirements stability	Extent of requirements changes during the project	0: requirements cannot be defined ahead 1: more than 50% of requirements are changed 2: less than 30% of requirements are changed 3: requirements are defined ahead, do not change, only priorities are changed 4: requirements are defined ahead, changes occur, but are not preferable 5: none or few changes of requirements
Product	Reuse	Extent of usage or development of reusable artefacts within the project	0: non-targeted reuse 1: usage of developed components 2: project focused on building reusable classes 3: reusable components built within the

			project 4: reusable components built for usage within the enterprise 5: project focused on building reusable components at the enterprise level
Product	Solution size	Size of the solution (system) measured by number of use cases	0: less than 10 use cases 1: 11 - 40 use cases 2: 41 - 100 use cases 3: 101 - 200 use cases 4: 201 - 300 use cases 5: more than 300 use cases
People	Lack of project manager experience	Experience of project manager the methodology requires	0: more than 5 years of project manager experience 1: 4 - 5 years of project manager experience 2: 3 - 4 years of project manager experience 3: 2 - 3 years of project manager experience 4: 1 - 2 years of project manager experience 5: less than one year of project manager experience
People	Lack of team member qualification	Level of team members qualification the methodology supposes	0: more than 70% of team members with good qualification, generalists 1: more than 70% of team members with good qualification, but specialists 2: about 50% of team members with low qualification, 3: about 60% of team members with low qualification 4: about 70% of team members with low qualification 5: about 80% of team members with low qualification
People	Lack of team member motivation	How motivated team members are or should be	0: very motivated staff with high moral values 1: active, motivated staff, share knowledge 2: staff fulfil tasks, share knowledge 3: staff fulfil tasks, do not share knowledge 4: low motivated staff, do not share knowledge 5: low or none motivation

Table 3. Complementary Criteria Process Group

Complementary Criteria Process Group			
Group	**Criterion**	**Description**	**Scale**
Process	Scope	Number of software life cycle processes covered by the methodology, assessment is based on mapping the methodology processes to the Process Reference Model defined in the	0: SW implementation processes 1: SW implementation and SW support processes 2: project management processes 3: SW implementation, SW support

Complementary Criteria Process Group			
Group	**Criterion**	**Description**	**Scale**
		ISO/IEC 12207 international standard	and project management processes 4: systems and software processes including project management 5: high level of PRM 12207 process coverage
Process	Life cycle model	Life cycle model used by the methodology	0: none 1: waterfall model 2: V-model 3: spiral model 4: iterative model combined with sequential model 5: iterative model with iteration length less than one month
Process	Role	Number of roles the methodology deals with	0: one role 1: 2 – 5 software engineering roles at the project level 2: software engineering and management roles at the project level, less than 10 roles 3: software engineering and management roles at the project level, more than 10 roles 4: software engineering and management roles at the project level, management roles at the enterprise level 5: both software engineering and management roles at the project and enterprise level
Process	Process description particularity	Level of detail in the process description	0: none process description 1: process goals and responsible role are defined 2: process goals, responsible role and metrics are defined 3: process goals, responsible role, metrics and output are defined 4: inputs, outputs, roles and tasks are defined 5: complete process description, inputs, outputs, roles and tasks performed by roles are defined
Process	Documentation	Amount of documentation the methodology requires	0: none documentation required 1: minimal documentation (agile) 2: only software requirements specification, 3: more than 15 documents 4: more than 30 documents 5: more than 50 documents
Process	Metrics	How the methodology uses metrics and how important they are	0: no metrics used 1: some metrics are defined, low importance 2: some metrics are defined, high

Complementary Criteria Process Group			
Group	**Criterion**	**Description**	**Scale**
			importance 3: systems of metrics defined, tool support 4: complete systems of metrics, methodology is dependent on them 5: complete systems of metrics used for process improvement
Process	Quality management	Level of testing and quality management incorporated in the methodology	0: no quality management 1: only acceptance testing 2: software testing 3: software and systems testing 4: quality standards and quality measures are defined 5: quality management within the whole life cycle, tool supported

Table 4. Complementary Criteria Support Group

Complementary Criteria Support Group			
Group	**Criterion**	**Description**	**Scale**
Support	Integrity of resources	How accessible the methodology is	0: no resources 1: methodology is described in papers 2: methodology is described in various sources, papers, blogs, websites 3: methodology is described in one source 4: methodology is described in one source and also other sources are available 5: methodology is delivered as an application with a content management tool
Support	Availability	Availability of the methodology	0: no public availability 1: available in commercial publications 2: commercial product 3: freely available 4 open source license 5: open source license with a content management tool
Support	SW tools support	Availability of content management tools for methodology publication and customisation	0: no tools for methodology administration 1: methodology administration tools from third parties 2: --- 3: commercial methodology administration tools 4: --- 5: methodology administration tools delivered with the methodology

Complementary Criteria Support Group			
Group	**Criterion**	**Description**	**Scale**
Support	Methodology implementation support	Level of support for methodology implementation, e.g. consultation, training, methodology configuration	0: no support 1: consultancy by distributor 2: external consultants for methodology implementation 3: methodology configuration 4 methodology configuration and training 5: complete methodology implementation
Support	Methodology customisation	How the methodology deals with customisation	0: no customisation 1: customisation is possible at the beginning of the project 2: customisation is recommended at the beginning of the project 3: customisation is possible during the project (in each iteration) 4: customisation is recommended during the project (in each iteration) 5: customisation is supported by software tools
Support	University courses	Whether the methodology is being taught at universities	0: no university courses 1: some methodology technics are taught 2: methodology is taught at universities in the world theoretically 3: methodology is taught at universities in the world practically 4: methodology is taught at universities in the Czech Republic theoretically 5: methodology is taught at universities in the Czech Republic practically
Support	Training and certification	Availability of training and certification either worldwide or in the Czech Republic	0: no training and certification 1: training worldwide 2: training in the Czech Republic 3: training and certification worldwide 4: training and certification in Europe 5: training and certification in the Czech Republic
Support	Localisation	Availability of the methodology in the Czech language	0: no localisation 1: --- 2: --- 3: partial localisation 4 --- 5: complete localisation

Methodology Selection Process

The METES defines the ISD methodology selection method as one of the multi-criteria analysis methods. Therefore, the criteria weights are defined. The process of the ISD methodology selection is

divided into two steps. In the first step, the ISD methodologies applicable for the project are selected. The applicable ISD methodologies are those for which each project key selection criterion value (product criticality, project duration, user accessibility, team size, distribution) ranges in between minimal and maximal values defined for the methodology criterion. In the second step, one or more recommended methodologies are selected based on the complementary criteria.

EVALUATION OF SELECTED ISD METHODOLOGIES

In this section, particular results of the brand new 2017 assessment of selected ISD methodologies are presented and also the research question is addressed. The results of the assessment are presented in a tabular and graphical form. In Table 5, the optimal values of the selection criteria, i.e. criteria in the Product and People group are depicted.

Table 5. Evaluation of ISD Methodologies - Product and People Group Criteria, optimal values

	RUP	**OpenUP**	**FDD**	**Scrum**	**XP**	**DAD**	**Less**	**SAFe**	**Kanban**
Product criticality	5	2	3	3	4	4	4	4	2
Project duration	4	2	3	3	4	4	5	3	3
Requirements stability	2	1	1	0	0	2	1	1	0
Reuse	3	2	1	1	1	4	0	1	0
Solution size	5	2	4	5	3	4	5	5	3
Lack of project manager experience	3	3	3	1	2	2	1	1	1
Lack of team member qualification	4	5	2	0	1	1	1	1	1
Lack of team member motivation	4	1	2	0	0	1	1	1	1
User accessibility	3	3	1	0	0	2	2	2	2
Team size	5	1	3	1	2	3	5	5	1
Distribution	5	1	1	3	0	3	5	5	1

Table 6 shows the values of the Process group criteria for all assessed methodologies whereas Table 7 presents the values of the Support group criteria.

Table 6. Evaluation of ISD Methodologies - Process Group Criteria

	RUP	**OpenUP**	**FDD**	**Scrum**	**XP**	**DAD**	**Less**	**SAFe**	**Kanban**
Scope	4	3	3	2	1	3	2	4	2
Life cycle model	4	5	5	5	5	5	5	5	0
Role	3	3	3	2	2	3	5	5	0
Process description particularity	5	5	3	0	2	0	0	1	0
Documentation	5	4	3	1	1	1	1	2	0
Metrics	3	1	3	1	1	2	2	3	2
Quality management	5	2	3	1	5	3	2	3	1

Table 7. Evaluation of ISD Methodologies - Support Group Criteria

	RUP	**OpenUP**	**FDD**	**Scrum**	**XP**	**DAD**	**Less**	**SAFe**	**Kanban**

Integrity of resources	5	5	2	2	4	4	2	3	2
Availability	2	5	1	4	3	1	3	2	3
SW tools support	3	5	0	4	1	1	1	4	3
Methodology implementation support	5	0	2	5	2	1	1	2	0
Methodology customisation	5	5	3	4	4	4	3	4	3
University courses	5	5	4	5	5	1	4	1	4
Training and certification	5	4	4	5	5	3	5	3	4
Localisation	0	3	0	5	0	0	0	0	3

The results of the evaluation are more demonstrative when graphical form is used. The graph can be developed either simply for individual methodologies or altogether in order to compare several methodologies. This case is presented in Figures 3, 4 and 5 where the evaluation of Scrum and selected scaled agile methodologies, i.e. DAD, SAFe and LeSS is presented.

Figure 3 shows optimal criteria values within the Product and People group for Scrum and agile scaled methodologies DAD, LeSS and SAFe. All methodologies can be used for mission critical projects, SAFe and LeSS even cross-border, and for medium and long-term projects. As all these methodologies are agile, they are suitable for projects with high extent of requirements changes, however do not support reusable components development. Both Scrum and scaling frameworks can be used even for large solutions. All these methodologies require an experienced project manager, high qualified and motivated team members and high user involvement (part of the team or on daily basis). A significant difference between Scrum and scaling frameworks is evident in the Team size and Distribution criteria values. SAFe and LeSS have the highest value as they are applicable for large teams (more than 100 people) distributed in various countries.

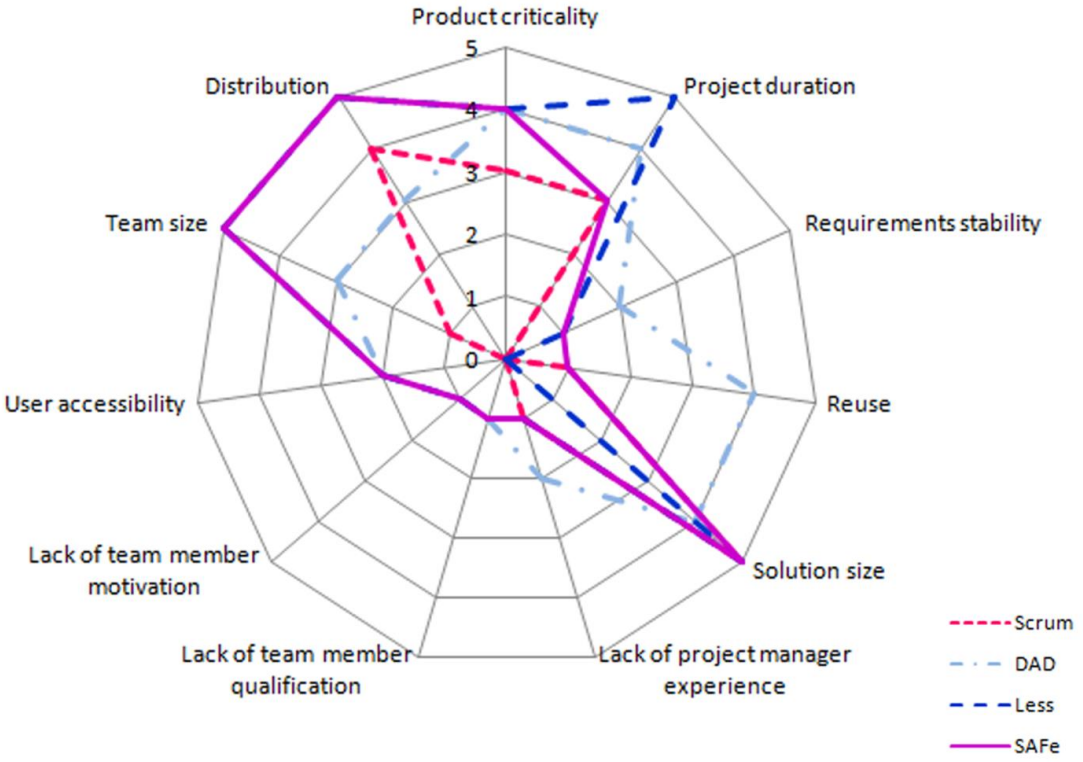

Figure 3 Comparing Values of Product and People Group Criteria of Scrum and Scaled Agile Methodologies

The METES system defines criteria in the Process, Product and People group in the way that the size of the graph area corresponds to the so called methodology weight. Methodology weight was defined by Cockburn (1999) as the product of the methodology's size and density where the methodology size expresses the number of control elements in the methodology and the methodology density the amount of precision and the tightness of tolerance in the methodology (Cockburn, 1999). Based on this definition, heavyweight and lightweight methodologies are distinguished. As depicted in Figures 3 and 4, the Scrum methodology is evidently much more lightweight in comparison with scaled agile methodologies such as DAD, SAFe or LeSS.

Figure 4 captures the comparison of the criteria values for the above mentioned methodologies within the Process group. The Scope criterion is assessed based on mapping the processes to the Process Reference Model (PRM) defined in the ISO/IEC 12207 international standard. Scrum covers only small part of the PRM ISO/IEC 12207 processes, i.e. project management processes (Scope=2). The same value applies to LeSS whereas DAD has a value of 3, because it covers software implementation processes, technical processes, project management processes and also organizational processes. The highest value of the Scope criterion applies to SAFe (Scope=4), which covers even more organizational processes. The Life cycle model criterion is assessed at the level 5 for all methodologies as they use the iterative model with short iterations. The highest value (5) of the Role criterion is set for LeSS and SAFe frameworks that have defined project management and software engineering roles as well as roles at the organisational level. As all methodologies are agile methodologies, they do not define processes, only practices. Therefore, the Process description particularity criterion is assessed at the level 0, respectively 1 for SAFe as SAFe does define a role

responsibility for activities. The same logic applies to the Documentation criterion. It has value 1 corresponding to a simple, agile documentation with only SAFe having value 2 as it requires organizational documents such as the Vision, Project roadmap, etc. The Metrics and Quality management are assessed higher for LeSS and SAFe than for DAD and Scrum having the lowest score.

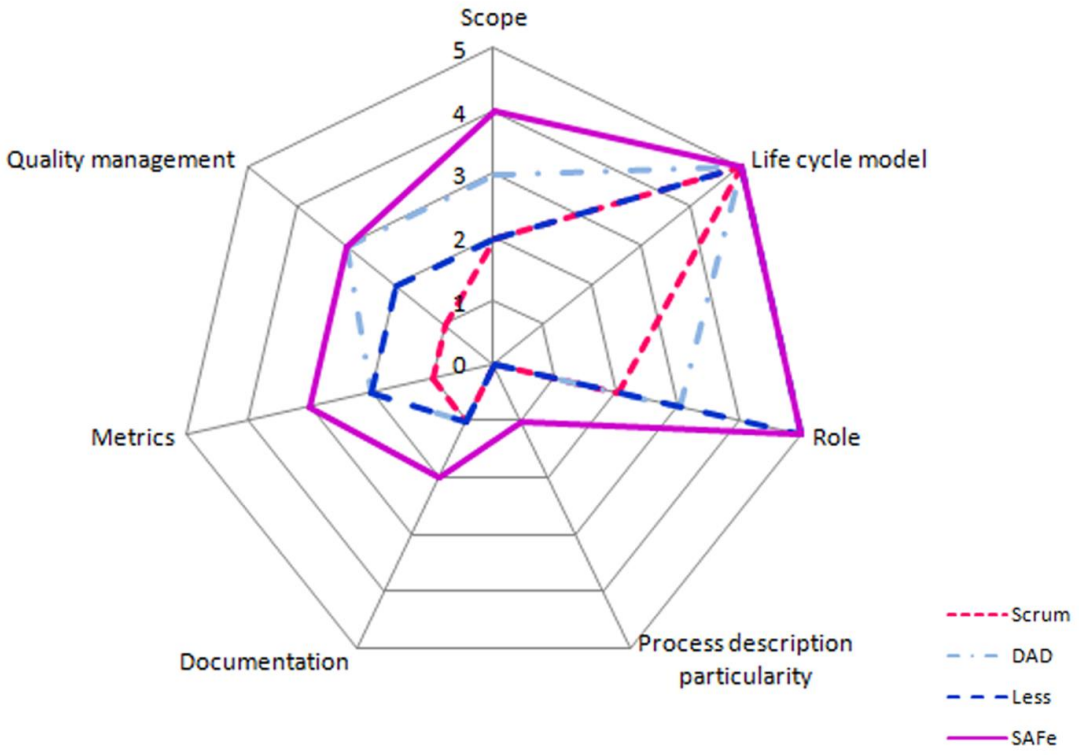

Figure 4 Comparing Values of Process Group Criteria of Scrum and Scaled Agile Methodologies

Figure 5 presents the evaluation of the criteria within the Support group. These criteria are important when particular methodology is to be selected as they help with the understanding of the integrity of methodology resources, the form in which the methodology is available, if it is available in a localized version, how well it is supported etc. Moreover, these criteria evaluate the availability of people trained in working with the methodology as they assess methodology courses taught at universities and availability of training and certification. This criteria group is unique in a comparison to other systems for methodology evaluation and selection. The Integrity of resources criterion has the highest value for DAD as it is described in one source, i.e. book (Ambler & Lines, 2011), and other sources are available. SAFe has the second highest value (value 3) as SAFe framework is described in detail in one basic source at the official SAFe website (SAFe, 2017). Other methodologies have various resources being not that clear. Although Scrum also has official websites, Scrum.org and Scrumguides.org, their information granularity is coarse-grained and other sources are needed (Integrity of resources = 2). The highest value of the Availability criterion applies to Scrum (value 4) as it has official open source licence, while the lowest value to DAD. The SW tools support criterion evaluates how administration of the methodology is supported by SW tools. Regarding Scrum, there is a methodology library for Eclipse Process Framework Composer which enables methodology publication and customization. The same value (4) is set for SAFe having commercial software

CodeBeamer available. Being a positive feature, all methodologies enable customisation even during the project. Methodology implementation support is definitely outstanding for Scrum which is widespread. From Figure 5 it is apparent that Scrum has a substantial advantage in this sense as it is available in Czech, taught at universities and locally trained and certified. Agile scaling frameworks are quite new, and thus there is not such a support in the local area and within the tools support. Nevertheless, these frameworks are taught at Czech universities as well, but only in theoretical lectures, and training and certification is also to some extent available in the Czech Republic.

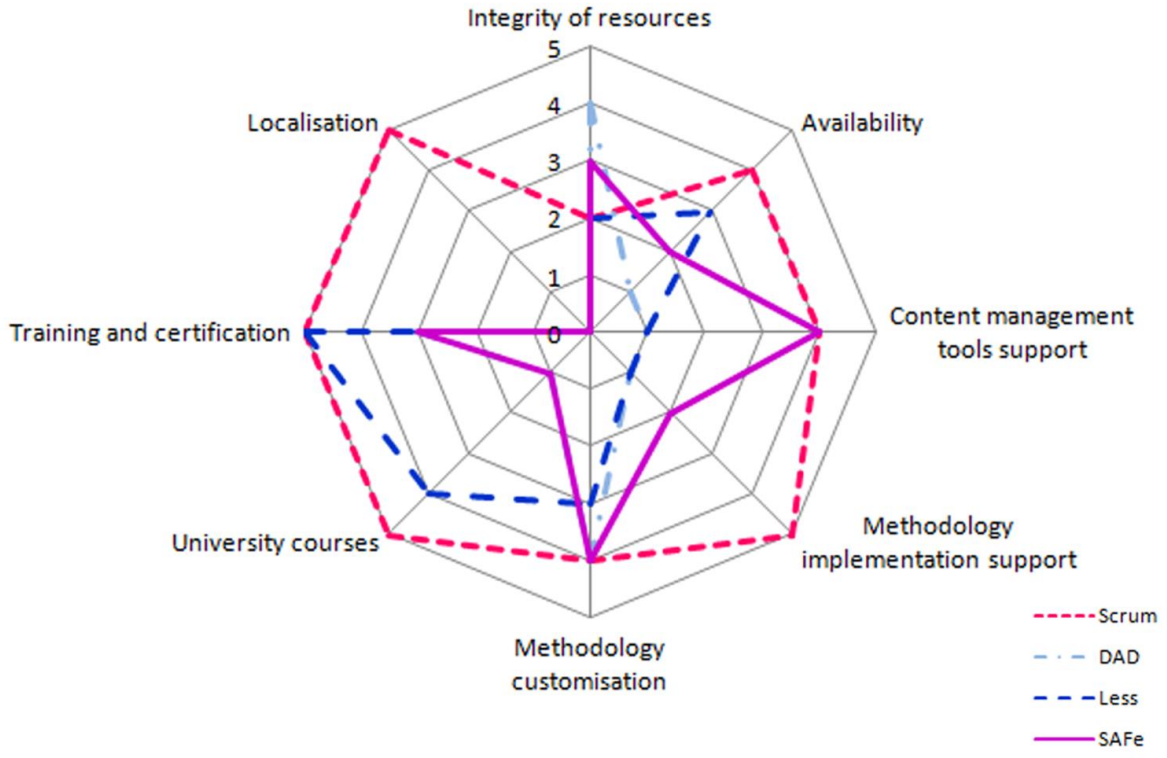

Figure 5 Comparing Values of Support Group Criteria of Scrum and Scaled Agile Methodologies

CONCLUSION

Software systems play a key role in today's society and a successful development of these systems is crucial. To increase the success rate of software development projects, the Software Process Improvement (SPI) initiative has been established. Besides international or professional standards, a lot of ISD methodologies are available. There has lately been a strong tendency towards adoption of agile methodologies even for larger projects and distributed environment. To support using agile in-the-large, several scaled agile methodologies and frameworks have emerged. In this paper, three scaled agile methodologies and frameworks, i.e. Scaled Agile Framework (SAFe), Large-scale Scrum (LeSS) and Discipline Agile Delivery (DAD), were described in more detail. The objective of this research was to examine newly emerged ISD methodologies, especially scaled agile methodologies and frameworks and answer the research question: What are the differences in characteristics and implementation support of various scaled agile methodologies and how these characteristics differ from other ISD methodologies? To fulfil the research objective, the assessment of ISD methodologies was conducted in 2017 using the Methodology Evaluation System METES. The conceptual model of the METES system was presented along with the description of the criteria and procedure for

methodology selection. In a comparison to other systems for methodology evaluation and selection, the METES system enriches the methodology selection process by introducing criteria assessing integrity of resources, availability of the methodology, support of the methodology implementation and customization, occurrence of university courses, training and certification, and localization of the methodology.

Besides tabular representation, the graphical representation was presented which enabled the comparison of several methodologies, i.e. Scrum, DAD, SAFe and LeSS. The reasons behind criteria value setting were stated and differences between individual methodologies were highlighted.

Main contribution of this research is of three kinds. First, newly emerged scaled agile methodologies, i.e. Scaled Agile Framework (SAFe), Large-scale Scrum (LeSS) and Discipline Agile Delivery (DAD) were examined and assessed according to the Methodology Evaluation System METES. Second, based on this assessment characteristics of these frameworks were depicted and visualized supporting a better understanding of these methodologies. Third, scaled agile methodologies were compared to each other and with other ISD methodologies.

Implications of this research mainly support practice and education. As selected ISD methodologies including lately emerged scaled agile methodologies are assessed according to METES, they can be used by companies for the selection of the appropriate methodology.

The METES with its basis of assessed ISD methodologies acts as a conceptual tool for the ISD methodologies understanding and mutual comparison and is used in university courses at the Prague University of Economics. Learning the evaluation criteria helps students in understanding various aspects of methodologies that should be examined. Evaluation of individual methodologies promotes a better understanding and needed comparison of such methodologies.

However, this research does imply limitations. The main limitation lies in the process of assessing ISD methodologies which was performed by students and verified by one expert. Therefore, the results must be regarded as an initial assessment that must be further validated using e.g. focus group or several experts. Furthermore, this research has a limited impact due to a strong regional focus. All criteria within the Support group of the METES are defined from the point of view of the Czech Republic and thus the results of their assessment are applicable in the context of the Czech Republic. It is certainly possible to redefine the criteria and their scale within the Support group to take into consideration other countries.

ACKNOWLEDGEMENT

This research was processed with a contribution of the long term institutional support of research activities by the Faculty of Informatics and Statistics, University of Economics, Prague.

REFERENCES

Ambler, S.W. (2013). IT Project Success Rates Survey Results. Ambysoft [online]. Available at: http://www.ambysoft.com/surveys/success2013.html.

Ambler, S., & Lines, M.. (2011). Disciplined agile delivery: an introduction. IBM Software, Somers, NY .

Amro Mohammad Hani Al-Said Ahmad. (2014). Agile Large-Scale Software Development: Success Factors, Challenges and Solutions. i-Manager's Journal on Software Engineering, vol. 8, no. 3, pp. 1.

Boehm, B., & Turner, R. (2004) Balancing Agility and Discipline: A Guide for the Perplexed, Addison Wesley. ISBN 0-321-18612-5.

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. Softw. IEEE 22 (5), 30–39.

Buchalcevova, A. (2004) Methodology Patterns. Orlando 21.07.2004 – 25.07.2004. In Chu, Hsing-Wei, Aguilar, Jose, Ferrer, Jose, Syan, Yu-Ru, Cheng, Chin-Bin (ed.). CITSA 2004. Orlando : IIIS, 2004, s. 200–203. ISBN 980-6560-19-1.

Buchalcevova, A. (2009) IS Development Methodologies (in Czech) Praha : Oeconomica. 206 p. ISBN 978-80-245-1540-3

Buchalcevova, A. (2011). The Methodology Evaluation System Can Support Software Process Innovation. Information Systems Development. Springer New York. 561-571.

Cockburn, A. (1999). A methodology per project. Humans and Technology Technical Report HaT TR, 4.

Dikert, K., Paasivaara, M., & Lassenius, C. (2016) Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of Systems and Software. Sep 30;119:87-108.

Dingsøyr, T., & Moe, N. (2014). Towards principles of large-scale agile development. In: Dingsøyr, T., Moe, N., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (Eds.), Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation. In: Lecture Notes in Business Information Processing, vol. 199. Springer Interna- tional Publishing, pp. 1–8. doi: 10.1007/978- 3- 319- 14358- 3 _ 1 .

Disciplined Agile 2.X (2017) [online]. Available at: http://www.disciplinedagiledelivery.com/.

Dybå, T., & Dingsøyr, T. (2009). What do we know about agile software development? Softw. IEEE 26 (5), 6–9.

Gregor, S., & Hevner A. (2013) Positioning and Presenting Design Science Research for Maximum Impact MIS Quarterly Vol. 37 No. 2, pp. 337-355

Hecksel, D. L. (2007) Methodology Evaluation and Selection. [online] Sun Software Services, Sun Microsystems. Available at: http://www.davidhecksel.com/projectcontext/whitepaper.html. Accessed: 7/10/2007.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004) Design Science in Information Systems Research, MIS Quarterly, (28: 1).

Johnson, J. (2006) My Life is Failure. The Standish Group International, Inc. ISBN 1-4243-0841-0.

Kataria, N. (2016) Implementing Agile Development at Scale : An Industry Case Study, University of Denver, Ann Arbor, ProQuest Dissertations & Theses A&I; ProQuest Dissertations & Theses Global, https://search-proquest-com.zdroje.vse.cz/docview/1811433824?accountid=17203.

Larman, C., & Vodde B. (2013) Scaling Agile Development. CrossTalk 9: 8-12.

LeSS framework (2017) [online]. Available at: https://less.works/less/framework/index.html.

Pearson Education (2016) Product creation framework. https://neo.pearson.com/groups/product-creation-framework. Accessed:05/10/2016.

Peffers, K., Tuunanen, T., Rothenberger, M.A., & Chatterjee, S. (2007) A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems Volume 24 Issue 3, pp. 45-78.

SAFe (2017) [online]. Available at: http://scaledagileframework.com.

Scaled Agile. (2016) SAFe 4.0 Introduction: Overview of the Scaled Agile Framework for Lean Software and Systems Engineering.White Paper.

Taylor, P. S . (2006) Applying an Agility/Discipline Assesment for a Small Software Organisation, Product Focussed Software Process Improvement, 7th International Conf. PROFES 2006, LNCS 4034, Springer, 2006 pp. 290-304.

The Standish Group. (2015) .CHAOS Report 2015 [online]. Available at: http://www.infoq.com/articles/standish-chaos-2015

VersionOne, Inc, (2016) 10th annual State of Agile Development survey.