Agile Performance Indicators for Team Performance Evaluation in a Corporate Environment

Extended Abstract

Cihangir Ertaban
Turkcell
Ozyegin University
Turkey
cihangir.ertaban@turkcell.com.tr

Erkan Sarikaya
Turkcell
Turkey
erkan.sarikaya@turkcell.com.tr

Selami Bagriyanik
Turkcell
Bahcesehir University
Turkey
selami.bagriyanik@turkcell.com.tr

ABSTRACT

Software development is a must for almost all industries including services, production, health and even construction. Being so widespread, software development industry needs metrics for especially two reasons; performance evaluation of development teams and continuous improvement. Moreover, use of metrics and measurements provides the ability to understand the problems and waste in the value stream so that they can be eliminated. This paper proposes a model of metrics -to be called as Agile Performance Indicators- which is also being developed and tested in the largest Digital Operator in Turkey.

CCS Concepts

• Software and its engineering~Agile software development

KEYWORDS

Agile Metrics, Agile, Agile Performance Indicators, Performance Evaluation, KPI, Scrum, SP, FSM, CFP, Velocity, Quality, Defect Density, Lead Time, Customer Satisfaction.

1 INTRODUCTION

Today, most industries use software development as a critical strategic differentiation instrument in their business. Some opinion leaders in industry even claim that every business will be a software business eventually [1,2]. There are large companies (like the one mentioned in this paper) employing

hundreds of development teams. In such cases, performance evaluation of these teams and also the individuals in the teams is a key aspect.

Secondly, improvement is anything that makes the current situation better and continuous improvement is making small changes collaboratively to reach a more efficient and effective state [3]. In order to establish a common understanding about a situation being better or worse than another; it must be quantified. It is only possible to quantify the situation if an accepted set of well-defined measures is followed.

For both cases, defining and estimating agile metrics is a challenge itself [4]. In this paper, we propose a set of metrics: Agile Performance Indicators (API) for agile teams and organizations including usage samples.

2 AGILE PERFORMANCE INDICATORS

2.1 Principles

Only a collaborative approach of the organization, management and the teams may result in successful use of metrics. In case of miss usage of these performance indicators such as comparing teams and employees, going too strict on minor changes, focusing too much on just numbers; metrics may become meaningless since teams may start inflating them intuitively.

In order to guide the teams and the organization, maximize transparency level, improve organizational alignment, and extract the most benefit from using the API's, we first propose a set of principles, which may be adopted and declared to the teams as ground rules.

These principles are a starting point to adopt not only Agile Performance Indicators, but any metrics or measures in lean and/or agile software development.

2.1.1 API's should be chosen together with the teams. Agile Performance Indicators include a set of agile metrics for the teams and the organizaton. The organization may choose a general set for all the teams; but still, the self-organized agile teams should be able to debate and choose among the remaining metrics which are left as optional.

2.1.2 Teams should not be compared. Although some metrics may allow comparison of different teams; comparing teams may be the biggest mis-usage. As an example, velocity (and even velocity per-person) is based on the teams own story point

definition and comparing velocity of two different teams is solely wrong.

2.1.3 Only the API's that will be followed should be chosen. API's provide a number of metrics but only the ones to be followed should be chosen. Any metric that will not be inspected is a waste of time and trust, for the team and the organization.

2.1.4 API's should be followed by all levels in the organization. Another problem with using metrics and performance indicators is the difference between the squad's goals and measures and top management's targets –if there is-. Eliminating this difference is key to align the squads and the organization.

2.1.5 API's should be aligned with the organization's strategy. Every organization has a strategy. Since using API's is a good way of targeting the agile team towards the organization's strategy, API's should be aligned towards the strategy. For instance, if the organization aims perfect quality over anything else, quality is the top and leading metric of the API's to be chosen.

2.1.6 The Organization should aim sustainable pace. Metrics make current situation available and help improvements; but the aim of the organization should be sustainable pace instead of always pushing so hard on the teams. Otherwise, strict measures may let the team members inflate numbers, manipulate graphs or become demotivated. Sustainable pace will always yield better software and products.

2.1.7 Revised when necessary. Welcoming change is a part of the Agile Manifesto [5]. So, API's should also welcome change and be revised by the team and the management in case there is a need for change.

2.2 Metrics

Once these principles are accepted, the teams may go on with the selection of the following metrics.

2.2.1 Production - Velocity. Velocity is the main measure for throughput [6]. It may be measured with Story Points or more systematic and objective functional size measurement methods such as Cosmic Function Points [7].

Velocity trend is a very important tool for the team to plan the next iterations in the most accurate way possible. This can be viewed as two different views, according to the ideal time estimation;

- Average velocity per team-sprint
- Average velocity per person-sprint

The main reason for person-sprint is not to compare different teams with each other, but to manage any possible differences of the same team size such as new team members or losing team members.

2.2.2: Production - Lead Time and Cycle Time. Another gauge for production is Lead Time [4] and Cycle Time. Lead Time is the total time between the ideation and realization of any increment and Cycle Time is the total time spent in the value stream [8]; which together are core measures for lean software development. Still, they may be crucial for agile frameworks.

2.2.3 Production – Distribution of Waste. Eliminating waste is a major aspect of Lean approaches. The teams using Kanban in the Digital Operator mentioned in this paper focus on the waste at any work passing through the board in order to analyze the bottlenecks and make improvements [4].

2.2.4 Quality – Number of Defects. Quality is another measure which should start at the beginning of any transformation. The most basic measure for quality is the number of defects trend [9]. The trend is expected to be decreasing by the progress in the sprints.

2.2.5 Quality – Defect Density. Defect density is like the gyroscope for number of defects between sprints; since the fluctuations between sprints may result in different number of problems. It is estimated by dividing number of problems to velocity of the team (Story points or function points) [10]. Considering the time lag between introducing a new defect and experiencing it by the users, a more cummulative approach may also be deployed.

2.2.6 Customer Satisfaction - # of Active Customers. One of the important sources of team motivation is to provide customer satisfaction with the value produced [11]. It is key for the team to understand and measure that produced value matches customer expectations. Number of customers is a core measure for customer satisfaction. In case the software is a product aiming for end users, number of active customers may even be the main target for the company.

3 PERFORMANCE MANAGEMENT

Performance management is based on individuals in most organizations. The company mentioned in this paper also has an individual based performance evaluation system. Unfortunately, this is a major obstacle in the agile transformation journey, because teamwork and team's common results is the main goal of agile frameworks. When an individual has different targets than the team's roadmap, backlog or sprint; the individual will naturally have problems choosing his/her own side or the team's shared side. As a result, team performance and shared performance measures is very important for teams' adaptation of agile frameworks.

On the contrary, agile transformation may not reach a shift in the culture and the mindset in short term. Agile frameworks normally expect self-organized teams to detect and try to improve low performers and individuals disturbing teamwork. After trials without improvement, teams should take more serious actions. Hence, in practice, the teams may not act and decide in a self-organized way and ask for management to make decisions. Although not a perfect way, management involvement and decision making for performance evalution should be kept in the picture.

This paper proposes using Agile Performance Indicators as a shared team goal and keeping individual evalution by the managers in a combined way.

3.1 Combining Team and Individual Goals

According to our model, each individual has a performance score based on two goals: Team's shared goals and Individual goals. Team's shared goals are based on the API's chosen from the set

of metrics and followed according to the principles by the whole team. The team is responsible for measuring and improving the trend. Team's shared goals have higher percentage on the individual; 70% as a base value.

Individual goals are directly assigned by the manager. They may include personal development of the individual, his/her support to the organization and subjective aspects, and they do not include tasks from the backlog (because backlog is covered in the team's shared goals). Individual goals are the part of the managers' control area. They have a lower percentage; 30% as a base value.

As an example for the model, an individual's performance evaluation result might look like:

- 10/10 for the shared team goals (70%)
- 7/10 for individual goals (30%)
- Combined result = (0.7*10) + (0.3*7) = 9.1/10

The model does not indicate any time frequency, so it may be once a year, once a quarter or whenever the organization decides. The important part is enhancing teamwork and keeping management support.

4 CONCLUSION

The main purpose of the model mentioned in this paper is to help lean and agile development teams evaluate their performance and reach a higher quality level with higher motivation. In order to strive for the better, the teams need to make experiments, get feed-back and take action. Understanding their exact position is a must for the team to apply Kai-Zen and make improvements. Moreover, in order to assess whether an action has created value and helped the team, the new situation should be measured, too. Agile Performance Indicators model provides a broad solution for measuring the team's situation so that improvements can be made.

Through the Agile Performance Indicators proposed in this paper, the team may choose all or some of the metrics and even combine them with some others like;

- Refactoring (Technical Debt Ratio)
- Unit Test Coverage
- Automated Test Case Coverage
- Code Quality Issues
- Security metrics
- Continuous Integration & Deployment Trends (CI & CD)

Lastly, Agile Performance Indicators are being developed and tested in one of the largest IT Organizations and Digital Operator company with more than 60 agile teams; which will provide broad feedback on the model. Tuning and improving the model is also part of further research.

REFERENCES

[1] Sargan, Cliff. 2015. Satya Nadella: Every business will be a software business. http://www.computerweekly.com/news/2240242478/Satya-Nadella-Every-business-will-be-a-software-business
[2] Mckendrick, Joe. 2017. Why every company needs to run like a software company. http://www.zdnet.com/article/every-company-needs-to-run-like-a-software-company
[3] Fritze, Christopher. 2016. The Toyota Production System - The Key Elements and the Role of Kaizen within the System.
[4] D. R. Greening. 2015. Agile Enterprise Metrics. 48th Hawaii International Conference on System Sciences, Kauai, HI, pp. 5038-5044.
[5] Beck, Kent; et al. 2001. Manifesto for Agile Software Development. Agile Alliance.
[6] S. Downey and J. Sutherland. 2013. Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft. 46th Hawaii International Conference on System Sciences, Wai-lea, HI, USA, pp. 4870-4878.
[7] Bağrıyanık, S., Karahoca, A., Ersoy, E.. 2015. Selection of a functional sizing methodology: A telecommunications company case study. Global Journal on Technology 7, 98–108.
[8] Mason-Jones. Rachel, and Denis R. Towill. 1999. Total cycle time compression and the agile supply chain. International journal of production economics 62.1: 61-73
[9] Hallowell, David L. Developing an Agile Planning and Tracking Scorecard. https://www.isixsigma.com/industries/software-it/developing-agile-planning-and-tracking-scorecard/
[10] Y. K. Malaiya and D. Jason. 1998. Estimating defect density using test coverage. Rapport Technique CS-98-104, Colorado State University.
[11] Cockburn. Alistair and Jim Highsmith. 2001. Agile software development, the people factor. Computer 34.11 (2001): 131-133.