Implementing Value Stream Mapping in a Scrum-based project - An Experience Report

Nayla Nasir, Nasir Mehmood Minhas
SERL Sweden, Blekinge Institute of Technology, Karlskrona, Sweden
naylanasirminhas@gmail.com, nasir.mehmood.minhas@bth.se

***Abstract*—The value stream mapping is one of the lean practices, that helps to visualize the whole process and identifies any bottlenecks affecting the flow. Proper management of the value stream can significantly contribute towards waste elimination by categorizing process activities to be either value adding or non value-adding. Lean development focuses on the value through the elimination of waste. Adding value through embracing change and customer satisfaction are also the benefits of Scrum.**

This study reports our experience regarding the implementation of VSM with Scrum. We followed the action research method, with an objective to see if VSM can contribute to the identification and reduction of wastes in a Scrum-based project.

We identified a noticeable amount of waste even with strict compliance to the Scrum practices. On the basis of identified waste, their root causes, and possible mitigation strategy we have proposed a future state map, that could help improve the productivity of the process. The results of our study are encouraging, and we suggest that adoption of VSM with Scrum could add more value to the Scrum-based projects.

Keywords: Scrum, Value Stream Mapping, Waste

I. Introduction

The focus of software development organizations is to develop high-quality software at a lower cost, faster pace, and maximum customer value. To achieve this, organizations have shifted their approach from classical development to iterative releases. In early 90's people started raising their voices to curtain lengthy processes and documentation in software development. Agile and lean are the methodologies which supported this concept. Both Agile and lean have their own well-defined set of principles. Some of the principles are shared among the two, at the same time each has its unique principles and practices [1]. The main difference between agile and lean is that agile is a bottom-up approach, whereas lean supports top-down process [2].

Scrum is a popular agile development method. It is well suited where requirements are random and complex [5]. It is described as a development process for small teams, which includes a series of time-boxed development phases, "sprints", which typically last from one to four weeks. Scrum value the customer, and encourage the participation of the customer in the sprint meetings [10]. For the overall process improvement, lean software development emphasizes two fundamental principles, namely identification of waste in the process and considering interactions between the individual parts of the software process from an end-to-end perspective. [8].

The overall goal of lean development is to achieve a continuous and smooth flow of production with maximum flexibility and minimum waste in the process. All activities and work products that do not contribute to the customer value are considered waste. Identifying and removing waste helps to focus more on the value creating activities [8]. Waste is an activity, which will not provide any value addition to the final product or customer aspects [12]. The use of lean principles and practices with agile is not new, lean practices are used for continuous improvement in the agile process. The use of Kanban (a concept related to lean) with Scrum is one of the examples [14].

One of the important lean practices is value stream mapping(VSM). It is the process of directly observing the flow of items, and summarizing them visually [16]. VSM helps managers to understand the current operational conditions and recognize improvement opportunities to maximize the performance towards perfection. Value stream mapping focuses on activities that add value to a product, at the same time it identifies the activities that do not add value [4].

This study briefly describes the main concepts associated with VSM and presents an analysis of VSM implementation in a Scrum-based project. The focus of the study is to see if the VSM can be beneficial when used in Scrum. In our project, by identifying the value-added and non value-added time in the current state of the process, we were able to identify and prioritize the various type of wastes. Furthermore, we have proposed the mitigation strategies to reduce the identified waste and thus suggested a future state of the process. Our experiment shows that the use of VSM with Scrum adds more value when comparing to the effort utilized on it.

The rest of this paper is organized as: Section II describes the background knowledge and concepts regarding value stream mapping. Section III elaborates on the method used for the study, it also enlists the research questions investigated in the report. VSM process execution is presented in Section IV. Section V presents the findings of the research. Finally, conclusions are provided in Section VI.

II. Background

This section presents the background knowledge and key concepts regarding the value stream mapping. The subsequent sections describe the following concepts: *A*) The value stream mapping, and its purpose, and *B*) How to implement VSM.

A. Value Stream Mapping

Womack et al., [15] define the value stream mapping as: *"The process of charting out or visually displaying a Value stream so that improvement activity can be effectively planned."*
Value stream mapping, if implemented properly, can have significant contributions towards waste identification and increased process efficiency [8]. To draw a value stream map, the concept of value and the value stream should be clearly understood.

What is Value? Value is the end-goal and therefore the establishment of value parameters at the start of a project is the key to achieve improved productivity and customer satisfaction. Emmitt et al., [4] suggests the following two characteristics of value:

- The perception of value is individual and personal, and is therefore subjective. Agreement of an objective best value for a group will differ from the individuals' perception of value.
- Values will change over time.

These characteristics highlight the complexity of value and thus emphasize on having a clear understanding and establishment of value before doing anything else.

The Value Stream Rother and Shook [11] define value stream to be *"all the actions (both value added and non-value added) currently required to carry a product through the main flows essential to that product: 1) the production flow from raw material to customer, and 2) the design flow from concept to launch"*

The Flow and Flow Items: To look for a bottleneck in a production system, it is important to understand what flows through that system i.e. the flow items. All work that flows through a software value stream is characterized by one, and only one, of the following flow items[6].

- Features and Defects: If we consider feature additions and defect fixes as the flow items. We can characterize work across all the people and teams in a value stream as applying to one of these units. Having visibility into every process, we could identify exactly how many people were involved in creating, deploying, and supporting a particular feature. The same goes for a defect fix. [6]
- Work on Risks: Another kind of work that is invisible to users and is pulled through the value stream is the work on risks. It includes the security, and compliance work that must be scheduled onto development backlogs. This work competes for priority against features and defects. It is not pulled by the customer because the customer usually can't see it until it is too late. [6].
- Debt Reduction: The fourth type of work that can be observed is debt reduction. If automation is not done to reduce infrastructure debt, it could impede the future ability to deliver features. This work tends to be pulled by software architects.[6].

How to Measure Flow? A number of metrics can be found in literature that can be used as measures of software delivery flow. They include lines of code, function points, work items, story points, deployments, and releases.
Each of them captures a notion of value flow from a different perspective, and has its limitations when used to depict end to end flow. [6]

Metrics for VSM: Following are the metrics that are associated Value stream mapping and can be used as a reflection on the current state of a process[13]:

- Takt time: Takt time is the rate at which a company must produce a product to satisfy customer demand.
- Cycle time: The time that elapses from the beginning of a process or operation until its done.
- Total cycle time: The total of all cycle times for each individual operation or cell in a value stream. Ideally, if there is no waste then the Total cycle time equals total value-added time.
- Queue time: The time that a work unit will wait for a downstream operation to be ready to work on it.
- Lead Time: Number of minutes, hours, or days that must be allowed for the completion of an operation or process, or must elapse before a desired action takes place.[3]

Purpose of Value Stream Mapping: Value stream management aims for perfection by maximizing flow [11]. It contributes to continuously refine and adjust the software process to improve its performance in terms of lead-time, quality of the software product and reduction of change requests [8]. VSM allows to see the whole by letting you visualize more than a single process level. It not only allows to identify the waste but also helps to see the sources of waste by highlighting the bottlenecks [11].

B. Implementing VSM

Value stream management is a process for planning and linking lean initiatives through systematic data capture and analysis. Value stream management consists of eight steps [13].

1) Commit to lean:
The first step to improvement is willingness to change. Successful VSM implementation requires a commitment from management to maximize the visibility of operations and accept any changes as suggested by this process visualization activity.
2) Choose the value stream:
A flow item must be selected as a target for improvement. This will define the scope of the VSM process.
3) Learn about lean:
Having the knowledge of key lean concepts, such as maintaining a consistent work-flow, the lean waste and its types, and the importance of continuous improvement (kaizen) etc, will help to categorize the activities as value adding or non value adding, hence identifying the waste.
4) Map the current state:
The current state is determined by gathering information on current practices, and a current state map is drawn by visually presenting this information.

5) Determine the lean metrics:
On the basis of the current state map, the values for the lean metrics (i.e the lead time, the cycle time, the queue time) are derived to reflect the current efficiency of the process.
6) Map the future state:
Highlighting the non value adding activities helps to identify and prioritize wastes. A future state map is presented after addressing the root causes of waste with an intention to improve the process.
The Development of current and future state is an overlapping effort. Future state improvement ideas will come while working on current state, and working on future state may highlight important current state ideas that have been overlooked.
7) Create Kaizen plans:
The next step is to devise plans for achieving the future state by incorporating the changes as suggested in the future state map.
8) Implement Kaizen plans:
The final step is the implementation of Kaizen plans that describe how to achieve future state and how to cope with this transformation.

III. Research Methodology

We adopted action research as main research method for this study. Action research is a participatory approach concerned with developing practical knowledge [9]. This methodology leads to a better understanding of practical and theoretical outcomes of a process through direct participation, rather than based on perceptions and interests of an external researcher. We carried out a Scrum based project with a small team of six members. The participants were assigned to the following roles, 1) Scrum master, 2) product owner, 3) business analyst, 4) developers, and 5) QA. All the members of the team are the students of masters in software engineering.
The research activity is based on research questions presented in Table I.

TABLE I
Research questions and Objectives.

ID	Research Question	Objective
RQ1	Is VSM capable of visualizing the current state of a Scrum-based project?	To investigate if VSM can identify the current state (waste) in a Scrum-based project.
RQ2	What is the impact of using VSM in a Scrum-based project?	To determine the effectiveness of VSM, regarding waste reduction in the future state of a Scrum-based project.

A. Methods Used:

To extract knowledge about the VSM, we carried out a literature review and studied literature presented in various research articles. After having a clear understanding of VSM, we utilized the acquired knowledge to observe the outcomes of value stream management in a practical scenario. The steps taken for this study are illustrated in Figure 1.

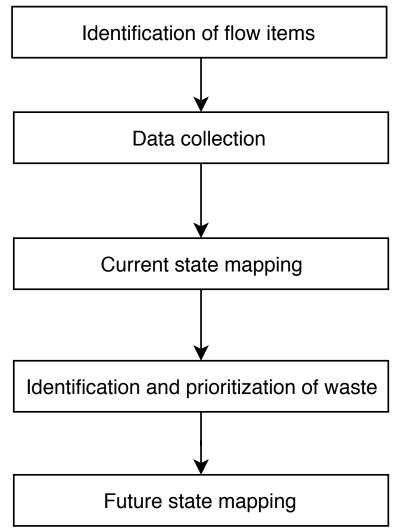

Fig. 1. Steps followed to implement VSM

The VSM activity was performed on a Scrum based process. The first author of the report was the member of the team executing the process, and served as a Scrum Master. There were two iterations in the process, but the perception of value changed for the customer after the demo of first sprint and the project scope was redefined. The tasks delivered at the end of the fist sprint were rendered as waste as the customer was not going to utilize the outcomes of that work. The sprint under observation consisted of 12 working days, and the value stream was mapped for the same period. A working day was considered to be of four hours as the team members were allocated fifty percent of their capacity to this project.

IV. Process Execution and Results

For the implementation of VSM, we followed the steps defined by Tapping et al. [13]. A brief description of these steps is presented in Section II-B. The steps adopted for our experiment are presented in Figure 1.

A. Identification of Flow Items

The first step is to determine the flow items for the value stream and use some measures for them. Kersten [6] defines four types of flow items (Features, Defects, Work on Risk, and Debt Reduction) that can be used for value stream mapping. Further, he emphasizes that only one of these four can be selected to draw a value stream. The flow items for the intended VSM are the features prescribed by the user stories, and these features are represented as tasks (PF-23 to PF-35), in the sprint backlog.

B. Data Collection:

An online Scrum management tool was used as a virtual Scrum board, for visibility and availability purposes. This tool was also used for time based data collection. All the team

TABLE II
SPRINT EXECUTION DETAIL

Sprint 2(26/09/2018 to 10/10/2018) ……(12 Working Days, Each working day=4 hours)									
DATE	ANALYSIS	Task ID	DESIGN	DEVLOPMENT		CODE REVIEW		QA	DONE
				CODING	READY TO REVIEW	IN PROCESS	WAITING FOR QA		
26/09/2018	Tasks defined, priorotized, assigned and added to sprint Backlog (4 hrs)								
27/10/2018		PF-23	PF-23 3:00 hr						
28/09/2018		PF-25	PF-25 01:00 hr	PF-23 1:00 hr					
		PF-27		PF-27 00:30 hr					
		PF-28		PF-28 00:30 hr					
10/01/18		PF-34	PF-34 3:00 hr	PF-25 01:00 hr	PF-23				
		PF-26		PF-26 00:30 hr	PF-27				
		PF-30		PF-30 00:30hr	PF-28				
		PF-29		PF-29 00:30 hr	PF-26				
		PF-33		PF-33 00:30 hr	PF-30				
10/02/18				PF-29 00:30 hr	PF-23				
				PF-33 00:30 hr	PF-27				
				PF-25 01:00 hr	PF-28				
					PF-26				
					PF-30				
10/03/18		PF-31		PF-31 02:00 hr		PF-23 01:00 hr			
				PF-25 00:00 hr		PF=27 00:30Hr			
				PF-33 00:00 hr		PF-28 01:00hr			
				PF-29 01:00hr		PF=26 00:30Hr			
						exception 01:00hr			
10/04/18		PF-32		PF-32 01:00 hr	PF-28		PF-26		
				PF-31 00:00 hr			PF-23		
				PF-25 01:00 hr			PF-27		
				PF-33 00:00 hr					
				PF-29 00:30hr					
				PF-30 00:30hr					
10/05/18		PF-32		PF-32 00:00 hr	PF-28		PF-26		
				PF-31 00:00 hr	PF-29		PF-23		
				PF-25 00:00 hr			PF-27		
				PF-33 00:00 hr					
				PF-30 00:00hr					
10/08/18				PF-32 00:30 hr	PF-28			PF=26 01:00Hr	
				PF-31 00:00 hr	PF-29			PF-23 01:00 hr	
				PF-25 00:30 hr				PF=27 01:00Hr	
				PF-33 00:30 hr					
				PF-30 00:30hr					
10/09/18				PF-32 00:30 hr	PF-28				PF-26
				PF-31 00:30 hr	PF-29				PF-23
				PF-25 00:00 hr	PF-32				PF-27
				PF-33 00:30 hr	PF-31				
				PF-30 00:30hr	PF-33				
10/10/18				PF-25 01:00 hr	PF-30	PF-32 00:30 hr			
					PF-25	PF-31 00:30 hr			
						PF-33 00:30 hr			
						PF-30 00:30hr		PF-32 00:30 hr	
						PF-29 00:30hr		PF-31 00:30 hr	PF-32
						PF-28 00:30 hr		PF-33 00:30 hr	PF-31
10/11/18						PF-25 00:30 hr		PF-29 00:30hr	PF-33
								PF-28 01:00 hr	PF-29
								PF-30 00:30hr	PF-28
								PF-25 01:00 hr	PF-30
									PF-25

members reported the actual time using the tool every time they worked on a task. This input was further validated in the daily stand-up meetings. The data on actual time spent on each task was also maintained in an excel file, which depicts the day to day progress (See Table II). The tasks are color-coded which makes it easy to visualize the progress of each task across different operations in the process. Time spent on each task was recorded, and if there were multiple tasks in an operation on a particular day, time spent on task switching was also recorded. There are columns to show the time for which tasks waited until they were retrieved by the next operation.

Presenting a detailed picture of sprint execution,the excel file can be used to conduct an analysis and identify the value added Time (VAT), as well as the non value added time (NVAT) per task, for each working day. The Data Collected during this phase is summarized in Table III.

C. Current State Mapping

This activity is completed in two steps, firstly on the basis of analyzed VAT and NVAT, we calculated the required metrics for VSM and then we generated the current state map.

a) VSM Metrics:

1) **Cycle Time:** The cycle time is a measure of the time required to complete one cycle of an operation or to complete a function. It consists of both value added and non-value added time within an operation. Value added time (VAT) is referred to the time spent on the activities that add some value to the customer. Whereas, non-value added time (NVAT) is the one that is spent on activities that could be essential for the process but adds no value to the customer. The cycle time is calculated for each task and the cycle time for the individual

TABLE III
CYCLE TIMES AND LEAD TIMES FOR THE CURRENT STATE

Tasks ID	Analysis Cycle time		Design Cycle time		Development Cycle time		Queue Time	Code Review Cycle time		Queue Time	Quality assurance Cycle time		Lead time
	NVAT	VAT	NVAT	VAT	NVAT	VAT		NVAT	VAT		NVAT	VAT	
PF-23	0,08	0,25	2	2	0,66	1,00	1,60	0,00	1,00	1,33	3,00	1,00	13,92
PF-25	0,08	0,25	0	4	1,88	5,00	0,00	0,00	0,50	0,00	0,50	1,00	13,21
PF-26	0,08	0,25	0	0	0,20	0,50	1,60	0,00	1,00	1,33	0,33	1,00	6,29
PF-27	0,08	0,25	0	0	0,66	0,50	1,60	0,00	0,50	1,33	0,33	1,00	6,26
PF-28	0,08	0,25	0	0	0,66	0,50	8,67	0,17	1,50	0,00	0,25	0,50	12,58
PF-29	0,08	0,25	0	0	1,28	2,50	5,00	0,17	0,50	0,00	0,25	1,00	11,03
PF-30	0,08	0,25	0	0	1,47	1,50	1,60	0,17	1,50	0,00	0,25	0,50	7,31
PF-31	0,08	0,25	0	0	1,02	2,50	0,00	0,17	0,50	0,00	0,83	0,50	5,85
PF-32	0,08	0,25	0	0	0,77	2,00	0,00	0,17	0,50	0,00	0,83	0,50	5,10
PF-33	0,08	0,25	0	0	1,88	2,00	0,00	0,17	0,50	0,00	0,83	0,50	6,21
PF-34	0,34	0	4	0	No progress here for these tasks								4,34
PF-35	0,34	0	0	0									0,34
Total	1,48	2,5	6	6	10,477	18	20,07	1,00	8	4,00	7,41	7,50	92,43
Cycle time(CT) per opertion	3,98		12		28,48			9,00			14,91		68,37
Average CT/operation=		0,33	4		2,848			0,90			1,49		

Total VAT for the system(hrs)=	42,00	Takt time= Total working hours/no. of tasks= 48/12= 4 hrs	
Total NVAT for the system (hrs)=	26,37		
Total Queue time for the system (hrs)=	24,07	Product cycle efficiency=	45%

operations is presented as a sum of cycle time for each task. For example,referring to Table III, the analysis was performed by the product owner in one working day (i.e. 4 Man-Hrs). The cycle time calculated to be 4 hours, of which 1.48-hour is the NVAT and 2.5-hour is the VAT. In Table III VAT for PF-34 and PF-35 is zero because these tasks were unfinished from the customer perspective as these tasks did not progress to the subsequent operations (i.e., Development to QA). Cycle times for other operations is Design 12 hours, Development 28.48 hours, Code review 9 hours, and Quality Assurance 14.91 hours.

2) **Average Cycle Time:** As the cycle time for each task varied depending upon the complexity of the task, so an average cycle time of an operation is calculated by dividing the cycle time of the operation by the number of tasks. For instance, in the development operation the cycle time was 28.47 and the total number of tasks in the operation were 10. So, the average cycle time is calculated to be 2.8. Average cycle time for each operation is presented in Table III.
3) **Total Cycle Time**:
The Total cycle time for the system is calculated by adding the cycle time of all operations (analysis to quality assurance). Total average cycle time calculated for our system is 68.37 hours.
4) **Queue Time:** Queue time is the time for which a task waits between two successive operations. There were two queues in our system, one between development and code review and the other was between code Review and Quality Assurance. The task with the highest queue time is PF-28, that waited for code review for 13.67 hours, because of its interdependence with two tasks (i.e., PF-29 and PF-30). The Queue time is calculated for each task, and the total queue time between development and code review is 20.07 hours, and the total queue time between code review and quality assurance is 4 hours. The queue time for the whole process is 24.07 hours.
5) **Lead time:** Lead time is the measure of time required to perform all the activities from the initiation of a task until it is completed. The lead time includes both the queue time and the cycle time. If there is no waiting time between the operations , then the lead time equals the total cycle time for the process.
In our project the lead time is calculated to be 92.43, it is the sum of the total cycle time and the total queue time.
6) **Takt Time**: The Takt time is used to synchronize the rate of production with the rate of demand. It establishes a cadence for the work flow, and contributes towards efficiency through continuous adjustments. It is computed by dividing the available time by the number of tasks. There were 12 working days with four hours for each day (i.e., 48 work hours), and tasks to be accomplished were 12. The Takt time is calculated to be 4 hours (total work hours / total number of tasks). That indicates that each task needs to be worked by the team for four hours.

b) The Current State Map: A Current State Map was generated based on the data presented in Table III, and is shown in Figure 2. We considered the customer as the start point for our CSM because the first step in value stream management is to establish a clear perception of the value as defined by the customer. The process of implementing the current state map was initiated as soon as an agreement was made on user stories. Each cell of the current state map represents an operation. The number of people carrying out the operation is also mentioned within each cell. An activity is considered to be value adding if it contributes to customer value, and all non value adding activities are considered a waste. The value added time (VAT) for an operation is represented by the trough of the time-line and the crust shows the waste represented as a sum of NVAT and queue time. The

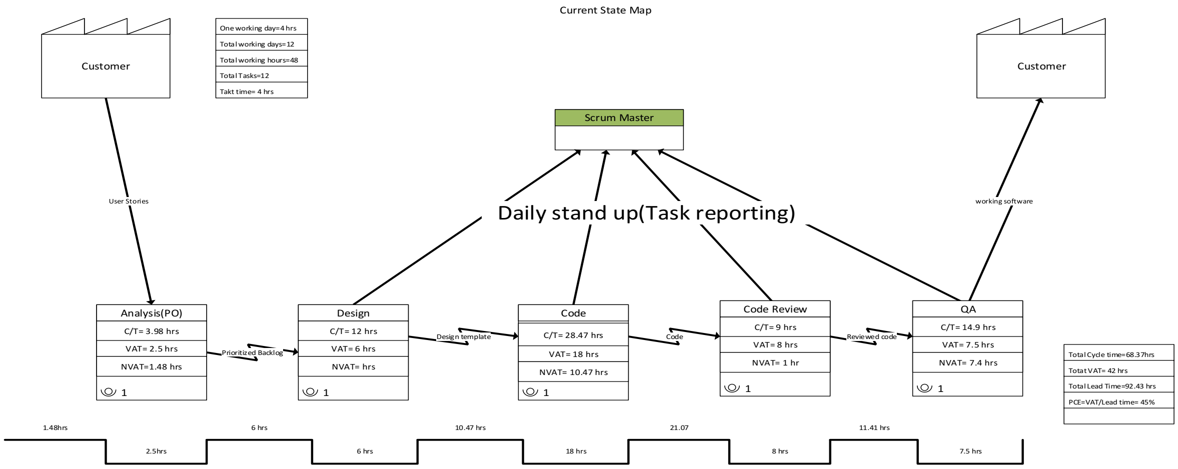

Fig. 2. Current State Map

TABLE IV
WASTE IDENTIFIED AND PRIORITIZED IN TERMS OF IMPACT

Priority	Type of Waste	Task Involved	Impact	Root Cause	Mitigation Strategy
1	Queue Time (Delays & Movements)	PF-23, PF-26, PF-27, PF-28, PF-30	24.07 Hrs.	i)Upstream operations are producing items at an inconsistent rate and the downstream operations cannot pull them efficiently, creating a push. ii) Interdependence of tasks	i) Merge operations where possible (e.g., coding and coding review) ii) Reduce task inter-dependency.
2	Task Switching	PF-25, PF-29, PF-30, PF-31, PF-32, PF-33	21.69 Hrs.	Multitasking within operations due to interdependence of tasks	Use work in progress (WIP) limit
3	Partially done work	PF-34, PF-35	4.68 Hrs.	Time constraint	Eliminate higher priority waste
4	Defects	PF-30	0.5 Hrs.	Exception handling was not addressed during development	Pair Programming

flow of information between cells is in electronic form.The takt time , lead time and process cycle efficiency are also presented in the CSM. The purpose of VSM is to visualize the complete value stream, and it should represent the complete work flow, right from the inception of demand to the delivery of the product to the customer. So customer is considered to be the endpoint for the VSM discussed in this study.

D. Identification of Waste

There are seven types of waste in Lean, as identified by Taiichi Ohno – the founder of the Toyota production process [7]. Cawley et al. [1] presented a connection between waste in lean manufacturing and waste in Lean software development. They listed the following wastes for the software development: 1) Extra features/code, 2) Delays, 3) Task switching, 4) Extra processes, 5) Partially done work, 6) Movements, 7) Defects, and 8) Unused employee creativity. On the basis of this categorization, we identified four waste categories in our project, namely queue time (delays and movement), task switching, partially done work, and defects. Identified waste categories along with the tasks involved are presented in Table IV. We have prioritized the waste on the basis of impact (in terms of time). In our project the top priority waste is the queue time with 24.07 hours, and the lowest priority waste is defects with 0.5 hours. We also identified the causes of the waste regarding our project (See TableIV). Finally, we listed the possible solutions to reduce the identified waste.

E. Future State Mapping

The future state map suggests improvements in the process. The goal is to prepare a possible course of action to address the root causes of the waste found in the current state of the process. In the light of solutions presented in Table IV, we reviewed the current state and suggested an ideal state of sprint execution with significant reduction in lead time, queue time and NVAT. For instance, to eliminate Queue time we can merge two operations, i.e., Code and Code Review by using the concept of pair programming. This will help to rectify errors during the development operation, and the delays between these two operations can be removed. It will also help to reduce the defect rate. Similarly, our next priority was to address the issue of task switching. For this, we can define the limit for work in progress(WIP) within an operation at any given time. Limiting work in progress can regularize the flow and will result in less multitasking and a notable reduction in NVAT within the operations. We propose to set this limit to a maximum of three tasks in development and quality assurance. It will create a pull by allowing the retrieval of a task when a successive operation is ready to work on it. Subsequently, the queue time between operations can be reduced to zero.

TABLE V
CYCLE TIME AND LEAD TIME FOR FUTURE STATE

Tasks ID	Analysis		Design		Code+ Review		Queue Time	Quality assurance		Lead time
	Cycle time		Cycle time		Cycle time			Cycle time		
	NVAT	VAT	NVAT	VAT	NVAT	VAT		NVAT	VAT	
PF-23	0,08	0,25	2	2	0,5	2	0	0	1	7,83
PF-25	0,08	0,25	0	3,5	1,3	5,5	0	0,33	1	11,96
PF-26	0,08	0,25	0	0	0,3	1	0	0,33	1	2,96
PF-27	0,08	0,25	0	0	0,3	1	0	0,33	1	2,96
PF-28	0,08	0,25	0	0	0,12	1,5	0	0	0,5	2,45
PF-29	0,08	0,25	0	0	0,25	3	0	0,08	1	4,66
PF-30	0,08	0,25	0	0	0,13	3	0	0	0,5	3,955
PF-31	0,08	0,25	0	0	0,25	3	0	0,08	0,5	4,16
PF-32	0,08	0,25	0	0	0,75	3	0	0,08	0,5	4,66
PF-33	0,08	0,25	0	0	0,75	2,5	0	0	0,5	4,08
PF-34	0,08	0,25	0	4	0,75	3	0	0	1	9,08
PF-35	0,08	0,25	0	0	0,5	2	0	0	1	3,83
Total	0,96	3	2	9,5	5,90	30,5	0	1,23	9,5	62,585
Cycle time(CT) per opertion	3,96		11,5		36,40			10,73		62,585
Average CT/operation=	0,33		3,83		3,03			0,89		8,090
Total VAT for the system(hrs)=			52,5	Takt time= Total working hours/no. of tasks=				4,00		
Total NVAT for the system (hrs)=			10,09							
Total Queue time for the system (hrs)=			0	Product cycle efficiency=			84%			

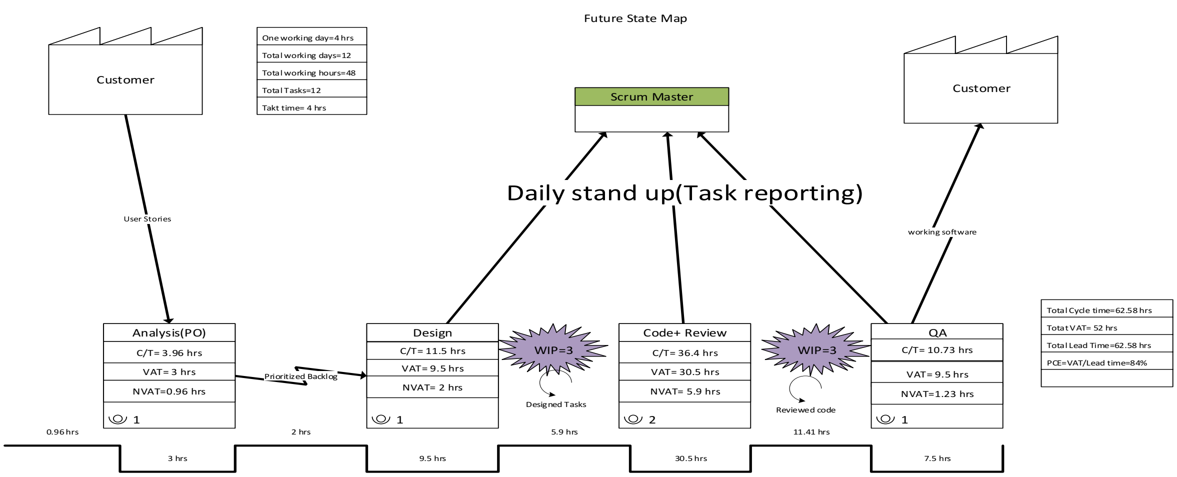

Fig. 3. Future state map

Finally, there will be no partially done work as there will be enough time to complete all the tasks. The suggested changes are presented in the Table V. It can be seen that the lead time is equal to the total cycle time as there are no delays between the operations. The proposed future state map is provided in Figure 3. The map includes calculations for takt time, lead time, and process cycle efficiency. The customer is the start, and the end point of the future state map as well.The cells symbolize the operations within our process, along with the details of cycle time and the number of workers for each operation. The Kaizen burst symbol represents suggested improvements. We used it between design and development, and then development and QA, to represent the improvements we have proposed in the form of WIP. Also, the flow between these operations is represented by a pull symbol instead of a push.

V. DISCUSSION

RQ1: IS VSM capable of visualizing the current state of a Scrum-based project?
The first objective of this study was to investigate the capability of VSM regarding the identification of current state (waste) in a Scrum-based project. The current state map presents the current set of activities been performed during a process, and is identified on the basis of detailed study of the actual process. We applied the action research method and carried out a project with a team of six people practicing Scrum as their development methodology. Being aware of the overall process flow, the Scrum Master (first author of the study) was responsible for value stream management. Required data for VSM was extracted from an online Scrum board with a little effort, and its validation was the part of every day stand-up. The detailed analysis of activities in terms of time, and value to the customer, helped to categorize them as value adding or

non value adding. Also some bottlenecks were highlighted that affected the overall productivity and increased the lead time for the process . The data of the current state is presented in the Table III, and visual representation of the current state is presented in Figure 2.

The current state map helped us to recognize the waste within and between the operations. We categorized the waste by using the list of waste provided by Cawley et al., [1]. The current process cycle efficiency is 45%, which could be improved by working on wastes. From our experience, we can conclude that implementation of VSM in a Scrum-based project does not require any additional effort, as most of the required data is available within the Scrum management process. Furthermore, we infer that VSM could be utilized with Scrum without affecting its practices.

RQ2: What is the impact of using VSM in a Scrum-based project?

The second objective of the study was to determine the effectiveness of the VSM, regarding waste reduction in the future state of a Scrum-based project. VSM not only helped us to identify the bottlenecks in the process, but also highlighted the root causes of the waste and allowed us to prepare a mitigation strategy. Table IV presents the prioritization of identified waste along with the root causes and mitigation strategy for waste reduction in each category. By utilizing the proposed mitigation strategy, we have created a future state map for the process. A set of data was created for an ideal state of sprint execution, see Table V, and this data was used to propose the future state. The visual representation of future state is shown in Figure 3. The future state map, shows a significant reduction in the lead time (i.e.92.43 hours to 62.58 hours), while also completing the partially done tasks in the current state of the process. This implies more productivity and thus a more satisfied customer, which is a primary goal of agile software development.

The waste is also reduced (i.e., 50.44 hours to 10.09 hours) by utilizing the team to maximize the time spent on value adding activities and eliminating the waiting time between process. The process cycle efficiency has improved visibly from 45% to 84% which implies a better utilization of available resources. The results we obtained by the implementation of VSM in our project are encouraging. Considering the difference, while comparing wastes and productivity in the current and the future state of the process, it can be stated that VSM can serve as an effective method to improve the efficiency of a Scrum team to an even higher level of performance. Value stream management allowed the team to realize impediments in the continuous flow of tasks, and to discover new ways of becoming more effective.

VI. Conclusion

This paper presents a research activity employing the value stream management to determine its effectiveness as a waste reduction mechanism in Scrum. It was inferred that VSM can be effectively used to reflect the overall flow, to identify the bottlenecks and to distinguish between value adding and non value adding activities.

Significant amount of waste was identified, along with its root causes. Based on this knowledge, mitigation strategies were defined to reduce or eliminate waste so as to increase efficiency, and to deliver tasks with better quality and less lead time. Addressing the wastes can contribute towards sustainable and more predictable development cycles and can enhance the Scrum experience. It was also observed that implementation of VSM in a Scrum based project was a fruitful activity as it requires very little effort when compared to its contribution to overall process improvement.

Based on current state of the process, we have proposed a future state map. Our next focus will be to identify the steps needed to develop the kaizen plans for the implementation of the future state of our process. And then to measure the success as achieved by the implementation of these plans in a practical scenario.

Acknowledgement

The authors would like to thank all the participants of the project. This work has been supported by EASE, the Industrial Excellence Centre for Embedded Applications Software Engineering (reference number 2015-03235).

References

[1] Oisín Cawley, Xiaofeng Wang, and Ita Richardson. Lean software development–what exactly are we talking about? In *Lean Enterprise Software and Systems*, pages 16–31. Springer, 2013.
[2] Robert N Charette. Challenging the fundamental notions of software development. *Cutter Consortium, Executive Rep*, 4, 2003.
[3] Business Dictionary. Lead Time. http://www.businessdictionary.com/definition/lead-time.html, 2018. [Online; accessed 10-oct-2018].
[4] Stephen Emmitt, Dag Sander, and Anders Kirk Christoffersen. Implementing value through lean design management. In *Proceedings of the 12th International Conference*, pages 361–374, 2004.
[5] Harleen K Flora and Swati V Chande. A systematic study on agile software development methodologies and practices. *International Journal of Computer Science and Information Technologies*, 5(3):3626–3637, 2014.
[6] Mik Kersten. What flows through a software value stream? *IEEE Software*, 35(4):8–11, 2018.
[7] Taiichi Ohno. *Toyota production system: beyond large-scale production*. crc Press, 1988.
[8] Kai Petersen and Claes Wohlin. Software process improvement through the lean measurement (spi-leam) method. *Journal of systems and software*, 83(7):1275–1287, 2010.
[9] Peter Reason and Hilary Bradbury. *Handbook of action research: Participative inquiry and practice*. Sage, 2001.
[10] Linda Rising and Norman S Janoff. The scrum software development process for small teams. *IEEE software*, 17(4):26–32, 2000.
[11] Mike Rother and John Shook. Learning to see. *Lean Enterprise Institute, Cambridge, MA*, 1999.
[12] RS Russell and BW Taylor. Operations management, 2nd edn, 1999.
[13] Don Tapping, Tom Luyster, and Tom Shuker. *Value stream management: Eight steps to planning, mapping, and sustaining lean improvements*. Productivity Press, 2002.
[14] Xiaofeng Wang, Kieran Conboy, and Oisin Cawley. "leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6):1287–1299, 2012.
[15] James P Womack, Arthur P Byrne, Orest J Fiume, Gary S Kaplan, and John Toussaint. Going lean in health care. *Cambridge, MA: Institute for Healthcare Improvement*, 2005.
[16] Jim Womack and Dan Jones. *Seeing the whole value stream*. Lean Enterprise Institute, 2011.

