



Copyright © IEEE.
Citation for the published paper:

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of BTH's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by sending a blank email message to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Waste and Lead Time Reduction in a Software Product Customization Process with Value Stream Maps

Shahid Mujtaba, Robert Feldt, Kai Petersen
School of Computing
Blekinge Institute of Technology
Box 520 SE-372 25 Ronneby, Sweden
Email: smu,rfd,kps@bth.se

Shahid Mujtaba, Kai Petersen
Business Unit Multimedia Products
Ericsson AB
Box 518 SE-371 23 Karlskrona, Sweden
Email: shahid.mujtaba,kai.petersen@ericsson.se

Abstract—Custom-developed adaptations of software products are increasingly important to meet different and changing customer needs and heterogeneous system environments. Efficient software customization processes with short lead times are thus a priority for companies to stay competitive. The purpose of this case study is to identify waste-related problems in a software product customization process by using value stream maps (VSM). The study was conducted at the telecom company Ericsson AB; the empirical data was collected using document analysis, extraction of phase times from a requirements tracking tool and interviews. It was used to construct a value stream map that showed the present state of the process. The map was then used in interviews with key stakeholders where they identified waste and proposed measures to avoid them. These solution proposals were subsequently used to construct a future value stream map showing a reduced lead time. Static validation showed that the VSM methodology was useful and that the identified solutions were of value. Our results have applicability for software engineering process improvement in general and for customization processes in particular.

Keywords-Value Stream Mapping; Value Stream Analysis; Software Development Wastes; Software Product Customization; Software Process Improvement; Lean Software Development;

I. INTRODUCTION

Software product customizations (SPC) are customized solutions that the customers want to be implemented in their current software release. They are typically of high priority and cannot wait to be implemented in the next main release of the overall product. This is specifically challenging in dynamic markets, like the telecom market, where companies have to deliver new customizations with short lead times before market needs change and the customizations become obsolete. In consequence, customizations are pre-dominantly developed in parallel to main product development. However, at the same time efficient customization handling are an important business case that can give a competitive advantage, i.e. urgently fulfilling the individual needs of the customers.

Reducing lead times is generally important in software development, but even more in product customizations. There are several reasons why one should care about SPC

lead time. For one, short lead times reduce the risk of market-lockout [1]. In the SPC context that means the company provides a customized feature to the customer before a competitor introduces it into their products. Since customization are smaller and can be developed in parallel and with more speed than the main releases efficient SPC handling can become the main means for market adaptability. Secondly, an early enrollment of a new product increase the probability for market dominance [2]. SPCs allow to react on current market needs that were not identifiable in the normal requirements engineering process and thus are an important source to develop relevant features before the competitors. Thirdly, another benefit of being early on the market is that the product conforms more to the expectations of the market [3]. That is, if the SPC is delivered close in time to the actual request for it, it is more likely to fulfill the need of the customer.

A preliminary investigation of SPC challenges at Ericsson revealed some problems with long lead times, which motivated the need for this case study. The case analysis showed that SPC challenges are not just concerned with development issues, but also issues related to unnecessary delays that get in the way of regular SPC development and affects lead time. Therefore, a decision was made to visualize and analyze the SPC value stream and to propose possible measures to optimize it. This further motivated the need to employ VSM as a lean methodological tool for doing the value stream analysis.

Lean software development provides a contemporary approach to software engineering management together with a set of practical tools to improve process efficiency. The concept of waste is fundamental to lean thinking, and any efforts to systematically eliminate or reduce waste in a software development process can help to reduce lead time (shorten time-to-market), and improve customer satisfaction [4]. The tool used for uncovering and eliminating waste is ‘Value Stream Mapping’ (VSM) [4]. A value stream is all the actions (both value added and non-value added) currently required to bring a product through the main process steps to the customer a.k.a end-to-end flow of process. It provides

managers the ability to step back and rethink their entire development process from a value creation perspective. Generally the biggest delays or loopbacks in a value stream provide the biggest opportunity for improving the process capability [4].

The goals of the case study are to, (a) perform value stream analysis to identify and analyze wastes in an industrial software product customization process, (b) gather solution proposals for waste elimination or reduction with the focus on reducing the total lead time, (c) create a new value stream based on the proposals to use as a guide to improve the process. We also want to understand more about the usefulness of value stream maps in these types of software process improvements.

II. BACKGROUND AND RELATED WORK

Toyota revolutionized the automobile industry with their approach of ‘Lean Manufacturing’ in the 1980’s. Womack and Jones, through publications such as [5], [6], enhanced the original concepts of lean manufacturing into the framework of ‘Lean Thinking’ popularized during the 1990’s. The authors distilled the essence of the lean approach into five key principles to show how the lean concept and practices can be extended beyond the automotive industry to any company or organization in any sector. In the early 2000’s, Mary and Tom Poppendieck further transferred these principles and practices from the manufacturing domain to specifically suit the software development environment and called it ‘Lean Software Development’ [4].

The ‘waste’ concept is at the center of the lean philosophy. Waste is defined as *any activity that consumes time, resources, or space but does not add any value to the product as perceived by the customer*. All lean approaches focus on identifying and eliminating waste by looking at the flow of value from request to delivery. So if a customer wants something, what steps does that customer’s request go through until value is delivered to the customer? How fast does that process flow? If the customer’s request has to wait in a queue for approval or if the tested code has to wait for system integration implies that the work is not flowing fast. The work will flow fast only when customer’s request will move rapidly without any interruption or waiting involved. There are seven different types of wastes that may occur in a software engineering process namely partially done work, motion, waiting, extra processes, extra features, task switching, and defects. These are known as the ‘seven wastes of software development’ and they have been transferred from the manufacturing domain to the software engineering domain by Poppendieck et al [4].

To date, most of the VSM work has been focused on analyzing waste in the context of manufacturing operations. Rother and Shook, in their famous work [7], provide a detailed account on the practical implementation of VSM, which is also used in other sectors and service related

industries. For instance, the ‘Lean Advancement Initiative’ research consortium at MIT, has for many years applied lean concepts for improving aerospace product development processes and extensively used VSM for diagnosing and eliminating waste in those processes. Their main work ‘Product Development Value Stream Mapping Manual’ [8], offers extensive academic research for learning. There are some techniques available for optimizing a value stream with the intent of reducing waste in a process such as Value Engineering (VE), Value Analysis (VA), Design Structure Matrix, and so on [9]. However, the technique mentioned for analyzing wastes in software development processes is ‘Value Stream Mapping’ (VSM). To the best of our knowledge, no study in software engineering has been reported so far, which aims at using VSM as a tool to visualize a software product development flow, and not just a SPC flow.

III. VALUE STREAM MAPPING CONCEPTS

VSM is based upon lean principles and is a powerful tool to use for identifying opportunities for significant process improvement. VSM helps to uncover bottlenecks in a process that prevents it from flowing at its optimum. It enables organizations to understand any workflow, taking an end-to-end view of their process capability. Figure 1 shows the notations used for drawing the VSM. These notations have been used by Poppendiecks [4] to demonstrate an example of how value stream mapping are applied to software development processes. A ‘Process Box’ indicates a single step or activity in the process in which the work is flowing. The step is written inside the box. An ‘Arrow’ indicates the direction of flow from one step to another. The ‘Timeline’ indicates the sequence of process steps arranged in a chronological order with different times displayed along a line (drawn from left to right). The different times shown in Figure 1 represent different measurements, which are described below.

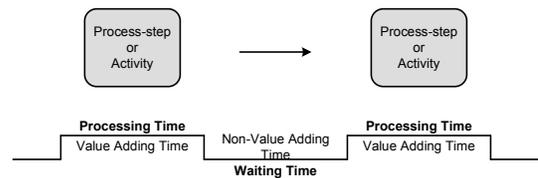


Figure 1. VSM Notations

VSM Measurements: The measurements used in a VSM are: *Lead Time (LT)*: The average time it takes for one request to go through the entire process from start to finish including all queuing/ waiting times between sub-processes a.k.a ‘Throughput Time’ or ‘Turnaround Time’; *Queue Time (QT)*: The average time between sub-processes that the request sits around waiting a.k.a ‘Waiting Time’; *Processing Time (PT)*: The time the request is being worked on by one person or a team a.k.a ‘Processing Time’.

VSM States: In value stream mapping, the ‘Current State Map’ describes the current state of the studied process and is used as baseline for the improvements while the ‘Future State Map’ describes the development process as it can look in the future, if the identified wastes have been avoided.

IV. RESEARCH METHOD

The company studied is Ericsson AB. The market of the company can be characterized as highly dynamic. Furthermore, the customers require highly customized solutions in their telecommunication applications (e.g. specific needs for operators in different countries). A large number of telecom operators is addressed, including many potential customers, which makes the development market-driven. The products are mature and have been on the market for more than 5 years. The products need to have high availability. The products can be considered large-scale, with the overall product having more than 5 million lines of code. The company’s development is globally distributed across three different continents with its main office in Sweden. The organization is ISO 9001:2000 certified. The development site studied employs over 600 people. The company is in the list of the Fortune 500. The development model of the organization is incremental development. Furthermore, agile practices are used [10].

A. Research questions

The research questions are driven by the goals formulated in the introduction. The following research questions are to be answered:

- *RQ1:* What are the most critical wastes in a software product customization value stream and what typically causes these wastes?
- *RQ2:* What can be done to eliminate or reduce the identified wastes in the product customization value stream and what impact would that make on shortening the total lead time?
- *RQ3:* What lessons can be learned to better address the challenges of software product customizations?

B. Data Collection

Below we describe the multiple data sources used in the study. Figure 4 provides a one-to-one mapping of the various steps that were used to collect data and corresponding outcomes that led us to create the value stream maps.

1) *Document Analysis and Discussions:* We looked at the process documents and presentations to discover the main process steps of SPC development and to understand the flow of value. We also talked to experts in order to clarify process and flow understanding and things missing in the documentation. The information was used to construct the current state map of SPC value stream. The main actors involved and different process steps used in the current state map are described below. The main actors are:

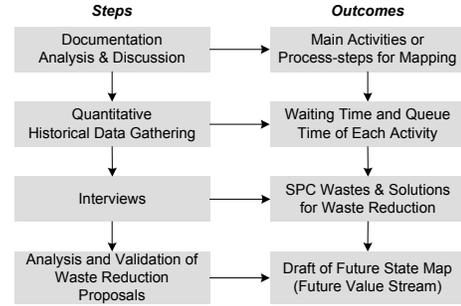


Figure 2. VSM Mapping Process Steps

Customer: The customer or operator who demands rapid delivery of a software customization from the market unit in their own country or region. *Market Unit:* The market unit (MU) is the interface to the customer and communicates customer need or high level requirement to the development unit. From MU side, it is the Solution Architect who communicates with the development unit. *Development Unit:* The development unit (DU) analyzes requests and design and implement customization solutions. From the development side, it is the system analyst who communicates with the MU. The main process steps are: *Solution Proposal:* A document containing detailed requirements and functional specifications. *Decision Board Meeting:* Responsible for taking a strategic decision on accepting or rejecting a customer’s request for a customization. *Business Quotation:* A document containing bidding details, which includes a solution proposal, initial cost estimates, and preliminary delivery-date of design. *Technical Design:* A document containing the technical design. *Implementation:* Refers to coding activity and function test. *System Testing:* Tests a completely integrated system to verify that it meets its requirements. *System Integration Test:* Verifies that a system is integrated to any external or third party systems.

2) *Quantitative Analysis of Historical Data:* The company uses a distributed requirements tracking tool for managing all customization requests. We used this tool to obtain historical data for doing the measurements mentioned in III. In the first step, we made a selection of requests for which we needed the data. The criterion was to select all SPC requests, which arrived in the DU in last one year and were delivered to the customer. In the process, we filtered out those requests, which were (1) rejected by the customer at some point; (2) still processing or not delivered to the customer; and (3) reuse cases. Next, we examined each request individually and recorded the time stamps for the start/end of each activity in our VSM. Once the data set was complete, we calculated the processing time PT for each process step by taking the difference between the start and end dates of the activity. Similarly, the waiting time QT between two consecutive processes was calculated. In

situations where either one of the dates were missing, we either looked for the end date of the preceding activity or the start date of the next immediate activity to determine the missing start/end date. Poppendieck [4] contends that the lead measurements for VSM can be based on guesstimates. That is, as mentioned before it can simply be a paper-and-pencil exercise where one walks around the organization and talks to different people involved in each activity to record average amount of time being spent on each one of them. Therefore, we tried doing the same but during our initial contacts with the people, we found out that there existed a lot of variation in the lead time for a single activity and thus it was difficult for them to give us a good average measure. So we decided to alter our approach. That is, to strengthen the value stream analysis, we looked into multiple data sources to collect actual lead times and then determined the averages to avoid any potential bias.

3) *Interviews: Selection of Interviewees:* We selected six key people in the organization who are involved with the day-to-day tactical and/or the operational level decisions and also have significant understanding of the customization workflow. Hence, we did not consider roles such as developers or testers. The included roles were Solution Architect (Customer Solution Environment), Account Manager (Customer Relations), System Analyst (Requirements Analysis), Tech Coordinator (Team Coordination), Project Manager (Project Management), and Processes Owner (Process Management).

Structure of Interviews: Each interview had three parts, namely: (1) introduction and warm-up, (2) main body of the interview, and (3) closure. The overall duration of the interviews was one hour. In the introduction and warm-up, the interviewer introduced himself and explained the goals of the study. All interviewees were assured about the complete anonymity of their views. For a quick verification of the current state map developed by the researchers, the interviewer asked some questions to make sure the interviewee agreed with the information in the map. It also helped the interviewees in getting themselves familiar with the map before answering the main questions. The questions asked in part 1 were 1) Is the information in the map accurate and complete? If not what is missing or should be changed?, and 2) Are the average duration(s) (lead times and waiting times) calculated for the phase(s) in line with your experience? If not, what do you think is/are more reasonable approximation(s) based on your observation?

In the second part, the map was a basis for questions relating to the main research questions. The questions posed were 1) What are the most critical wastes that can be identified by looking at the current state map?, 2) What is the main cause of occurrence of each waste?, and 3) What measures can be taken to eliminate or reduce the identified waste? How can it be incorporated in a future value stream? In the closure part, we asked interviewees to do a brief

evaluation of VSM as a method.

Transcription of Interviews: The interviews were recorded and later transcribed. In the first step, each waste along with its root cause and the solution proposal to reduce it was extracted from the raw data. This list contained a total of 6 wastes. In the next step, we looked at the individual definitions of software development wastes to find out which one of them were related to which one of the wastes in our list. Since this came out to be a one-to-one mapping, each waste was categorized as unique and thus given the appropriate name.

4) *Analysis of Solution Proposals:* Having identified the wastes and gathered solution proposals by means of the current state map, the proposed changes in the current value stream were outlined in a draft of the future state map. We analyzed all one-to-one cases (waste and its proposal for reduction) and determined the impact of each change on mapping measurements described in Section III.

5) *Static Validation:* Draft of the future state map was presented to upper management in a series of seminars. The findings were well received and created extensive discussion. The description of the present state and all solution proposals were considered valid. The map and discussions it helped create were considered important for getting support for actually making the changes at different levels of the company. However, some risks that we had identified in interviews were also mentioned by management at the seminars; they are discussed in section VI-B. Overall, the positive feedback from seminars regarding the method and its results validated the usefulness of VSM as an effective process improvement tool.

V. RESULTS

This section presents the findings of the case study. First, the SPC current state map is presented in Section V-A. Next, we describe the SPC wastes in detail in Section V-B and their solution proposals in Section V-C. The analysis leading to a future state map is presented in Section V-D.

A. Current State Map

Due to confidentiality reasons we are not allowed to show the actual times in the current state map, instead they are shown as average percentages of the average total lead time. For the same reason we cannot reveal the total lead time. Figure 3 shows that the total PT or VA time amounts to 62% of the LT and the total QT or NVA time amounts to 38% of the LT. Hence, the proportion of NVA to VA tells us that more than one third of the resources are spent waiting and/or doing activities that are not adding any value for the customer. The percentage measurements in the figures are referred to as days in the text.

The back-and-forth arrows in the map represent asynchronous communication between two process steps for feedback and/or clarification. The waiting times for these are

Table I
TABLE OF WASTES

Waste	ID	Description	Sol. ID	Proposal to Eliminate Waste
Waiting	W1	Waiting Time in Design	S-W1	Delay Delivery Commitments
Waiting	W2	Waiting Time for Customer Sign-offs	S-W2	No Solution Proposed
Waiting	W3	Waiting Time for System Integration Testing	S-W3	Parallel Testing with Early Beta Release
Extra Process	W4	Extra Process for Solution Proposal Building	S-W4	Provide Quick High Level Estimate
Extra Process	W5	Extra process for Design	S-W5	Merge Solution and Design Process Steps
Motion	W6	Motion of Requirements	S-W6	Close Collaboration

not mentioned, since they were not available to us. However, we were told that such communication loop-backs cause major delays and affect lead time.

The Table I provides an overview of the wastes identified in the VSM and the interviews. There are three wastes of waiting (W1-W3), two of extra processes (W4 and W5), and one of motion (W6). The corresponding solutions to remove the wastes are also listed in the table (S-W1 to S-W6).

B. SPC Wastes

The current state map of the SPC process revealed only the Waste of Waiting. However, two additional wastes were identified during the interviews, which were 1) Waste of Motion and 2) Waste of Extra Processes. This was possible because the map acted as a trigger for interviewees to think about and point out few other sources of wastes in the value stream. Also, the reasons for different waste occurrences were not apparent from the map and pointed out during the interviews. Below we describe each waste in detail.

W1: Waiting Time in Design: Once the customer agrees to buy the technical design, the request has to wait an average of 15.2 days to enter into the design phase (see W1 in Figure 3). This time lag arises due to the current way of planning SPC deliveries and scheduling customer commitments. Since the customers want to know in advance, the delivery date of the technical design along with the solution proposal, the time is allocated quite early in the development plan. That is, even before the customer signs a contract and commits to a SPC. The arrow before *Decision Board Meeting* in Figure 3 indicates that the date is agreed upon in a meeting prior to sending the business quotation to the MU. Furthermore, the SPC requests are realized in order of their arrival according to the first-come, first-serve (FCFS) basis. So at any given time, a lot of slots in the development plan are pre-occupied by many requests that are not orders. In consequence, ordered SPCs that have a commitment from the customer have to wait for requested SPCs without commitment to disappear from the slots first. In some cases the slot can be months ahead.

W2: Waiting Time for Customer Sign-offs: The waste of waiting occurs when customers take too long time to decide

whether they want to buy the technical design after receiving a business quotation. This means that the development has to wait for an average of 20.2 days to get customer buy-in (see W2 in Figure 3). According to the SPC process, there is a quotation expiry period within which the order must arrive. When a customer does not meet this deadline, the request should be canceled and the time slot reserved for doing the technical design should be removed from the development plan. However, the deadlines does not seem to be strictly followed since the average QT represented by W2 in Figure 3 is greater than the quotation expiry time.

W3: Waiting Time for System Integration Testing: The waste of waiting occurs if the MU finds a fault in system integration testing and waits for a long time to get the correction (see W3 in Figure 3). The map shows that the SPC takes 10 days PT in system testing before the hand-over to MU takes place for system integration testing. Upon finding any problems in system integration testing, the MU reports a fault and unnecessarily waits to get the correction because a SPC is not handed over to them without bypassing system testing, even in the case of a small change. In order to get a correction earlier than the next planned date (completion of 10 days testing cycle), the MU has to raise the priority of the identified fault to the status of emergency correction. Even if they get the correction on a priority basis, few weeks delay could occur.

W4: Extra Process for Solution Proposal Building: This waste of extra process occurs when effort is put in writing the solution proposal and subsequently preparing the business quotation but customer refuses to accept the proposal or give buy-in for it. This is the case for almost 50% of the completed solution proposals (see W4 in Figure 3). This becomes evident, if we look at the total lead time between the moment SPC opportunity arrives to the moment a business quotation is sent to the MU, which is quite long (LT = 25 days). During this period, much needless effort is expended and resources are engaged, which could be saved otherwise.

W4 also produces the waste of waiting. That is, it implies that the process is not efficient enough to acknowledge new SPC requests and provide MU with a business quotation

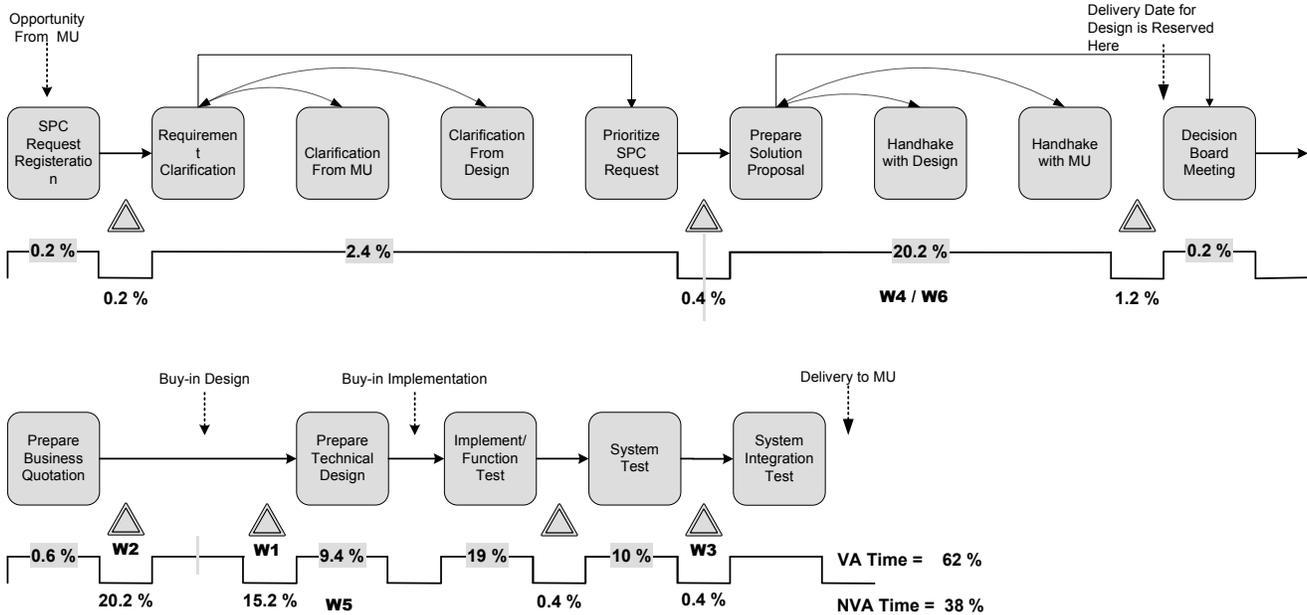


Figure 3. SPC Current State Map

early on. According to the interviewees, customers initiating a SPC should receive an immediate response from the company. They need a proposal within a short time period because they want to conduct a business feasibility study to assess the cost and benefits before making a final decision. Hence, delaying a response to customer has one big disadvantage for the company. If the customer does not see a successful business case (for example, due to the SPC's high cost or large change impact) after months and rejects the proposal, then all the effort in creating the proposal is wasted. Thus, the company is putting a lot of resources (time/effort) in creating the proposal without being sure whether the customer will commit or reject the proposal.

W5: Extra Process for Design: The waste of extra process occurs when documentation is split into two different deliverables (i.e., solution proposal and technical design), performed at two different points in time during the process flow. Waiting aside, this causes extra resources and increases the total lead time of SPC development. The map in Figure 3 illustrates that after customer buy-in, another 9.4 days are consumed in writing the technical design. According to the interviewees, on the one hand, splitting of documentation effort for big/complex SPC requests has a value because it may take a very long time to both investigate and design a solution in one flow. Meanwhile, the need might have changed and the customer may no longer need that SPC. On the other hand, when the same is done for small/simple SPC requests, the splitting becomes a source of redundancy. This is because sometimes a solution proposal has quite sufficient details that it could be used to implement a SPC directly without requiring a separate technical design document.

W6: Motion of Requirements: The waste of motion occurs due to communication between the development unit and the market unit for feedback and/or clarification during the solution building phase. This is indicated in Figure 3 by two arcs that go between solution proposal and handshake with design/ MU. What we have found out is that this process repeats itself quite many times during this phase, implying a lot of back and forth movement of requirements. This communication also occurs in order to make sure all stakeholders are approaching the same solution, since no one has the same view on the requirements. The analysts look at the requirements from the end-to-end systems perspective, design has a technical orientation, and the MU knows the customer and its environment. The communication mostly happens on emails, tracking tool or via conference calls. These types of communication creates a disconnected environment and also give the feeling of not working as one team. Furthermore, few hours of difference in time zones between the two sides cost at least half a day in waiting, if not more.

C. Solutions to Reduce SPC Wastes

In this section, we present the solutions to eliminate or reduce SPC wastes.

S-W1: Delay Delivery Commitment: The waste of waiting in design (i.e., 20.2 days QT) can be reduced through delaying the commitment and changing the current way of planning. That is, no commitment should be made to deliver the technical design until a purchase order is received. In other words, fixed upfront planning should be avoided to reduce the inventory of requests and thus decrease the QT

for incoming orders. This can be achieved when planning the deliveries based on actual orders rather than requests.

S-W2: Customer Sign-offs For the waiting time for customer sign-offs, no concrete solution has been provided by the interviewees as it is hard to force the customers into a response on signing a contract.

S-W3: Parallel Testing with Early Beta Release: The waste of waiting for system integration testing (i.e., 10 days PT in system testing) can be reduced by providing MU with an early beta version of the SPC so that the system integration testing can be started in parallel to system testing. This will enable MU to report any faults earlier before the system testing cycle finishes, which in turn will allow the DU to immediately address the problem. If needed (in case of some serious issue), the DU can even choose to delay the delivery for few days to include the correction, which would still be much faster than repeating the whole process of system testing again.

S-W4: Provide Quick High Level Estimate: The waste of extra process for solution proposal building can be reduced by providing customer with a quick response in the form of a High Level Estimate (HLE) as an interim to solution proposal. The HLE would not only give customers the possibility to do their business feasibility early on but also provide them with the ability to decide whether they want to pursue the opportunity further or not. If the customer sees a problem with the projected effort/ cost and wants to decline the HLE, the process can be stopped before solution proposal starts. Consequently, the risk that the effort in doing the solution proposal can go waste would be minimized. Moreover, the customers would not have to wait for months to get a response as in the case of current situation. It was suggested that HLE would have a very short turn around time. It would provide only an indication of where a SPC can fall in terms of cost and effort and what analysis would be needed to investigate the solution.

S-W5: Merge Solution and Design Process Steps: The waste of extra process for design can be eliminated by combining the documentation of solution proposal with the documentation of design to produce only one technical document that possesses both the requirements and the design of the SPC solution. However, this would be done only for small to medium size SPC requests and/or those requests that are relatively simple to investigate. This also means that it would require a mechanism to quickly find out about the size/ complexity of the SPC request as soon as it arrives so that the decision to combine or not to combine the two process steps could be taken.

S-W6: Close Collaboration: The interviewees stressed the importance of close collaboration between the DU and the MU. They suggested that the solution architects from the MU should spend some time in the DU and work together with the analysts, designers and the testers. They may work together for few days, develop common understand-

ing, discuss and debate issues and potential solutions, and then create requirements. This could be helpful in avoiding excessive communication loop-backs, and thus could help to reduce the LT of writing a Solution Proposal. According to the interviewees, the Solution Architects may not need to come for every SPC but for a request that is not straight forward or is complex i.e. the decision may depend on a case to case basis.

D. Analysis & Future State Map

The future state map is created by combining the waste reduction solution proposals presented in Section V-C. The goal of combining these proposals is to construct one future state map so that it could be used as an input for discussion with the relevant stakeholders in the company. In the following, we briefly explain how each of these proposals aided us in drafting a future state map as shown in Figure 4. The new process steps and measurements (LT/PT/QT) that are likely to be impacted, have been highlighted green to illustrate potential changes. The reason for not providing a new improved (LT/PT/QT) is discussed in Section VI.

Impact S-W1: We can predict that the waiting time in design (QT = 15.2 days) can be reduced when no commitment is made prior to getting the actual order from the customer. To reflect this change in the future state map in Figure 4, the ‘Commit Design Delivery’ decision point is moved next to ‘Buy-In Design’. In the current state, the inventory of development slots may get full to its capacity because it is filled with both requests and orders. In the future state, the development slots would be filled with only orders and no requests, which in turn would reduce the length of the queue and thus the waiting time for new orders.

Impact S-W2: As mentioned earlier, no solution has been proposed by the interviewees for reducing the waiting time for customer sign-offs (QT = 20.2 days). However, we have mentioned few potential solutions in Section VI and that is why we have marked the waiting time as ‘impacted’ in Figure 4.

Impact S-W3: Assuming that if MU finds a fault during system integration testing while the system testing is still going on, we can predict that the correction delivery time can be reduced. This is reflected in Figure 4 by adding an alternate arrow (originating from the end of Implement/Function Test process-step) to show early availability of SPC for system integration testing. Since it was an additional waste identified during the interviews for which the waiting time was not available in the current state, therefore, the ‘impact’ is not visible in the future state map.

Impact S-W4: A new process step called ‘HLE’ is introduced in Figure 4 to create high level estimates for the new incoming requests. Since the customers would also need some time to make a decision based on this estimate, therefore, a new decision point ‘Buy-In From Customer’ has been added afterwards. This will introduce two ‘New’

waiting times before and after as shown in Figure 4. We can predict that if the customer makes a buying decision at this point, the waste of extra process for solution proposal building (PT = 20.2 days) can be significantly reduced, since we already know that almost half of the SPC requests are rejected later.

Impact S-W5: The proposal for removing the extra process for design is dependent on S-W4. That is, based on a HLE, the effort/size/complexity of the SPC can be judged and hence it can be decided whether the solution proposal and technical design could be combined together or not. If they cannot be combined, the request can follow the old path illustrated by ‘Go as Before’ in Figure 4. Otherwise, the request can take the alternate path illustrated by ‘Go Fast Track’ in Figure 4.

Impact S-W6: A new process step called ‘Solution Exploration and Discussion’ is introduced in Figure 4. We can expect that the communication and coordination overhead can be reduced through temporarily co-locating the team to discuss the solution before proposal writing starts. SW6 is independent of S-Ws4.

VI. DISCUSSION

A. Lessons Learned

The lessons learned can be generalized to other companies also that operate in a dynamic market with globally distributed customers, and develop large-scale software products. In the following we present the implications/lessons learned from this study.

Commit as late as possible: When committing too early and reserving development resources for planned SPCs bares the risk of creating long waiting times for ordered SPCs, as discussed before. In consequence, a lesson for companies is to commit as late as possible. This is essential from a risk perspective as the company is making high investments without any insurance of commitments. The risk is significant considering the dynamics of the market Petersen et al. [10] investigated the development of regular product features in a similar context and found that changes in the market needs are a significant source for waste in the requirements phase. That is, in regular development 26 % of the regular requirements had to be discarded. These market-dynamics motivate even more why shortening the lead times is important. However, we would like to emphasize that this should not be done on the cost of quality, as quality is an important factor in a product’s success. A recently conducted large survey, consisting of respondents who practice agile software development and had experience practicing plan-driven software development in the past, indicates that ‘customer commitment’ is one of the nine important success factors that have a statistically significant relationship with projects’ success [11]. Also, the notion that production should not be based on forecast and

thus commitment should be delayed until demand is present constitutes one of the principles of lean thinking [12].

Concurrent Development: When doing an analysis of value stream maps it is important to look for activities that can potentially be done in parallel. In this study we identified the two testing activities (system test and system integration test) that should be run in parallel. This allows to fix faults soon after their introduction. The sooner a fault is found in time after its introduction, the easier and less costly it is to fix [4]. In this particular case another reason was that the process prescribed that one has to re-execute the system test if a small defect is found in the system integration test.

Responsiveness of the Customer: As could be seen in the analysis no solution was proposed to reduce the time the customer needs to sign the contract. The reason is that it is risky for a company trying to force their customers in signing the contracts as this would possibly offend them. Instead, we believe that it is important to generate incentives for the customers why they should sign the contracts as early as possible. For example, customers that keep within the requested deadline receive higher priorities in the development, or get an activation for nice-to-have features in their products.

Co-located development: It is a challenge to achieve a common understanding of the requirements of a software product. This is particularly challenging when people working on the same task are not sitting together [10]. The same holds true for this case study. Even though it is not possible to physically co-locate the teams permanently we expect an improvement in understanding the requirements when the teams get together for a short period of time.

B. Risks

This section discusses some of the potential risks involved in implementing the solution proposals that constitute the future state map of SPC.

Parallel testing: The parallelization of testing activities (S-W3) poses one important challenge. According to the future state map, if the SPC goes directly into system integration testing without going through in-house system testing, there would be a risk of code-break since SPC has been only function tested. This means the approach requires some level of ‘built-in-quality’ in the code and for that quality standards should be established to avoid trivial problems with the implementation.

Bottleneck due to lack of experts: The High Level Estimates (HLE) (S-W4) are beneficial as we found in the future state map. However, this might create a new bottleneck as very experienced people having in-depth knowledge of the system will have to do the estimate when using expert estimation. It is not always necessary that a company who uses HLE to quickly respond to their customer requests also have enough experienced people at hand to do the job. From the implementation perspective the single document containing the solution proposal and design (S-W5) in a

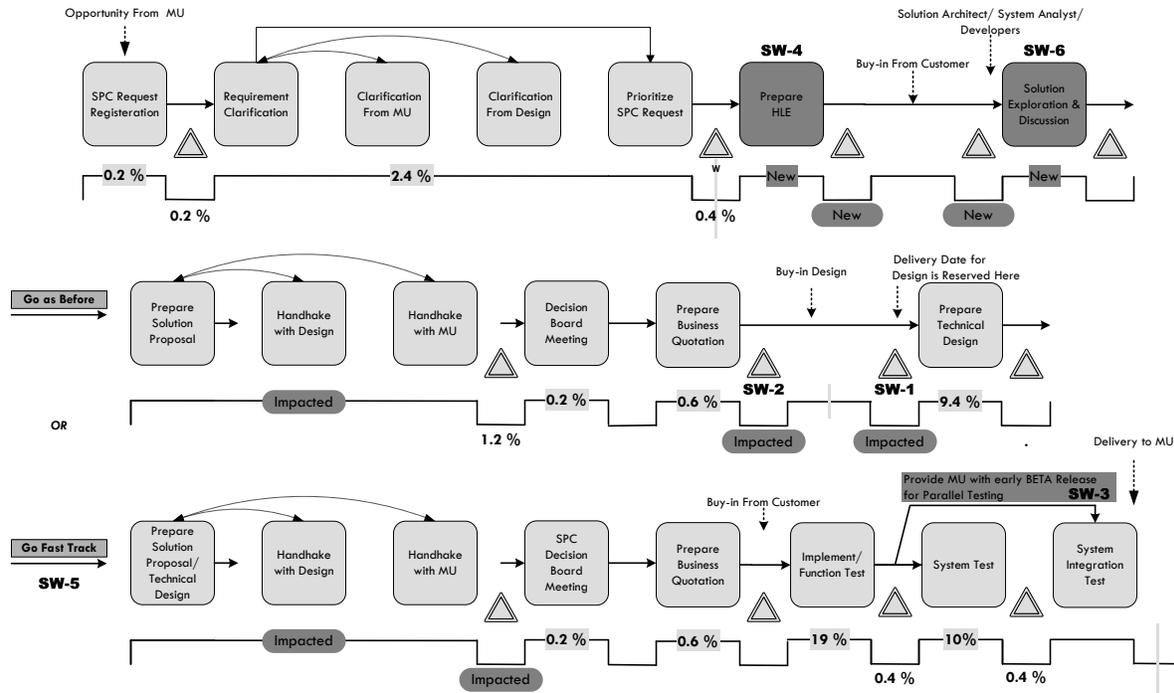


Figure 4. Draft of Future State Map

new process may not going to be as detailed as with the long process in the current state value stream. That is, the lack of detailed design elements in an input document for implementation can make less experienced developers run into trouble. In order to avoid this problem, the development teams should have experienced leaders who could understand and translate a high-level design into working code. Also, it is important to co-locate people (as discussed before) as this has the additional benefit of sharing knowledge to avoid this problem [13], [10].

Value Stream Analysis: The VSM enables one to easily spot the waste of waiting without having a need to show the map to the stakeholders. However, when people in the organization are asked to analyze the map, some addition wastes could be found. For instance, in our study, the interviews resulted in two additional wastes (extra processes, motion). The remaining wastes (if existed) that are not identified in a value stream analysis are considered unimportant and hence should not be of major relevance. Furthermore, when using VSM as a tool in a different context, it can be expected that different wastes would be found.

VII. CONCLUSION

In this study we used value stream maps to identify wastes in a software product customization process, with the goal of reducing lead time. Three types of waste were identified: waiting, extra processes, and motion wastes. A detailed analysis showed that 38% of the total time to develop a software product customization was not adding

any value. The major waiting times were related to the design phase, to the time the customer requires to sign the contract after receiving an offer, and to the time to correct faults identified in system integration testing after introducing the customizations. The value stream map was validated in interviews with key stakeholders and then used to discuss and create solution proposals. Based on the solution proposals a future state map was created. Since creative processes like software development are less clearly defined and harder to predict than manufacturing processes we cannot yet quantify the improvements the solutions will lead to. However, the interviewees were in agreement that the maps and the VSM methodology are useful since they are easy to understand and help in visualizing problems and possible improvements.

The main lessons learned about the customization process at the company are to commit as late as possible to a customization request, to look for opportunities to parallelize development activities, to provide incentives for the customer to respond quickly to an offer, and to ensure that early development activities that are currently separate should, at least temporarily, be co-located to clarify requirements. These lessons learned are important for other companies in similar situation to consider, in particular for their product customization processes. Our study shows that value stream maps can be an important tool in analyzing and reducing lead time in software product customization. Its benefits should also carry over to more general software engineering

improvement activities and useful to both researchers and practitioners.

VIII. FUTURE WORK

Typically value stream mapping does not have a means to represent loop-backs that may exist between two process steps. This may be needed to perform detailed quantitative analysis, where required, as the processing times or queue times for the same activity could be different for each iteration of a loop. This is particularly important with the increased use of incremental and agile development that can be characterized as highly iterative. It would be of interest to see how the value stream mapping technique could be extended in cases where iterative flows exist. Also, due to the lack of empirical evaluation of value stream maps in the software engineering context further investigations of the usefulness of the technique are needed. The studies should focus on the application in different contexts.

ACKNOWLEDGMENT

The authors would like to thank all the interviewees for their participation in the study. Further thanks go to PerOlof Bengtsson and Joel Cassel from Ericsson.

REFERENCES

- [1] M. A. Schilling, "Technological lockout: an integrative model of the economic and strategic factors driving technology success and failure," *Academy of Management Review*, vol. 23, no. 2, pp. 267–284, 1998.
- [2] G. L. Urban, T. Carter, S. Gaskin, and Z. Mucha, "Market share rewards to pioneering brands: an empirical analysis and strategic implications," *Management Science*, vol. 32, no. 6, pp. 645–659, 1986.
- [3] G. S. Jr., "Time - the next source of competitive advantage," *Harvard Business Review*, vol. 66, no. 4, 1988.
- [4] M. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit (The Agile Software Development Series)*. Addison-Wesley Professional, 2003.
- [5] J. P. Womack, D. T. Jones, and D. Roos, *The Machine That Changed the World : The Story of Lean Production*. Harper Perennial, 1991.
- [6] J. P. Womack and D. T. Jones, *Lean thinking : banish waste and create wealth in your corporation*. London: Free Press Business, 2003.
- [7] M. Rother and J. Shook, *Learning to See: Value Stream Mapping to Add Value and Eliminate Muda*. Brookline, MA: The Lean Enterprise Institute, 1999.
- [8] *Product Development Value Stream Mapping (PDVSM) Manual*, Massachusetts Institute of Technology - Lean Advancement Initiative (LAI), 2005.
- [9] Y. Cai and J. You, "Research on value stream analysis and optimization methods," in *4th International Conference on Wireless Communications, Networking and Mobile Computing*, oct 2008.
- [10] K. Petersen and C. Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," *Journal of Systems and Software*, in print, 2009.
- [11] S. C. Misra, V. Kumar, and U. Kumar, "Identifying some important success factors in adopting agile software development practices," *Journal of Systems and Software*, June 2009.
- [12] M. Poppendieck, "Principles of lean thinking," <http://www.poppendieck.com>, accessed Aug 25, 2009.
- [13] B. W. Boehm and R. Turner, *Balancing agility and discipline: A guide for the perplexed*. Boston: Addison-Wesley, 2003.