



AN ATTEMPT TO IMPLEMENT AGILE MANAGEMENT METHODS IN A SELECTED DEPARTMENT OF AN ENTERPRISE

Grzegorz Ćwikła¹, Krzysztof Bańczyk²

¹Silesian University of Technology, Faculty of Mechanical Engineering, Konarskiego 18A, 44-100 Gliwice, Poland

²Research and Development Centre for Mechanical Appliances OBRUM Ltd, Toszecka 102, 44-117 Gliwice, Poland

Corresponding author: Grzegorz Ćwikła, grzegorz.cwikla@polsl.pl

Abstract: In the first part, the article presents Agile methodologies on the example of the Scrum method, including the scheme and basics of using this method, as well as examples of Agile implementation outside the IT. The second part presents an attempt to implement Scrum in the process engineering (PE) department of chosen company. The scope of tasks of the PE department and methods of operation before implementation of Agile methods was described and analysed, including problems that led to Agile implementation. The assumptions, applied methods and stages of implementation of the Scrum were presented. The first attempts to run process engineering department in accordance with the Scrum methodology were described, which resulted in positive results, but also problems that were identified, what called for adjustments in the way the department works using Scrum. The article ends with an attempt to assess effects of implementing described solutions.

Key words: Agile project management, Scrum, process engineering, implementation.

1. INTRODUCTION

High competition resulting, inter alia, from globalization, which is a feature of the modern market, results in enterprises seeking methods of increasing efficiency in various areas of its operation. One of these areas is managing and organizing the work of teams responsible for various aspects of the company's work. In the field of industrial production, the concept of Industry 4.0 has recently been promoted, covering issues related to production methods suitable for the modern world (Paprocka et al., 2018). Even before formulating the assumptions of Industry 4.0 (2012-2013), in 2010 there were publications classifying different generations of management systems (Appelo, 2010) and introducing the concept of Management 3.0, based on agile methodologies, which in turn were formulated for the needs of software development at Manifesto for Agile Software Development (Beck et al., 2001)

The concepts proposed in Manifesto, organising and supporting the methodology of software development have been later, after success in IT, treated as the basis for developing a whole range of their application in

branches of business and industry other than software (e.g. Agile: Project Management, Product Development, Manufacturing, etc.). Agile methods have a real impact on improving the work of project teams, in the report (VersionOne, 2018) created on the basis of surveys, the following positive effects were indicated: improved ability to manage changing priorities, better match of IT and business aspects, shorter time to market, increased productivity and improvement of team morale, better product quality and management of a dispersed team, lower project risk and reduction of project costs.

This article presents the basics of agile methodologies and preliminary considerations and attempts to apply agile methods to support project development management and process engineering, resulting from the implementation of these methods in an enterprise involved in defense-related projects, taking into account the specifics of this industry. Ultimately, it is planned to fully introduce these methods into the daily practice of the company, what is associated with the implementation of a practical PhD thesis program.

2. BASICS OF AGILE CONCEPT

Manifesto (Beck et al., 2010) was not the first attempt to introduce a modern approach to creating large software projects, it was a synthesis of previously proposed and practically used methods such as: Rapid Application Development (RAD), the Unified Process (UP) and Dynamic Systems Development Method (DSDM), Scrum, Crystal Clear and Extreme Programming (XP) and Feature-Driven Development. Agile methods have been contrasted with the classic project management methodology, known as waterfall. The waterfall method can be depicted as successive operations that cannot be performed in a different order, each must be fully completed and forwarded to the next workstation, department, employee or team of specialists. The software developers stated that testing the application only after its completion exposes the project to the large costs associated with removing errors in the case that the written code contains a lot of errors. In accordance

with Agile principles (or values), software developers should prefer (1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change over following a plan. Agile teams should treat the values as a list of things to do and recognize that things listed after the word "over" are less important, which doesn't mean they won't be considered at all. Using Agile methods, priorities can be shifted from iteration to iteration and new features can be added into the next iteration, considering that changes always improve a project and provide additional value (Von Rosing, 2015).

The first postulate means that in practice, when running projects, people should not be reduced to the role of resources that can be used (as in classic management methods). Team members know best how to perform specific tasks, do not adapt work to tools, only tools for projects. According to the second point, the most important thing is to provide the customer with a product that works in accordance with his requirements, and the task of creating documentation falls to the background. The third point emphasizes the role of interaction with the future user of the project during its creation, allowing for ongoing changes resulting from the observation and analysis of the operation of the product's working versions. The last point suggests that the blind implementation of the plan created before the start of the project will probably not allow you to create a satisfactory product, because in a changing environment plans should not be a constant factor - it should be possible to adapt them. Two more rules are also often mentioned: the whole process should remain agile and control over process costs should be maintained.

The Manifesto also formulated 12 Agile principles that suggested the desire for the fullest customer satisfaction. The Twelve Principles are the guiding principles, they describe an environment in which change is welcome, and the customer is the focus of the work.

The twelve principles of agile development include:

1. Customer satisfaction through early and continuous software delivery – customers should receive working software samples at regular intervals, rather than waiting long periods of time between releases.
2. Accommodate changing requirements throughout the development process – it is the ability to avoid delays when a requirement or feature is subjected to changes.
3. Frequent delivery of working software –the team operates in iterations, ensuring regular delivery of working pieces of software.
4. Collaboration between the business stakeholders and developers throughout the project – when the business and technical team are closely collaborating, better decisions can be made.
5. Support, trust, and motivate the people involved –

motivated employees are more likely to deliver the most satisfactory results.

6. Enable face-to-face interactions – improves internal team, as well as client collaboration.
7. Working software is the primary measure of progress – customer should be able to see how project is progressing.
8. Agile processes to support a consistent development pace – teams should establish a repeatable and maintainable speed at which they can deliver working software modules.
9. Attention to technical detail and design enhances agility – the team can maintain the pace, constantly improve the product, and sustain change.
10. Simplicity – focus on current steps to finish it fully.
11. Self-organizing teams encourage great architectures, requirements, and designs – allow team members to chose tasks, taking into account their skills, experience and preferences.
12. Regular reflections on how to become more effective.

Most of these rules can also be applied in areas other than software development, there may of course also be further rule suggestions resulting from the specificity of the tasks to which the Agile rules will be applied. In addition to Agile Software Development, there are also Agile Project Management, Agile Process Engineering, Agile Supply Chain, Agile Manufacturing, Agile Service Management, Agile Enterprise, etc. Agile approach can also be used in Business Process Management, including planning, analysis, design, implementation, ongoing management, monitoring and improvement (Von Rosing, 2015).

Summing up this part of the considerations, it can be stated that the core of Agile methodologies is the organization's ability to introduce rapid changes and adapt structures and activities to the constantly changing needs of clients.

2.1 Agile on the example of the Scrum framework

Scrum (originally this word means a type of rugby formation, consisting of teams holding hands in front of each other and jointly pushing members of the opposing team) is a framework that is a set of methods and tools used to create computer software with agile methods, proposed by J. Sutherland (MacKay, 2018). In accordance with Scrum assumptions, the first stage involves planning the system (*Planning & System Architecture*), then iterations (*sprints*) not longer than 30 days are performed, carried out until project completion (Figure 1). The project begins with collecting a list of user requirements, usually saved in the form of the so-called *user stories*, each of which describes one feature of the system. The *Product Owner*, who is a member of the (usually interdisciplinary) team (3-9 people) responsible for

contacting the *client* and presenting his expectations, is required to prioritize the requirements and the main goal of the first run. The *Scrum Master* is responsible for the advance of the process in accordance with Scrum guidelines. As a result of the analysis, a product / requirement list (*Product Backlog*) is formulated. The project goal is explicitly recorded in the team members' room.

Next, there is *Sprint Planning*, during which tasks with the highest priority are selected. The time of

completion, labor intensity, complexity and risk of each task are estimated. The result is a task list with estimated time consumption, referred to as a *Sprint Backlog*. After the planning stage, the team proceeds with the implementation of the sprint run, during this time *Product Owner* should work with the team to make them understand requirements the well as possible, but without interfering with the implementation. The scope of the *sprint* should not be changed.

Scrum Process

Enter your subhead line here

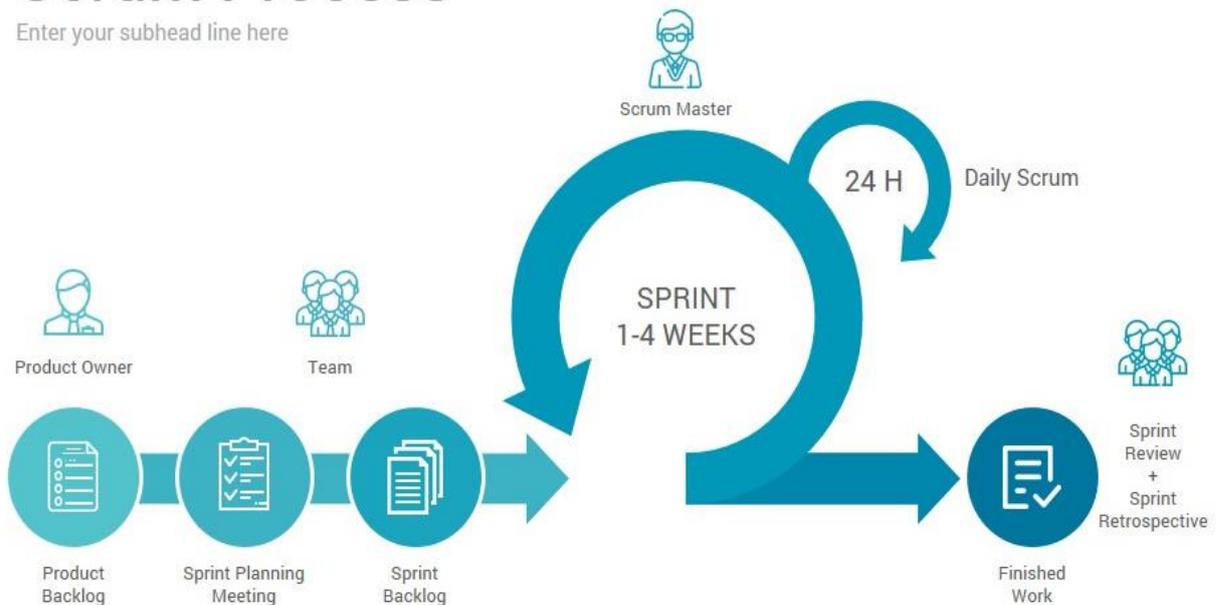


Fig. 1. Scrum methodology diagram [8]

The Scrum team is by definition a self-organizing body, there is no top-down assignment of tasks to individual team members, who themselves choose tasks carried out in accordance with common arrangements, skills or preferences. One of the basics of the methodology are daily short (max. 15 minutes) meetings (*Daily Scrum*), during which tasks completed the previous day, problems occurring during their implementation and tasks planned for a given day are discussed (Warcholiński, 2017).

Each sprint ends with a meeting (*Sprint Review*) of project participants, where a product made in its result is presented, team members can express an opinion about it, and an initial meeting date for the next sprint is set. *Sprint Review* is usually combined with *Sprint Retrospective* - at the end of the Sprint, the team stops for a moment, looks at the process, tools, interactions to decide how to improve the process.

3. OPERATION AND RESPONSIBILITIES OF THE PROCESS ENGINEERING DEPARTMENT

An attempt to implement agile methods was carried out in the Process Engineering Department (PE, PED) of an example company, showing high variability in production processes, which is an obstacle for process engineers. The range of tasks of the department includes 3 main groups (Figure 2).

Most of the tasks carried out by the process engineering department, in addition to those related to daily production support, are extensive and before the attempt to implement Agile methods were managed as a project, which in practice meant their division into smaller tasks and setting implementation dates according to the waterfall method. This gives us the chance to try to introduce agile methods for long-term tasks implementation. Subsequently, an attempt to use Agile for the short-term tasks of the process engineering department is also planned.

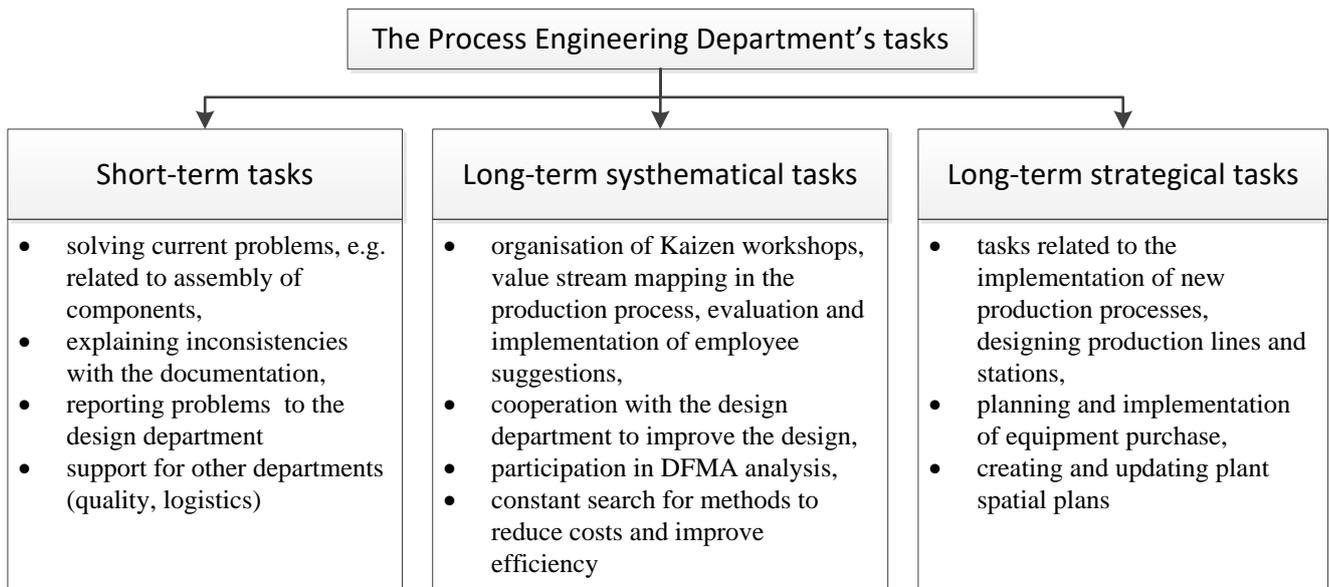


Fig. 2. Tasks of the Process Engineering Department in the example company

3.1 Organization of work before implementing Agile methods

The process engineering department in the described company consists of a manager, process engineers and process engineer assistants. Before attempting to introduce agile methods, tasks were dispatched during weekly meetings (about 2 hours) at which the scope of activities of individual team members was determined. Tasks assigned to the PED were entered into the database through an e-mail, database client or internet application platform (Microsoft Sharepoint). The database allowed to record data such as: name, date of completion and start, task priority, attachments related to implementation, description of task's progress, task status and completion percentage.

This database was updated by employee's continuously during the week and reviewed at weekly meetings, which in practice boiled down to reviewing mainly the highest priority tasks. Employees reported the progress of work and together with the manager determined whether the task was completed or additional action should be taken. Then the manager assigned subsequent tasks that should be undertaken by a given team, divided into specific people. Tasks related to daily activities were chosen by the employees themselves, based on information provided orally or via email by employees of other departments.

Thanks to the analysis of the department's operation, the following problems were identified:

- Reporting of work progress was not frequent enough, leading to a lack of current information on the progress of individual tasks. Employees asked by someone from another department about the status of a given case were unable to answer because they did not have information whether someone was already dealing with it, at what stage of completion it is now and when it can be expected to be completed.

- The existence of many channels for the feeding of tasks and information to process engineers (database, e-mail or verbal communication) and the lack of defined procedures, which meant that in the sheer volume of cases, tasks were often not entered into the database. This caused tensions and escalation of trivial matters into serious problems.

- Tasks entered into the database often disappeared in a large number of other priority tasks, which resulted from poor communication with other departments - it was not described why a specific task is important and should be completed within a specified period of time.

- Great freedom in the allocation of people to deal with specific matters, insufficient supervision over execution.

- Despite the priority specified in the database, the task was often not taken into account. This resulted in many tasks being dealt with at the same time, which resulted in delays in completing important tasks.

- The lack of focus on selected priority tasks that employees should try to complete within one or two weeks heightened the overall impression of the lack of progress.

- The system in which task completion was reported only once a week may be the reason for the subconscious lack of employees' involvement in task completion ("there is still a lot of time to meeting").

3.2 Attempt to implement the Scrum methodology

In accordance with the implementation practices described in the literature [MacKay, 2018], an attempt was initiated to implement the Scrum method in the PED in the following phases:

1. *Create a sense of urgent need* - a training was conducted by an employee of the company's software development department (who has experience with

Agile methods). The basic assumptions of this methodology, methods of task implementation, advantages and potential threats during implementation were presented.

2. *Gather a lead team* - due to the small size of the process engineering department team (10 people), this whole department has become a lead team.

3. *Build a vision and strategy of change* - it was specified that the department should work in a way that uses agile methods similar to Scrum. During the brainstorming, current problems with the department's operation were identified and the concept of the department's functioning using agile methods was proposed. A skeleton of a new database was created to contain tasks performed and intended for execution by PED employees. The fields that such a database should contain and the fields that should be filled in by the applicant (also a person outside the PED) have been specified.

4. *Communicate vision and strategy* - the meeting presented the concept of action and strategies (including improving team work thanks to the Scrum method, better task management and more efficient use of time for meetings).

5. *Grant the power to implement changes* - PED team members have been authorized to act as part of team implementing agile methods by introducing appropriate procedures, creating a task management form, preparing a table for tasks tracking, and creating an agenda for daily and weekly meetings.

A much more extensive task entry form (Figure 3) has been developed, covering, in addition to the fields

used so far (task name, name of the project it includes, subject of the task, expected result, client), a new set of data allowing to describe and classify the problem in more detail. The new form allows PED employees to define, among others:

- Assigned to - a member of the team who deals with the task and the assignment period;
- Start and Due dates, what will happen if the deadline is exceeded;
- Description - a list of detailed actions necessary to complete the task;
- Internal priority - 5 groups defined, from *Emergency* to *Nice to have*;
- Project category - what type of task, several categories have been defined, e.g. *Downtime / Ongoing production, Suggestion system, Trainings, Kaizen events, Warranty, Relayout*, etc.;
- Task status - specifies at what stage the task is being completed (*New item, Approved, In progress, Completed, Cancelled, Waiting on someone else*);
- % complete - a numerical, arbitrary value given by the person carrying out the task informing what (estimated) part of the task has been completed.

A client form (Figure 4) has also been defined and made available to PED clients (internal and external), allowing to specify the *Task name, User story, Expected result, expectations, Deadline*, information what will happen if the deadline is exceeded, and a proposed assignment of the executing person.

The form consists of the following fields and controls:

- Task Name ***: Text input field.
- Project name**: Text input field.
- User story ***: Large text input area.
- Expected result, expectations ***: Large text input area.
- What If Delay ***: Large text input area.
- Deadline**: Text input field with a calendar icon.
- Assigned To**: Text input field with placeholder text "Enter names or email addresses...".
- Sprint planning**: Text input field.
- Start Date**: Text input field with a calendar icon.
- Due Date**: Text input field with a calendar icon.
- Description**: Large text input area.
- Line**: Dropdown menu with "MVD" selected.
- Internal priority**: Dropdown menu with "(3) Important and Not Urgent" selected.
- Project category**: Dropdown menu with "Downtime / Ongoing production" selected.
- Task Status**: Dropdown menu with "New Item" selected.
- % Complete**: Text input field with "0" and a percentage sign.
- Customer**: Dropdown menu with "Production" selected.

Fig. 3. The new database form for PED employees

Task Name:

User story: ★

Proszę podać pełną informację najlepiej w postaci:

- 1) Ja jako..... (np. planista),
- 2) Chciałbym aby
- 3) Potrzebne jest mi to do

Expected result, expectations: ★

Deadline: 📅

What If Delay: ★

Assign to: 👤 📄

Customer: Line: ★ Attachment:

Fig. 4. The new database form for PED clients

6. *Generate short-term winnings* - tests of new procedures, created forms and cooperation with employees of other departments cooperating with process engineers proved to be a success, which showed the need to take the next implementation steps. These effects were identified on the basis of surveys conducted among employees of the Process Engineering Department and cooperating departments, because there was no other way to conduct systematic studies of the effects of pilot implementation.

7. *Don't let go* - Not all actions were successful right away - initially, attempts were made to use the functions of MS Planner to facilitate visual task management by placing them in groups corresponding to the status of the task. It was also possible to obtain statistics on specific sprints or all tasks. However, this program was not integrated with the database and therefore required additional workload, which resulted in the resignation from its use. It was decided to introduce further modifications and improvements. New rules for the operation of the PED were determined, including:

- Sprint time - one week (when the maximum time of 1 month is assumed in the classic Scrum);
- Sprint planning meetings are to last a maximum of 3 hours and also include Sprint Retrospectives;
- Retrospective meetings regarding the operation of the Scrum method in the department are to take place once a month;
- The Daily Scrum will last up to 1 hour and employee will have to answer three questions on it:

What have I done since the last meeting? What am I going to do until the next meeting? What prevents me from completing tasks in this sprint?

It was observed that the Daily Scrum meetings were initially held in such a way that each member came to the conference room with his own computer and reviewed the things he dealt with. This caused a lack of commitment to participate in the meeting, each team member was busy with their own affairs, and the meeting time was up to 1 hour, which was considered an excessive waste of time. During the meetings, topics not related to tasks were also discussed, attention was distracted to things irrelevant to a given sprint. In accordance with the strategy of introducing changes, the team did not give up and introduced a correction of the Scrum methodology implementation.

8. *Create a new culture supporting new behaviors, replacing old habits* - attempts were made to limit the adverse phenomena described in the previous point, MS Planner was abandoned, in its place a table (whiteboard) inspired by Kanban methods was used, in which, in relevant areas related to the status of the task (demanded, planned, approved, completed, waiting) colored sticky notes (colors indicated priorities) with job descriptions were placed (Figure 5). A chart of task burning in the sprint was also introduced, visualizing the number of tasks completed on individual sprint days in relation to the planned number, which allowed easy visual determination whether all tasks planned for implementation have a chance to be completed by the

end of the sprint. Planning in this system also allows to draw conclusions from previous sprints, e.g. it was possible to determine that about half of the tasks performed by each employee are, in fact, unplanned tasks. This knowledge, together with the estimated

time of completion of planned tasks, allows to determine the real number and type of tasks that can be executed during weekly task period planning, so as to realistically plan all tasks that should be completed in the next period/sprint.



Fig. 5. Scrum in the PED - Kanban table and task burning chart

3.3 Identification of problems during the implementation of Agile methodology

According to [VersionOne, 2018], typical obstacles when implementing Agile methods are: inability to change organizational culture (54%), general resistance to change (46%), lack of adequate support from senior management (42%), lack of experience / skills in the field of Agile methods application (41%), insufficient training (35%), inconsistent processes and practices among teams (34%), lack of availability of the Product Owner (31%), ubiquity of traditional working methods (30%), selectivity in the use of tools (24%) and insufficient cooperation and knowledge sharing (21%).

In the case of the described company, many of these factors could be excluded, because the management and employees of the department had the desire and motivation to change and the culture of the enterprise organization itself supported innovation. Obstacles which were noticed during the implementation of agile methods in PED were mainly connected with the lack of experience with Agile methods, the unavailability of the Product Owner, the ubiquity of traditional working methods (used by the rest of the company) and selectivity in the use of tools. Efforts were made to address the problems by modifying the department's work procedures and the tools used.

4. ASSESSMENT OF IMPLEMENTATION OF AGILE METHODOLOGIES AND SUMMARY

Due to the nature of the work of the process engineering department, which is very different from software engineering, it is difficult to directly introduce Agile methods (in this case Scrum) fully based on experience and principles from the IT

industry. The main differences result from the greater variety of tasks of the department's employees and the fact that the role of typical external customers ordering software, in this case is mainly played by other departments of the parent enterprise. In the software development industry, the effectiveness of implementing agile methods is assessed by determining the return on investment in the project, better-quality software code (number of errors, time spent on making corrections) or faster completion of working program modules. The use of Agile methods in the engineering department of the sample company is still a positively evaluated, but still an experiment.

The management is considering expanding it to other departments, but the inhibiting factor is the fact that in the case of implementing Agile in the field of process engineering, unlike software, it is difficult to find reliable, objective and measurable indicators that can be used to determine the effectiveness of implementation.

The most obvious approach is to compare corresponding parameters before and after the introduction of Agile, e.g. the average time to close a task since its submission. This allows an indicative determination of whether the change in the philosophy of operation had a measurable impact on the faster completion of tasks, but it should be remembered that each task may be different and their apparent similarities may be illusory. In the described company, the average time of order completion was shortened, which indicated the legitimacy of continuing the adopted strategy.

Another assessment method is to observe the number of unfinished tasks in a sprint, however, especially in the initial period of the system operation, users do not have the sense to effectively estimate the implementation time, which can also be dependent on external factors,

which falsifies the picture of the effects obtained.

Another solution is to survey process engineers and their clients (in this case, they are mainly internal clients, employees of other departments of the company) on various aspects of the department's work on changed principles. As usual in the case of surveys, the answers are subjective and depend on the previous experience of the respondent. This method was used and analysis of the results showed that the use of Scrum is assessed by most employees as reasonable and bringing measurable benefits in the form of streamlining and systematizing their work as well as efficient problem solving.

Like other changes, the introduction of Agile methods is not always welcome and faces resistance. Organizations observe many types of friction that reduce momentum during the implementation of Agile. The difficulty of measuring the effects of implementation also does not make it easier to popularize innovative methods of operation. The introduction of Agile methods in the practice of a Process Engineering Department, which is the subject of an implementation doctorate, should give the opportunity to develop both the methodology of operation in accordance with Agile assumptions, as well as to propose methods for assessing and verifying the results of implementation.

5. REFERENCES

1. Appelo, J., (2010). *Management 3.0: Leading Agile Developers*, Developing Agile Leaders, Pearson Education, Boston.
2. Beck K., Grenning J., Martin, R.C., Beedle, M., Highsmith, J., Mellor, S, van Bennekum, A., Hunt, A., Schwaber, K., Cockburn, A., Jeffries, R., Sutherland, J., Cunningham, W., Kern, J., Thomas, D., Fowler, M., Marick, B., (2001). *Manifesto for Agile Software Development*, *Agile Alliance*. Available from <http://agilemanifesto.org>, Accessed: 1.12.2018.
3. Krenczyk D., Skolud B., Olender, M. (2016): *Semi-automatic simulation model generation of virtual dynamic networks for production flow planning*. IOP Conference Series: Materials Science and Engineering, **145**, 042021, doi:10.1088/1757-899X/145/4/042021
4. MacKay, J., (2018). *The Ultimate Guide to Implementing Agile Project Management (and Scrum)*. Available from <https://plan.io/blog/ultimate-guide-to-implementing-agile-project-management-and-scrum>, Accessed: 1.12.2018.
5. Paprocka, I., Grabowik, C., Kempa, W.M., Krenczyk, D., Kalinowski, K., (2018). *The influence of algorithms for basic-schedule generation on the performance of predictive and reactive schedules*, Conf. Series: Materials Science and Engineering, **400**, pp. 1757-8981, doi:10.1088/1757-899X/400/2/022042.
6. VersionOne, (2018). *VersionOne 12th Annual State of Agile Report*. Available from <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>, Accessed: 1.12.2018.
7. Von Rosing, M., Von Scheel, J., Gill, A.Q., (2015). *Applying agile principles to BPM* [in] *The Complete Business Process Handbook*, Morgan Kaufmann/Elsevier.
8. Warcholinski, M., (2017). *Differences Between Lean, Agile and Scrum*, Available from <https://brainhub.eu/blog/differences-lean-agile-scrum/>. Accessed: 1.12.2018.

Received: April 12, 2019 / Accepted: December 20, 2019 / Paper available online: December 25, 2019 © International Journal of Modern Manufacturing Technologies