Process Efficiency – Adapting Flow to the Agile Improvement Effort

Frank Verbruggen, Hi Efficiency, feamar@icloud.com
Jeff Sutherland, Scrum Inc., jeff@scruminc.com
Jan Martijn van der Werf, University Utrecht, j.m.e.m.vanderwerf@uu.nl
Sjaak Brinkkemper, University Utrecht, s.brinkkemper@uu.nl
Alex Sutherland, Scrum Inc., alex@scruminc.com

Abstract

In Scrum, we measure performance using velocity. However, the velocity of one team cannot be compared to the velocity of another, since it is a relative measure that is only of meaning to the team using it. So can we objectively measure the performance of teams?

Measuring Value Added Time as a percentage of Total Time is a metric that is used in Lean Manufacturing to help get a better understanding of production processes and optimize those processes.

This paper introduces an adaptation of this metric to the Agile environment. Giving teams an objective insight into their efficiency helps them optimize their efficiency and compare themselves to other teams. This adapted metric is called Process Efficiency and is comparable across teams, technologies, and domains of practice.

1. Introduction

Efficiency and effectiveness are measures of how good any entity can deliver value to its clients. While effectiveness focuses on doing the right thing, efficiency is focused on how quick that value is delivered. Faster delivery lowers the cost of production in many environments. A small reduction in cost can produce a significant gain in profitability. It can also increase effectiveness. The Chief engineers at Toyota drive 95% of the profitability by (1) designing a highly desirable product and (2) designing for low cost to increase market penetration [1]. In this paper, we focus on efficiency as a driver for delivering value to customers.

In the context of this paper, we define being efficient to mean having a low amount of waste in the production process. So, an increase in efficiency is directly linked to a decrease of waste. Waste is defined as any activity that does not add value to the product. Focusing on efficiency in this way is at the core of the Toyota Production System [2].

Whilst focusing on effectiveness is very important, in teams with very low efficiency focusing on efficiency could drive production up by a factor of up to 2098 [3]. This means that focusing on efficiency could be the difference between delivering value in 1 day or 6 years.

To give development teams insight into their efficiency, this paper introduces a metric called *Process Efficiency* and it is defined within the Scrum way of work [4]. Scrum derives from observations of Lean hardware teams [5]. Process Efficiency is derived from a standard metric used for decades in Lean Manufacturing - value-added work time divided by clock time [6].

In practice, Process Efficiencies exceed 25% for processes that have been improved through the use of Lean methods [6] whereas the average Scrum team Process Efficiency for completing a Product Backlog Item is on the order of 5-10% according to polls of participants in Scrum classes in the U.S. and Europe [7]. Measuring Process Efficiency can significantly improve the performance of Scrum teams as it is directly correlated with increase in team performance [4]. It also has the advantage of being independent of teams, technologies, or domain of work.

This paper defines how to measure Process Efficiency, allowing any Agile team to start using it right away and gain insight on their productivity. A previous study at a CMMI Level 5 Scrum company [4] showed that improving Process Efficiency doubled productivity. This was enabled with a checklist that determined whether a story was “Ready” to be brought into a sprint. Published research on this effect led to introduction of the concept of “Ready” in the Scrum Guide [8] and publication of a pattern called “Definition of Ready” by ScrumPlop.org [9].

The remainder of this paper is structured as follows. Based on a discussion of the current situation in Section 2 and an analysis of the problem in Section 3 we define the metric Process Efficiency and how to calculate it. Section 4 defines how to measure and calculate Process Efficiency. Initial measurements on Process Efficiency, presented in section 5 allows teams to study their production process. Section 6 lays out future steps to enable this. We conclude the paper with some final thoughts.

2. The current situation

Both governments and companies invest heavily in their development efforts. They spend large amounts of their budget on development of better services and products for their customers and civilians. The forecast for IT spending worldwide in 2018 is $3700B [10], about 4.4% of global GDP. Getting maximum value from such investments is in the interest of companies and society alike.

There are big differences in the productivity of development teams. Some data suggests that the difference in efficiency between an efficient team and an inefficient team can be as much as a factor of 2098, although a more realistic range might be to a factor of 310 [3].

Using only velocity [11]**,** it is difficult to know how efficient you are as a team as compared to other teams. Velocities of different teams are not comparable. This is because velocity is measured in story points, and every team can assign their own unique meaning to what a story point is and signifies. Therefore, velocity cannot give insight in the efficiency of a team as compared to other teams nor can it directly point out how to increase velocity.

Many optimization efforts focus on improving velocity [9]. However velocity is a relative metric. Consequently, if efficiency is low, a relatively high optimization in velocity still yields low waste reduction, and thus low Process Efficiency.

Consider a team that is improving its velocity by 10%. If it is a very inefficient team with 99.9% waste, then waste reduction is only 0.01%. On the other hand, a very efficient team with 60% waste, reduces waste by 4%. In both cases velocity is increased by 10%. However the difference is a factor of 400. Consequently, using only velocity as a metric does not provide enough insight to improve their production process. Therefore, we need an additional metric that gives more insight into team efficiency.

Process Efficiency as a metric can help both beginning and advanced teams. For beginning teams, in order to understand how well they are doing, they need to be able to compare their own efficiency to that of others. For a more advanced team, in order to further improve their process, they need to have an absolute measure to know where in the process they still can improve.

3. Understanding what is wrong

That we are experiencing low efficiency stems from a misconception. We assume that high efficiency is achieved by maximizing the utilization of all the parts. This idea is called resource utilization maximization and it is applied throughout society [12].

Because being efficient means having low waste in your production process, let's take a look at resources utilization maximization from the perspective of a single feature that needs to finish.

Consider the following example. BizzCorp, a supplier of a CRM solution CustomerFirst, has a team that is fully dedicated to maintaining CustomerFirst. Elizabeth, a customer, has informed BizzCorp about a bug in CustomerFirst.

The maintenance team does daily standups every morning, and during the daily standup they discuss what to do during the day, since they all want to be working.

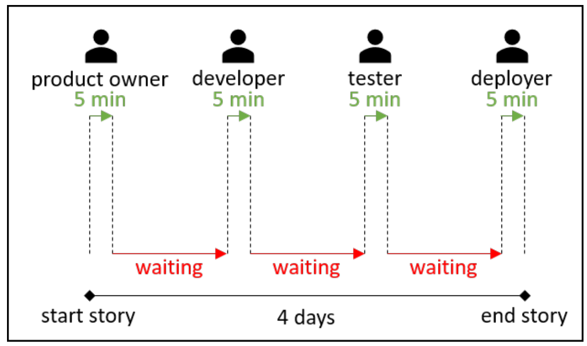

Figure 1: Inefficient process

Early that day, the bug is being reported by Elisabeth. About 4 hours later, the Product Owner checks the bug, and decides that it needs to be fixed. All the developers are busy with the work they took in during the daily standup, so no one can take the task, and they decide that it will be postponed to the day after.

The developer fixes the bug in 5 minutes of coding, and then wants to pass it on to the tester. The tester has already been fully booked for the day though, so testing will be postponed to the day after. Testing also takes 5 minutes and after approving the team decides to deploy it the day after, because the deployer was fully booked. Then finally can Elizabeth can get value from a bug free product again.

The problem in this example and so many others, is that by maximizing the utilization of resources, these resources work together in a poor way. And queue's and wait times start to appear.

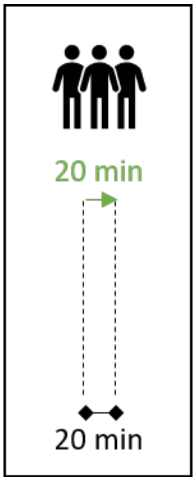

Figure 2: Efficient process

Instead, focusing on throughput maximization i.e. finishing stories as soon as possible results in better efficiency from the user's/client's perspective. To do anything and everything to make sure that a story spends as little time in progress as possible, so that Elizabeth has to wait the shortest time possible to get her bug fixed.

So the waste in this process is 31 hours and 40 minutes. And while in this example Elizabeth gets her value in 4 days, she could have also gotten it in 20 minutes. Had the team worked together in an efficient way, like in figure 2, they would have delivered value 95 times faster.

Lean Manufacturing uses this knowledge and focuses on elimination of waste to improve Process Efficiency [13].

4. Process Efficiency

This section introduces the Process Efficiency metric and how teams can measure it. It is a relative metric on a scale from 0 to 100% that can be used to do absolute comparisons. The 0% efficiency represents products that never go to production, where a 100% means complete uninterrupted focus from start to end on adding value for the customer. Process Efficiency is interesting for any development team that wants to improve on their performance. It adapts a metric that is well known in Lean Manufacturing. It does so in a way that is natural for Agile teams by giving teams insight in the amount of time that they are adding value and the amount of time they are wasting. In this section, we define the elements a team needs to measure to calculate their Process Efficiency for user stories.

The goal of introducing Process Efficiency in development teams is to reduce throughput time for stories and create focus on more rapidly fulfilling the needs of the customers of those teams. In this way, Process Efficiency fits well with teams that focus on Throughput Maximization, instead of Resource Utilization Maximization. As desired side effects, Process Efficiency promotes swarming [9] and doing detailed analysis on the waste in the development process.

4A. Measuring Process Efficiency

To help teams reduce waste and improve their delivery speed, they need to focus on measuring and optimizing Process Efficiency. Process Cycle Efficiency is a metric that comes from Lean Manufacturing, and it describes the relative amount of time spent on adding value vs. not adding value to the product (or story). From [6] we get that *Process Cycle Efficiency= Value Added Time / Total Lead Time*. We use this definition to mean Process Efficiency in this paper.

Adapting this to a development environment, we create a definition for software development teams to work with. Intuitively, two things have to be measured by teams to calculate Process Efficiency.

They are:

1. the total time the story was in progress, ending when the story is successfully put to production;
2. the total time that the team was prohibited from adding value to the story.

The first metric is Cycle Time. Cycle Time represents the total amount of time spent on realizing the story. Cycle Time is measured in hours, and only working hours count. So, work starting on Friday at 4 PM and ending at Monday at 10 AM has a Cycle Time of 2 hours (assuming a 9 AM to 5 PM working days).

To calculate Cycle Time, teams must keep track of when they start work on the story and when it's in production.

Cycle Time: *The total time the story was in progress, measured in working hours. Calculated as:*
Time Story In Production – Time Start of Story

The second metric is Interruption Time. Interruption Time represents the sum of all the times that the team got interrupted from adding value to their story. Interruption Time is also measured in hours, and only working hours count, similar to Cycle Time.

The reason to use interruption time instead of trying to measure value added time is twofold. First of all, either one of these metrics is good enough, they simply add up to the total time. The second reason is that we observe that people in teams don't like to measure strictly when they are working. In our experience they do like measuring when they are prohibited from working.

Activities that add value to the story are considered to be value added time. Activities that do not add value to the story are considered to be interrupts.

Since teams consist of many people, it is possible that at any one time, some people are adding value, while others are not. If the people that are not adding value cannot add value at that time, that is not waste. If however some of the people that can add value are not adding value, that is considered waste.

An interrupt can therefore be partial, when some people that could add value are not adding value. Without proper tools, the amount of effort to measure this in practice is too high (for teams that were part of the initial measurements this was reported as a problem or potential problem).

When approximating partial interrupts, we propose to measure the length of interrupts by defining that a team is interrupted when a team member that can add value to the story isn't adding value to the story. As this approximation allows for less ways to game the system.

To calculate Interruption Time, teams must keep track of when they are interrupted from adding value to the story, write down when that interruption started and write down when they can resume work again.

Examples include, but are not limited to:

1. Meetings;
2. Manual Testing;
3. Manual Deployment;
4. Working on something else than the story;
5. Non Team members that prohibit the team from adding value (e.g. security checks).

These examples are all potentially time consuming activities which do not add value to the story (over their automated counterparts respectively). Yet these activities are common in the workplace. Examples 1, 4 and 5 are literally doing something else than working on the story, while examples 2 and 3 are activities that do not fit with today's standards about Continuous Delivery [14]. If a team isn't forced to stop working on a story, but decides to not work on a story (e.g. because they are multi-tasking), that counts as interruption time.

Interruption Time: *The sum of the time the team was interrupted. Where each interruption is measured in working hours. Calculated for each interruption as:*

Time Work Resumed – Time Start of Interruption

Because of our definition of interruptions, whenever a team is not interrupted, they are adding value to the story. Similar to the Process Cycle Efficiency in Lean Manufacturing, Process Efficiency can be calculated as the fraction of Value Added Time and Total Time.

Process Efficiency: *The time spent adding value to the story as a percentage of the total time spent on the story. Calculated as:*

(Cycle Time – Interruption Time) / Cycle Time

So, for a team that has worked on a story from Monday 9 AM to Friday 5 PM the Cycle Time is 40 hours (given an 8 hour workday). Assume that the team had 5 meetings which took a total of 5 hours. And they had to wait for approval from an external party before they pushed their product to production for 10 hours. In this example the process efficiency is (40 – (5+10)) / 40 = 25 / 40 = 5/8 = 62.5%.

5. Initial Measurements: 4 companies, 5 teams

5 teams volunteered to help by collecting data on their process efficiency. The results have varied immensely.

Team Harold from the Port of Rotterdam had a story that they finished in 1 day. They kept close watch on their interruptions:

Team Harold	Start	End
Story 1	09:10	17:20
Daily Meeting	9:30	9:45
Discussions about priorities	10:00	10:35
Manual Deploy	16:00	16:25
Discussions about prod issues	13:00	13:35

Their cycle time for story 1 was 490 minutes. Their interruption time was 110 minutes. Making their process efficiency 380 / 490 = 77.6 %

Team Harold also provided some data about another story they had finished, which started on Wednesday and was put into production on Thursday.

Team Harold	Start	End
Story 2	13:50	10:35
Production issues	14:40	15:15
Helping other people	15:50	16:15
Analyzing pull requests	17:25	17:40

Daily Meeting	9:30	9:45

Their cycle time for story 2 was 285 minutes. Their interruption time was 90 minutes. Making their process efficiency 195 / 285 = 68.4 %.

Team Pronto, also from the port of Rotterdam, also wanted to participate and had a story that ran from 5-7-2018 to 5-9-2018.

Team Pronto	Start	End
Story 3	5-7-2018 8:00	5-9-2018 8:05
Blocked by other story	5-7-2018 11:00	5-8-2018 12:30
PR	5-8-2018 13:10	5-8-2018 13:35

Their cycle time for story 3 was 1100 minutes. Their interruption time was 655 minutes. Making their process efficiency 445 / 1100 = 40.5 %

A postal delivery company that wishes to remain anonymous provided data on 3 stories that they worked on. They did not keep track of what their interruptions were, but they did keep track of when they were:

	Start	End
Story 4	5-7-2018 9:00	5-15-2018 13:20
Inter 1	5-7-2018 9:00	5-14-2018 14:30
Inter 2	5-14-2018 16:45	5-15-2018 13:20

Their cycle time for story 4 was 3140 minutes. Their interruption time was 3005 minutes. Making their process efficiency 135 / 3140 = 4.3 %

	Start	End
Story 5	5-15-2018 9:00	5-22-2018 17:00
Inter 1	5-15-2018 9:00	5-15-2018 14:30
Inter 2	5-15-2018 15:00	5-15-2018 17:15
Inter 3	5-16-2018 9:00	5-16-2018 10:30
Inter 4	5-16-2018 11:00	5-16-2018 12:00
Inter 5	5-16-2018 13:00	5-22-2018 17:00

Their cycle time for story 5 was 2880 minutes. Their interruption time was 2655 minutes. Making their process efficiency 225 / 2880 = 7.8 %

	Start	End
Story 6	5-15-2018 17:45	5-25-2018 13:00
Inter 1	5-15-2018 17:45	5-25-2018 8:30
Inter 2	5-25-2018 9:30	5-25-2018 10:30
Inter 3	5-25-2018 12:00	5-25-2018 13:00

Their cycle time for story 6 was 3615 minutes. Their interruption time was 3465 minutes. Making their process efficiency 150 / 3615 = 4.3 %

A bank that wishes to remain anonymous also tracked their process efficiency. After several weeks, they came with the following data:

The story was first submitted to our team on 4-20-2018, and we processed the story to be ready to be put on our backlog until 5-16-2018. It was now ready to get pulled into our sprint, which it did on 5-23-2018. We had refined and estimated the story, but had to conclude that other stories that hadn't been finished yet from previous sprints took more time than we had anticipated. At the current moment 6-1-2018, we have decided to remove the story from the sprint.

Their process efficiency for this story so far is therefore 0 %.

Scrum Inc., a Scrum training and consulting company had their Webside team provide data from two stories that they worked on.

Team Webside	Start	End
Story 7	5-14-2018 9:00	5-15-2018 17:00
Working on interrupt	5-14-2018 11:00	5-8-2018 12:00
Working on interrupt	5-15-2018 9:00	5-15-2018 10:00
Working on interrupt	5-15-2018 13:00	5-15-2018 14:00

The cycle time for the story was 960 minutes. The interruption time was 180 minutes. Thus the process efficiency was 780/960 = 81.25%

Their second story was a smaller story that was able be delivered independently. The team completed the story in an uninterrupted four hour period.

Team Webside	Start	End
Story 8	5-17-2018 8:00	5-17-2018 12:00

As there were no interruptions the process efficiency was 100%. Small independent stories that can be driven to completion by the team in under a day will often result in high process efficiency.

6. Follow up

The next step is to get a large number of teams to participate in data gathering and plotting all of their Process Efficiencies. The expectation is that levels of Process Efficiency can be identified that separate teams from each other with a factor of about 10. The collection of the data will include metadata. We want to then check if certain attributes in the metadata correlate strongly with certain levels of Process Efficiency. All of these results will be published in the next paper.

Process Efficiency lends itself to benchmarking of this sort because it is a relative measure with a fixed range (0-100%). This percentage is not dependent on the size or complexity of the story. It is not dependent on team size, technology used or the domain of the team.

What we *can* do is measure the Process Efficiency of a lot of different teams and a lot of different stories. If we have sufficient data points, we can see how much Process Efficiency varies across many different teams and many different stories.

From such data, any team can see what their current Process Efficiency is, and compare it to other teams and stories. From this, any such team can know how efficient they are, and how much room for improvement exists.

Based on the conclusions and the experience gained, treatments will then be defined to help teams gain significantly in Process Efficiency. These treatments will be applied in a smaller group of teams that can stand to benefit from them, and results of applying the treatments on Process Efficiency will then be published in further papers.

7, Scrumban, Kanban, Continuous Flow

Scrumban was proposed by Corey Ladas in 2008 [15] to solve the problem of Scrum implementations that were not lean. Takeuchi and Nonaka [5] defined the term Scrum by looking at lean manufacturing teams so Scrum derives from the Toyota Production System [2]. Ladas proposed to fix bad Scrum by focusing on flow.

David Anderson formalized the Kanban process for software development in 2010 [16] and Henrik Kniberg [17] has nicely summarized the key features of Kanban and compared them to Scrum. Kanban has three primary attributes – make work visible, limit work in progress, and measure cycle time. Scrum adds a Team, a Product Owner, Daily Scrums, Sprint Planning, Sprint Review, and Retrospectives. In order to gain high performance Kanban has to have a team with the basic Scrum events. High performing Scrums today are doing continuous delivery, essentially deploying each backlog item as it is complete, so it looks like Kanban. The Scrum Patterns Group has shown how to setup a buffer for a Scrum team to enable interrupt driven work without losing performance [18]. Thus the current approach maximizing flow is now called DevOps, assuming DevOps means automating deployment using Scrum as it is at Amazon [19].

Kanban measures cycle time which is the clock time it takes for a unit of work to complete. It is not a good measure of process efficiency at the unit level. A WIP limit and cycle time do not clearly indicate whether a team is lean. By definition, lean = process efficiency > 25%. Process Efficiency helps self-organizing teams by giving them clear insight into where their time is spent and how to lean out their approach.

Similar to “Getting Things Done” [20], Process Efficiency helps people understand how much of their time is spent on other things than achieving the goal. Teams with low PE can set goals for increased productivity by seeing which goals others have successfully achieved in similar situations.

Process efficiency teaches the team that taking a story end to end without stopping, they will produce more and better work. This “single-piece continuous flow” is a core objective of the Toyota Production System and to achieve it, Taiichi Ohno told his Project Managers they needed to eliminate Kanban which he viewed as waste [21].

8. Conclusions

Process Efficiency is a team metric that helps teams get an insight in how efficient they are realizing their stories. It can vary greatly depending on a number of factors, ranging from 0% to 100%.

Process Efficiency can, together with Lean practices, help a team to improve their way of work and become more efficient.

By following many teams and tracking their Process Efficiency, we will create a benchmark for team efficiency. With such a benchmark available, teams can then see how well they are doing and identify how big the improvement to their process should be.

Process Efficiency will help teams understand how well they are currently doing, as compared to how well they could be doing. This gives teams a perspective that Story Points can never give.

9. References

[1] T. Sakai, *The Secrets Behind the Success of Toyota*: Privately Published, 2018.

[2] T. Ohno, *Toyota Production System: Beyond Large-Scale Production*: Productivity Press, 1988.

[3] W. Myers, "Why Software Developers Refuse to Improve," *Computer,* vol. 31, pp. 112, 110-111, 1998.

[4] C. R. Jakobsen and J. Sutherland, "Scrum and CMMI Going from Good to Great," in *Agile Conference, 2009. AGILE '09.*, 2009, pp. 333-337.

[5] H. Takeuchi and I. Nonaka, "The New New Product Development Game," *Harvard Business Review,* 1986.

[6] M. L. George, *Lean Six Sigma: Combining Six Sigma Quality with Lean Speed*: McGraw Hill, 2002.

[7] ScrumInc. (2018, 15 Jun 2018). *Scrum Courses and Training*. Available: https://www.scruminc.com/scrum-training/scrum-courses-list/

[8] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive Guide to Scrum, The Rules of the Game," scrumguides.orgMarch 2017.

[9] J. Sutherland, N. Harrison, and J. Riddle, "Teams that Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams," in *2014 47th Hawaii International Conference on System Sciences (HICSS)*, Waikoloa, HI, USA, 2014.

[10] J.-D. Lovelock, A. O'Connell, W. L. Hahn, A. Adams, D. Blackmore, R. Atwal, S. H. Hong, N. Gupta, and P. Niketa, "Forecast Alert: IT Spending, Worldwide, 1A18 Update," Gartner, Stamford, CT2018.

[11] J. Sutherland and J. Sutherland, *Scrum : the art of doing twice the work in half the time*, First Edition. ed. New York: Crown Business, 2014.

[12] N. Modig and P. Åhlström, *This is Lean: Resolving the Efficiency Paradox*: Rheologica Publishing, 2014.

[13] S. Shingo, *Fundamental Principles of Lean Manufacturing*: Productivity Press, 2017.

[14] N. Forsgren, J. Humble, and G. Kim, *Accelerate*: Revolution Press, 2018.

[15] C. Ladas, *Scrumban and Other Essays on Kanban Systems for Lean Software Development*. Seattle: Modus Cooperandi Press, 2008.

[16] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*: Blue Hole Press, 2010.

[17] H. Kniberg, *Kanban and Scrum - making the most of both*: lulu.com, 2010.

[18] ScrumPlop.org. (2018, 26 Aug). *Published Patterns: Illigitimus Non Interruptus*.

[19] R. Monica, "Amazon Deployment Data from Roy Monica, Head of Engineering Amazon Devices Demand Forecasting," Seatlle: Amazon.com, 2018.

[20] D. Allen, *Getting Things Done: The Art of Stress-Free Productivity, Revised Edition*. New York: Penquin Books, 2015.

[21] J. Coplien, "An Alternative to Kanban: One-Piece Continuous Flow," in *Scruminc.com* vol. 2018, ed. Cambridge, MA: Scrum Inc., 2011.