

# QUANTITATIVE ASSESSMENT OF THE IT AGILE TRANSFORMATION

Cezary Orłowski, Artur Ziółkowski, Grzegorz Paciorkiewicz

*IBM Centre for Advanced Studies at Campus, Poland*

**Corresponding author:**

*Cezary Orłowski*

*WSB University in Gdańsk*

*Grunwaldzka 238a, 80-266 Gdańsk, Poland*

*phone: (+48) 508 87 67 84*

*e-mail: corlowski@wsb.gda.pl*

Received: 13 October 2016

Accepted: 23 January 2017

**ABSTRACT**

The aim of this paper is to present the quantitative perspective of the agile transformation processes in IT organisations. The phenomenon of agile transformation becomes a complex challenge for an IT organisation since it has not been analysed in detail so far. There is no research on the readiness of IT organisations to realise agile transformation processes. Such processes also prove to have uncontrolled character. Therefore, to minimise the risk of failure referring to the realisation of transformation processes, it is necessary to monitor them. It is also necessary to identify and analyse such processes to ensure their continuous character.

**KEYWORDS**

IT organisation, agile transformation, agile project management, scrum, maturity capsule.

## Introduction

Nowadays, the business environment is dynamic, characterised by high speed, high change, complexity, and uncertainty of factors [10–12]. The world of Information Technology (IT) has flattened. Due to the combination of technology progress and lowering political barriers, it is possible for people and organisations to work with almost anyone, in any place, at any time [7–9].

As an approach, project management - the application of knowledge, skills, tools, and techniques to project activities [1, 13, 14] - must ensure the successfulness of addressing and handling dynamic changes dealt with by a provider. Neither a client nor a business environment is constant, but also a provider's organisation usually changes, adopting for market situations that ensure the ability to remain competitive i.e. for the last twenty years, the organisational structure tends towards the flat hierarchy by removing middle management to shorten communication cycles and reduce costs.

The purpose of this paper is to adapt the estimation techniques in an IT organisation during trans-

formation processes supporting Scrum, which is one of Agile management approaches. The use of the introduced techniques supports the provider organisation (project managers) with data that can help understand and address challenges brought by dynamic changes in the business environment during software project development. The proposed techniques are not only focused on project evaluation but also project surroundings, including relations with customer and process appearing in the provider organisation. Such a complete approach is a result of the implementation of the C-S-P (Client, Supplier (provider), Project) Maturity Capsule model for a project realised in Scrum [3, 5, 6].

This paper has four parts. The first part is the introduction to the cross-section on project management practices describing why and how project management changed over last forty years towards Agile approaches (Fig. 1). The Scrum framework as a popular representative of Agile methodology is elaborated in this part.

The second part describes the mapping of the C-S-P Maturity Capsule to the Scrum framework as a method of negentropy evaluation of this process.

Negentropy is understood as a measurement of the degree of an organised process. The concept of the C-S-P Maturity Capsule is mapped on three equally important elements represented by Scrum roles, which are Development Team, Product Owner, and Scrum Master. To allow the negentropy measurement, the techniques of the Investigation Questionnaire and Normalised Velocity are proposed and elaborated.

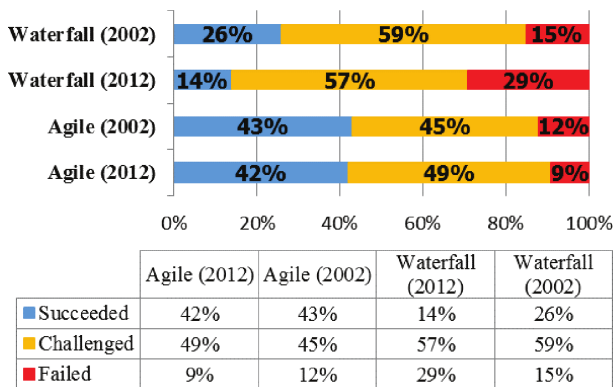


Fig. 1. Project results by style from CHAOS research in years 2002 and 2012 [2].

The third part presents verifications of the proposed techniques based on data acquired from the analysed Scrum Team. Conclusion and observation are divided, based on the point of view represented by interests of a recipient. The results discussed in the first part are given from the point of view of the Scrum Master, and focused on a narrow time horizon. The second part presents the view of the Product Manager fixated on a wider time horizon.

## Scrum Maturity Capsule

In Advanced Modelling of Management Processes in Information Technology, authors are introducing the term the Client – Supplier – Project (C-S-P) Maturity Capsule as a “set of maturity evaluation of the organisation of supplier (provider) and the client, as well as the project (estimated by scalar project negentropy)” [4, 5].

In this approach, a common measure of maturity is applied to all three components responsible for the overall evaluation. The C-S-P Maturity Capsule approach formulates a complete project description, as opposed to contemporary philosophy. That is to say that the maturity evaluation of components is not treated separately but as a whole. This integration expresses the need for the simultaneous study of the maturity of these three entities while evaluat-

ing, which allows for the effective and comprehensive quantification of project management.

The C-S-P Maturity Capsule is used for the verification of the model of information technology management (MITM) designed to address multilayers, usually complex, relations of enterprise architecture. Suited for company structure, it is too exaggerated for the Scrum process analysis. However, the idea of treating a client, supplier and project maturity as one whole, and using common evaluation measurements is the right approach. Adaptations of this idea lead to the definition of the Scrum Maturity Capsule.

This approach is good for the verification of the implementation of the Scrum process. However, it is not sufficient for the verification at the level of project management, whether the decision to pick the Scrum framework was the right call. The true value of the Scrum framework for the product development must be verified against benefits gained for product delivery.

Two recipients of the evaluation mark can be determined:

- The Scrum Master analysing the evaluation data in short term. The focus should be on feedback of the Scrum framework implementation and identification of impediments that must be removed. The area of operations is concentrated on Scrum Team processes.
- The project manager focusing on a long-term analysis and operating on Scrum of the Scrum area. Based on measurement data awareness of the process implementation and the impact on product delivery can determine what helps in taking proper actions.

In smaller projects, the Scrum Master and project manager might be represented by the same person, but this is rather impossible in a larger project. Usually, in that situation, Scrum Masters from all teams meet on Scrum of a Scrum meeting with their project manager to address Scrum Team needs i.e. blocking issues.

Regardless of how many Scrum Teams work on a project, each Scrum Team consists of the Product Owner, Development Team and Scrum Master [8]. The evaluation of Scrum roles that represent processes, for which they are responsible in Scrum, should provide proper verification of the Scrum framework implementation.

Mapping Scrum roles to the three-dimensional description of the C-S-P Maturity Capsule leads to the definition of the Scrum Maturity Capsule presented in Fig. 2.



Fig. 2. Scrum Maturity Capsule.

The Scrum Maturity Capsule adopted here has five-level scale values. It is recognised that many ratings of the Scrum Maturity Capsule implementation might fall into broad ranges below “Very Small” and above “Very High”. One of the implications of that fact will treat boundary value implementations in the same way regardless of how far those are situated. However, it is assumed that most organisations would fall into the proposed range of values, and the proposed scope of the evaluation is the most efficient for these.

The Scrum Maturity Capsule can be used by most Scrum Teams. It addresses needs for the evaluation of the Scrum framework for an individual Team working on a software project development, as well as for many Teams involved in this process. The project manager as well as the Scrum Master are beneficiaries of received results.

To support negentropy, the measurement techniques of the Investigation Questionnaire and Normalised Velocity are proposed and elaborated.

## Verification of the Scrum Maturity Capsule

The focus of this paper is on analyses of single Scrum Teams. The proposed evaluation techniques allow for the negentropy measurement that helps understand and identify the Scrum Team situation in terms of the process of product delivery. To fully present this situation, the focus was not only on the measurement of the Development Team Velocity but also on the relation with the customer and the Scrum framework implementation as an adaptation of complete approach introduced by the C-S-P Maturity Capsule.

The environmental pressure to deliver a product at a faster rate forces the team to abandon the commonly-used Waterfall approach and replace it with the Agile methodology. Scrum was introduced at the beginning of 2010 to analyse teams at the end of a realised project. Three months later, a new project was developed using the Agile approach. The decision to introduce the Scrum framework for software development was a directive from above, as a part of the whole organisational change.

Scrum as the Agile practice was chosen based on few factors, one of them being the fact that Scrum is not too light and not too heavy from among the available light approaches. It can be used by mature teams in terms of “thinking” as well as by the teams that have just introduced an agile practice.

The analysed Scrum Team is placed in an organisation with high dynamic changes related to people-to-project allocation. Although the size of the Development Team was constant during single Sprints, it was changing during the project life. For the analysed period of time, the Team size changed from five to eleven developers. Additionally, allocation of individual team members constantly varied in the case project. Even if the size of a team was identical from Sprint to Sprint, it could not be interpreted as exactly the same as the Development Team because the personnel rotation also occurred or different allocation was available at that time. The Scrum Master’s attempts to remove this inconvenient situation failed, as the changeability of the size of the Development Team was caused by the dynamic structure of the organisation, which was beyond the reach of the Scrum Team.

The awareness of the team of agile process development increased since 2010, and at the beginning of the fourth quarter of 2012, the need for the evaluation of the process occurred. At that time, it was the 15th Sprint of the project, but for the sake of this paper, it is represented as Sprint 1, as it was the first time when the measurement data were gathered. The presented data are from more than one year time period, starting in October 2011 (Sprint 1) and ending with the year 2012 (Sprint 27).

The Normalised Velocity chart provided great insights for the Scrum Team as addressed, opposite to the Typical Velocity measurement, team members’ fluctuation from Sprint to Sprint, and changing the allocation of the dedication to the project.

During retrospective meetings, the Scrum Master updated Normalised Velocity charts to visualise how well the Team did in comparison to previous Sprints. The focus was never on the calculated value of the

Normalised Velocity, only on the change in the graph trend and the observed tendency.

In Fig. 3, seven interesting observations in the Normalised Velocity are marked by the blue rectangle and will be discussed in comparison to the Typical Velocity. As a general observation, the trend provided by the Typical Velocity in some cases is much different in amplitude and direction changes from the trend observed based on the Normalised Velocity.

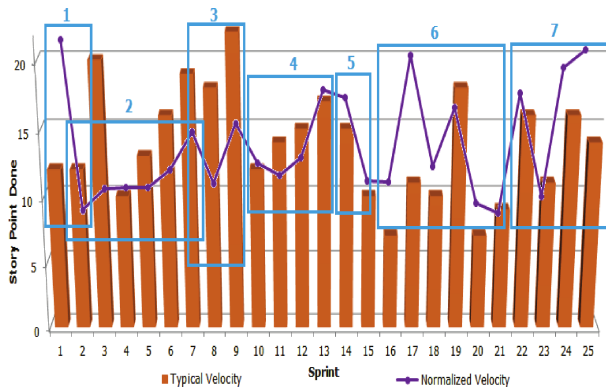


Fig. 3. Conclusion from the comparison of Typical and Normalised Velocity charts.

The first observation is between the first and second Sprints. In both Sprints, it can be proved (during the Typical Velocity stabilisation) that twelve Story Points were delivered and by the visible tremendous drop in the pace of the Normalised Velocity. The Normalised Velocity is a much better way to reveal the fact of the change in the size of the Development Team that took place between the Sprints. In the second Sprint, three new team members joined the Development Team. As new members, they were inexperienced in project matters, and the Development Team delivered the same amount of Story Points as in the previous Sprint, observed on the Typical Velocity as a constant pace. However, three new team members determined the growth of 60% compared to the original size of five team members. Therefore, such a strong change in the pace can be observed on the Normalised Velocity chart. To achieve the stability with the new size of the Development Team, at least twenty-six Story Points should be delivered in the Sprint 2.

The second observation can be described as a learning curve of new team members. On both the Typical and Normalised Velocity, the rising trend of the pace can be observed. However, a slight difference between charts is noticeable in the third Sprint. The amount of story points visualised on the chart of the Typical Velocity in that Sprint indicates a huge boost in the work pace compared to the previous

(second) Sprint. On the chart of the Normalised Velocity, the increase is also noticeable but is not as strong. Based on the chart of the Typical Velocity, the pace of delivery in the second Sprint is higher than in the seventh Sprint, but the Normalised Velocity provides the opposite conclusion. The difference in observations is caused by non-inclusion of changes in the Sprint length and Team size (the available time dedicated to projects) on the chart of the Typical Velocity. In the third Sprint, the Development Team, based on the Net Capacity calculation (Table 76. Project Sprint measured data), could dedicate 375 hours, which is 70% more time than that available in the previous Sprint (260 hours). Two factors are responsible for the change in the available hours between Sprints, namely, the third Sprint was longer by two working days, and a new member joined the team. Both factors were considered in the chart of the Normalised Velocity by calculating the Story Point in correspondence to available hours, which was not done for the Typical Velocity chart.

Delivering twenty Story Points in the third Sprint (compared to twelve Story Point in the second Sprint) is a noticeable increase, but it also must be interpreted in the context of new team members who joined in the second Sprint. Some of the Stories that started in the second Sprint were finalised in the third Sprint. The Development Team could not keep its pace before the enlargement (the second Sprint), and work was summarised in the third Sprint.

The exponential character of the learning curve observed from the second to the sixth Sprint, reflects the rising experience of new team members and their influence on the pace of delivery of the Development Team. The decision to increase the Team by three new members in the second Sprint (expanding the Team size by 60%), was made to increase the pace of delivery but was, in fact, a disruption. Although the Development Team delivered the increasing number of Story Points in the following Sprint, the pace was not as effective as it was before the enlargement of the team. This conclusion could only be made by analyses of the Normalised Velocity chart as it could not be determined from the Typical Velocity graph.

The third observation is a continuation of the learning curve of the Development Team, as described in the second observation. In Sprints seven, eight and nine, the size of the Development Team amounted to ten members (100% growth compared to the first Sprint), and an increase in the pace was characterised by the linear rather than exponential function. The change in the function characterization might be due to the fact that the Scrum Team exceeded the recommended maximum size of nine

members, and the internal noise became harder to remove. The time box of fifteen minutes for the Daily Scrum barely addressed the need for time, especially when some members were inexperienced in project matters. At this point, splitting the Scrum Team into two might be a wise solution. On both charts, the drop in the delivery pace is observed in the eighth Sprint. However, the stronger impact is visualised by the Normalised Velocity rather than by the Typical Velocity.

## Conclusions

In this paper, the implementation of the C-S-P Maturity Capsule for the project realised in the Scrum framework was elaborated and successfully verified (as an example). The adaptation of a wide approach to the project management analysis represented by the C-S-P Maturity Capsule concept led to the definition of the Scrum Maturity Capsule, where equally important components were represented by Scrum roles, which are the Development Team, Product Owner, and Scrum Master.

The maturity of capsule elements is expressed with the negentropy phenomena, which is a measurement for the organisation of a process. That approach allows for the interpretation based on the change in time as it is expected that the observed trend will be characterised by a rather low value at the initial stage of the project, and will increase to the point of saturation with the project progressing towards the finalisation. Additionally, the application of five-level linguistic scale from “Very Small” to “Medium” and “Very High” provides an intuitive interpretation of received results.

To support negentropy, the measurement techniques of the Investigation Questionnaire and Normalised Velocity were introduced. It was proven that both methods were powerful tools useful for a Scrum Team, as they provide tremendous insight into the Scrum process evaluation. The Scrum Team adopted the use of the proposed tools as a part of the Scrum framework, which allowed them to address and understand areas that were hard to determine, yet the only improvement was only possible following the identification of a problem.

Additionally, during this practice, the increase in the motivation of the Development Team was observed. The team strongly believed in data provided by the introduced techniques and did not want to commit to a smaller effort that had been observed in the previous iteration. It was an unexpected but positive observation proving that the visualization received from the proposed tools was trustworthy.

Once negentropy is calculated and expressed in the scalar value of the Scrum Maturity Capsule, it can be analysed. The recipient of the result might be the Scrum Master, who is especially interested in the evaluation data for short time intervals. The focus is on the Scrum Team and the implementation of the Scrum framework. Based on the data interpretation from the identification of impediments particular to the Scrum Maturity Capsule, is possible, and that allows applying an appropriate remedy.

The second recipient would be a project manager, focused on a long-term analysis and operating on the Scrum of the Scrum area. The measurement data can be used to determine the awareness of process implementation and the impact on the product delivery, which helps in taking proper strategic decisions to minimise the risk of project failure.

A high ratio of project failures (less than 40% are successful) presented and elaborated in this paper, based on the CHOAS report, proves that it is the main problem of IT providers. It is caused by the fact that software projects are characterised by a high level of complexity and must be effectively managed. There are no silver bullets, but as the Standish Group reports show, Agile projects are three times more successful than non-agile (waterfall) projects.

Currently, Agile practices are best-known approaches for complicated and complex projects, and the vast majority of today’s IT projects can be described by those adjectives. Project managers appreciate benefits of light approaches in comparison to the waterfall approach. The Agile approach is currently the mainstream movement in the management of software projects. In November 2013, the US Government announced that all IT projects undertaken by the Department of Defence must be Agile. It is nothing more than a confirmation of this fact, as usually, governmental organisations react slower than the private sector.

One of the most popular and commonly recognised Agile practices is the Scrum framework described in this paper. Scrum creators did not provide a measurement method for the process maturity. It is identified that to ensure the success of a project, the manager must be able to evaluate the state of the project during the implementation process, and verify it against the business environment and customer needs. To address this need, a complete approach is guaranteed by the Scrum Maturity Capsule, which, by the evaluation of Scrum roles (responsibilities) of the Product Owner, Development Team, and Scrum Master, provides a widespread verification of the Scrum framework implementation.

The Scrum Maturity Capsule can be used by most Scrum Teams. It addresses needs for the evaluation of the Scrum framework for an individual Team working on a software project development, as well as for many Teams involved in this process. Although the paper focused on a single Scrum Team, it would be recommended to verify the Scrum Maturity Capsule on more (preferably all) Scrum Teams working on same projects. An especially interesting aspect could be analysed, namely, what strategic decision was taken based on the observation from the introduced Scrum Maturity Capsule, and how it helped to mitigate project failure.



7th International Conference on Engineering, Project, and Production Management (EPPM2016) was co-organised by the Agency for Restructuring and Modernisation of Agriculture (Poland).

## References

- [1] Kowalczyk Z., Orłowski C., *Knowledge management based on formal models of software project management*, Cybernetics and Systems: An International Journal, 6, 8, 611–626, 2007.
- [2] Kowalczyk Z., Orłowski C., *Advanced Modeling of Management Processes in Information Technology*, Springer, 2014.
- [3] Pastuszek J., Orłowski C., *Model of rules for IT organization evolution*, Transactions on Computational Collective Intelligence IX 7770, 55–78, 2013.
- [4] Schwaber K., *Agile Project Management with Scrum*, Microsoft Press, 2004.
- [5] Sitek T., Orłowski C., *Ocena technologii informatycznych – Koncepcja wykorzystania systemów inteligentnych*, [in:] Komputerowo zintegrowane zarządzanie, No. 2, R. Knosala [Ed.], Oficyna Wydawnicza Polskiego Towarzystwa Zarządzania Produkcją, Opole, pp. 153–159, 2007.
- [6] Sitek T., Orłowski C., *Model of Management of Knowledge Bases in the Information Technology Evaluation Environment*, [in:] Information Systems Architecture and Technology: Models of the Organisation's Risk Management, A. Grzech, L. Borzemski, J. Świątek, Z. Wilimowska [Eds.], Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, pp. 221–231, 2008.
- [7] *Agile Software Development*, available: [http://en.wikipedia.org/wiki/Agile\\_software\\_development](http://en.wikipedia.org/wiki/Agile_software_development), last accessed 2nd December 2013.
- [8] *Version One, Agile Methodologies for Software Development*, available: <http://www.versionone.com/Agile101/Agile-Development-Methodologies-Scrum-Kanban-Lean-XP/>, last accessed 12nd January 2014.
- [9] *Heraclitus*, available: <http://en.wikiquote.org/wiki/Heraclitus>, last accessed 12nd January 2014.
- [10] James M., *Introduction to Scrum*, available: [http://scrumtrainingseries.com/Intro\\_to\\_Scrum/Intro\\_to\\_Scrum](http://scrumtrainingseries.com/Intro_to_Scrum/Intro_to_Scrum), last accessed 2nd November 2013.
- [11] Beck K. et al., *Manifesto for Agile Software Development*, <http://agilemanifesto.org/>, last accessed 12nd January 2014.
- [12] Schwaber K., Sutherland J., *Changes between 2011 and 2013 Scrum Guide*, available: <https://www.scrum.org/About/All-Articles/articleType/ArticleView/articleId/724/Scrum-Update-2013>, last accessed 12th December 2013.
- [13] Scrum Alliance, *Why Scrum*, available: <http://www.scrumalliance.org/why-scrum>, last accessed 2nd November 2013.
- [14] *Survey Monkey*, available: <http://en.wikipedia.org/wiki/SurveyMonkey>, last accessed 2nd December 2013.