FLOW-assisted value stream mapping in the early phases of large-scale software development

Nauman Bin Ali*, Kai Petersen*, Kurt Schneider +

*Blekinge Institute of Technology, Karlskrona, Sweden
Email: {nauman.ali, kai.petersen}@bth.se

+Software Engineering Group, Leibniz Universitat Hannover, Germany
Email: kurt.schneider@inf.uni-hannover.de

Abstract

Value Stream Mapping (VSM) has been successfully applied in the context of software process improvement. However, its current adaptations from *Lean* manufacturing focus mostly on the flow of artifacts and have taken no account of the essential information flows in software development. A solution specifically targeted towards information flow elicitation and modeling is FLOW. This paper aims to propose and evaluate the combination of VSM and FLOW to identify and alleviate information and communication related challenges in large-scale software development. Using case study research, FLOW-assisted VSM was used for a large product at Ericsson AB, Sweden. Both the process and the outcome of FLOW-assisted VSM have been evaluated from the practitioners' perspective. It was noted that FLOW helped to systematically identify challenges and improvements related to information flow. Practitioners responded favorably to the use of VSM and FLOW, acknowledged the realistic nature and impact on the improvement on software quality, and found the overview of the entire process using the FLOW notation very useful. The combination of FLOW and VSM presented in this study was successful in systematically uncovering issues and characterizing their solutions, indicating their practical usefulness for waste removal with a focus on information flow related issues.

Keywords: Case study research, Value stream mapping, Lean software development, information flow modeling, Process improvement

1. Introduction

Value stream mapping (VSM) is a *Lean* practice that maps the current state map (CSM), identifies value-adding and non-value adding activities and steps, and helps to create a shared action plan for an improved future state for the process i.e. a future state map (FSM) [1] [2] [3] [4]. VSM looks at both the material and information flow [2]. In software development, an equivalent analysis of material flow will look at the flow of *"work items"* e.g. a requirement, use case or a user story through the process (referred to as artifact flow). Contrary to simply the scheduling information that is captured in terms of "information flow" for production processes, for software development it will be pertinent to capture documented/verbal, formal and informal communication. Furthermore, information, knowledge and competence required to carry out the value-adding activities in the software development process have to be identified and analyzed.

The current adaptations of VSM have had a high focus on artifact flow, identifying waiting and productive times [1] [3] [5]. No particular challenge occurs here if the communication structure is relatively simple. However, in the more complex setting of the studied organization, it became equally important to focus on the information flow and explicate it since most of the challenges identified with typical VSM analysis were related to information needs and documentation.

The aim is to achieve an information flow that leverages the *Lean* principle of "pull' such that one process produces only what another process needs [2]. The existing guidelines and notation for VSM [2] [3] do not allow capturing the information flow. As a consequence, we cannot identify value-adding and non-value adding activities required to streamline the process of value creation. There is a need for a systematic, lightweight approach that could supplement VSM, without a lot of overhead. Furthermore, the notation used should be simple and intuitive without requiring additional training of practitioners to understand and analyze the information flows visualized using it.

Information flow modeling (FLOW) has been proposed as a systematic method to analyze and improve communication in software development processes by considering all types of experience and information flows in the organization [6].

We propose to combine value stream analysis with FLOW. As both share a focus on capturing and improving information flows, their combination can yield potentially useful results. FLOW applies a systematic approach and a structured yet simple graphical notation to identify and capture the flow of information between people, documents, and activities. It covers and combines formal, informal, documented and verbal information flows in a complex communication structure; hence, it can compensate the inability of existing VSM notation to capture this aspect of information flow in software product development.

To evaluate the usefulness and scalability of using FLOW to assist VSM, we conducted case study research in a large-scale product development at Ericsson AB. We conducted six value stream workshops, with two researchers (first two authors) acting as facilitators, and the third author (who is the inventor of FLOW) as an external observer.

The remainder of the paper is structured as follows: Section 2 presents related work. Section 3 describes the research method. Section 4 presents how FLOW can be used to assist in a typical value stream mapping activity. Section 5 reports the application of FLOW-assisted VSM in the case company. Section 6 presents the results of the study and a follow-up six months after its conclusion. Section 7 reflects on the experience of applying FLOW-assisted VSM and Section 8 concludes the paper.

2. Related work

The related work for this study has three main themes: first the broader theme of software process improvement using agile and *Lean* methods, second is the use of VSM and third is the use of FLOW for mapping and improving software development process.

2.1. Process improvement in Lean software development

The primary focus of lean process improvement is the identification of bottlenecks hindering the efficient flow of the process. Therefore, different measures and visualizations have been proposed.

Staron and Wilhelm [7] investigated the method and measures employed by Ericsson to identify bottlenecks in the development process. It is stated that the company on the basis of the Lean concepts of VSM and queues developed these methods and measures. With a similar intention as Staron and Wilhelm, Petersen et al. [8] have proposed and evaluated visualizations to identify and resolve bottlenecks in the development process and to follow-up on the degree of success of the resolutions. During this study, the time-line analysis (delineating the productive vs. waiting times) and cumulative flow diagrams were used to analyze the process. Cumulative flow diagrams can be used to guide the identification of root-causes to explain the reasons for the observed bottlenecks (cf. [9] as an example).

The work by Staron and Willhelm [7] acknowledges the underlying contribution of VSM but that was not the object of their study. While the earlier work on identification of bottlenecks by Petersen et al. [9] and [8] uses the typical visualizations used in a value stream analysis, but it does not use the VSM as a framework to guide the improvement activity in the organization.

2.2. Value stream mapping

Rother and Shook [2] presented the guidelines to conduct VSM for manufacturing processes where both material and information flow are mapped. Realizing the differences in product development and manufacturing McManus [3] adapted VSM for this new context. McManus [3] developed a manual for applying VSM for product development, which is the foundation of this work along with its extension by Khurum et al. [1].

Two applications of VSM were reported in the context of software product development [1], [5]. Mujtaba et al. [5] have reported a VSM for product customization process where the goal was to reduce the lead-time. Khurum et al. [1] identified value aspects that are not traditionally considered, but have to be evaluated on a case to case basis, based on what each organization conducting the VSM consider important from their customers' perspective. In both cases, VSM was done on mature products that have been on the market for a number of years. In this study, we apply VSM in the context of a new large-scale software product development distributed across sites in multiple countries.

Poppendieck and Poppendieck [4] also provide a brief three-step process to do a VSM. However, the overwhelming focus of analysis in these guidelines is on developing a time-line of progress through a process and making a distinction between waiting and productive time (c.f. [1] [4] [5]). The aim of *Lean* is to ensure that a process should make only "what" is required and "when" it is required by the next process [2]. The time-line analysis focuses primarily on "when" in the aim above and seems to overlook the "what" aspect i.e. what information needs exist that should be fulfilled.

In this study, we complement the time line based analysis of work items, which is typical in VSM, with the use of FLOW methodology to identify and analyze information flows in the case of large-scale software product development.

2.3. Information flow modeling

Many established software process models focus on documents only [10]. Verbal or informal communication, and the use of experience in sophisticated activities are, however, often neglected. In industrial reality, however, many requirements, decisions, and rationales are communicated informally, via phone, meetings, or personal email. The FLOW method offers a graphical notation (Figure 3) and a modeling approach that covers and combines documented (solid) and non-documented, verbal, or informal (fluid) flow of information. Traditional plan-driven software development tends towards solid information propagation, whereas agile or flexible approaches rely on direct communication [11], of fluid information.

Both solid and fluid information flows have complementary advantages and weaknesses. Solid information is typically stored in documents, files, recordings, or other kinds of permanent media. Creating and retrieving information takes longer and requires more effort, but does not rely on the creator, and can be performed at a later point in time and by many people. Fluid information flows faster and often more easily: phone calls, chats, or short messages do not get formally documented, but convey information as well. Fluid information is typically stored in a person's mind. Therefore, a face symbol is used to visualize it.

Many companies need to mix and adjust elements from both ends of the spectrum and try to find an overall optimum. FLOW was created to capture, visualize, and improve situations consisting of a complex network of fluid and solid information flows. Experience is considered an important special case of knowledge that must be present when difficult activities occur and important information is routed through the organization [12].

At the beginning of a FLOW analysis, the current communication channels and network of information flows must be identified. The FLOW method [12] [13] offers a technique based on interviews. This technique has been applied to medium-sized groups in several companies. Rather complex networks of solid and fluid information and experience flows were identified, modeled, and discussed with domain experts. In one case, a repository was mined as an indicator of document-based (solid) information flows to complement interview results [14]. The FLOW elicitation technique has not yet been applied to elicit and model complex information flows in very large organizations. Building the model is the first step towards improving the overall information flow. FLOW interviews were optimized for individuals. An important research question was, therefore, whether the interview technique could be transferred directly to a larger group of people (workshop), and how it needed to be adapted.

Several researchers have investigated dependencies and dynamics of software projects including the impact of information flowing through projects: Pikkareinen et al. [15], for example, investigate communication in agile projects and their impact on building trust. The focus of their work is on the impact of agile communication patterns on project success.

Winkler [16] uses the term information flow, too. He focuses on dependencies between artifacts. An artifact is a part of a document. His main interest is traceability of requirements. Winkler considers only document-based requirements and information flows. He uses different ad-hoc illustrations to discuss flow of requirements. Berenbach and Borotto [17] present requirements project metrics. All metrics are based on solid information only. However, information flow within their Formal Requirements Development Process could be modeled in FLOW. Some of their metrics could be extended to refer to fluid information, too.

Data Flow Diagrams (DFD) were designed to model data flow within computer systems [18], but they can also be used to model data flow in a software process. The main drawbacks of using a DFD to model information flows for our purposes are a lack of distinction between solid and fluid information, lack of symbols to capture fluid information and a lack of intuitive symbols to represent human stakeholders.

UML offers a wide range of diagram types. UML Activity Diagrams were designed to model processes using control flows, which is not the intention when capturing information flow. The drawbacks of using Activity Diagrams to model information flows include: no distinction between solid and fluid information, no means of explicitly modeling experience, no visual clues like faces, documents, many details and labels are irrelevant for information flow, and activities are forced in a sequence or order. Those who want to use UML to implement FLOW can use UMLs profiling mechanism to overcome these limitations.

Requirements Centered Social Networks (RCSN) were specifically designed for requirements engineering [19]. RCSNs are generated automatically and therefore only captures a few types of informal communication, such as e-mails and chat protocols. The following are its other main drawbacks if used for information flow modeling: 1) They are designed to visualize the network for a single requirement so they can only represent a single concrete case and not recurring flows. 2) It lacks means to express experience. 3) Too many different node shapes are confusing.

In principle, information flow can be seen as an aspect of a process. Process models such as BPMN or BPEL or workflow models such as Little-JIL [20] would, therefore, be an alternative to FLOW models. However, the scope of FLOW models is smaller, and the focus on fluid and solid information is more specific than in those models that are more general. In addition, the symbols of a FLOW model were carefully selected to visualize human and document-based stores of information. Process models tend to emphasize sequence and coordination of control flow more than the flow of information. The latter was considered more important for this work.

In Table 1 a comparison of several competing notations with FLOW is provided. It shows that certain model mechanisms from such notations can be used to address information flow aspects. However, this will require more or less severe violations of modeling rules or redefinition of symbols to capture information flow syntactically. Moreover, experts familiar with the original notation may find it difficult to work with the new semantic meaning of such an adapted visualization. Due to these reasons use of the FLOW notation is recommended whenever flow of requirements and information is the main focus [21].

Just as we argued that FLOW supports information flow analysis in the VSM, FLOW can benefit from VSM's systematic way of reflecting on wastes and identifying improvements. Besides the systematic approach, the time-line based analysis can help quantify the implications of challenges in information flow in software development. Hence, both VSM and FLOW have synergies and may be even more successful in combination.

3. Research methodology

This study aimed to use and assess FLOW-assisted VSM for the improvement of large-scale software product development process. In particular, to assess the ability to capture complex information flows and identify improvement opportunities in the development process. The scope of the process investigated in this study was restricted to the specification and design phases.

RQ: *How useful is the combination of FLOW and value stream mapping in large-scale software development?* The following sub-research questions were posed to answer it through a case study at Ericsson AB, Sweden, where VSM supported with FLOW was applied and evaluated:

- *RQ1.1: Can the interview-based technique for FLOW be transferred directly to a larger group of people (in a workshop setting), if not, how can it be adapted?*
- *RQ1.2: Which wastes and improvements could be identified with the combination of FLOW and VSM?*
- *RQ1.3: How do practitioners perceive the process and outcomes for the VSM supported by FLOW?*

3.1. Context

The case company is Ericsson AB, Sweden, which is an ISO 9001:2000 certified telecommunication vendor. The product that was studied is still under development and has not had a release to the customer yet.

At the time of this study, the product under investigation was a solution comprising 12 subsystems, the number has since then grown even further. The development unit has over 2500 employees in 10 different centers of excellence, which are located in Germany, Sweden, India, China, Canada, and the US. Teams use agile principles and practices in development, e.g. writing user stories, working in fixed length sprints (time boxing), and cross functional teams, etc.

3.1.1. Product development process overview

The overall process is conceptualized in Figure 1. The process starts with the identification of an opportunity for the organization to exploit e.g. a new potential market, increased customer retention if the product is compliant to certain standards. It is formulated as a Business Opportunity (BOP). Based on the business case, a decision to persist further with the development of a BOP is taken.

High-level analysis of the accepted BOPs with the consideration for the product anatomy, technical requirements and the business context is performed. This results in more detailed specifications in the form of one or more business use cases (BUCs).

Table 1: Overview of FLOW and other competing notations for denoting information flows (based on Schneider et al. [21]).

		FLOW	DFD	UML Activity Diagram	Little-JIL	RCSN
Diagram	Main purpose	Process improvement by considering solid and fluid flows alike	System design	Process design	Process programming	Req. Engineering (RE) awareness
	Main focus	Information, in particular requirements	Data	Activities	Steps	Social network
Concepts	Information store	Person (fluid), Document (solid)	Data store, External	Data store stereotype	Parameters, Agents	Persons
	Information flow	Dashed arrows (fluid) Solid arrows (solid)	Data flow arrow	Data/object flow edge	Parameters with control flow	Communication flow
	Distinction of solid and fluid	Different symbols (see Figure 3)	No	Through stereotypes	No	Style/color of arrow
	Explicit experience	Explicit edge color	No	Association stereotypes	Parameter Type	No
Challenges	When used for information flow modeling	No	Stakeholder as process, data store, or external? Labeling rules violated.	Requires stereotyping for symbols, extended meaning etc.	Fine-grained symbols lead to a very detailed model, as it was a process programming language	Notation is not fully defined, many symbols.

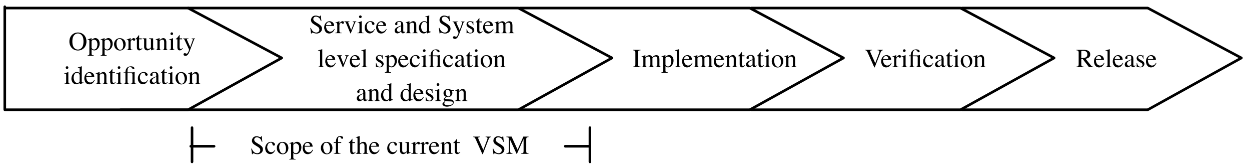


Figure 1: Overview of the product development process in the case company.

BUCs are the work-item used to plan, control and manage development process in the company. A detailed analysis of BUCs is performed and the systems that are required to implement the BUC are identified. A further detailed analysis by the system teams involved in the implementation of a BUC produces more detailed system level artifacts called business sub-use case (BSUCs). At this level, the system teams take on the responsibility of implementing the BSUCs for their systems.

3.2. Scope of the process investigated

The complexity and large scale of the product development entailed that it was impossible to cover the entire process in sufficient detail to analyze it for the challenges and improvements (with over 12 subsystems and also the number of geographical sites involved) in six workshops.

Thus, it was decided to limit the current VSM activity to the *"design and specification phase at both BUC and BSUC study level"* of the overall product development process. Furthermore, this phase is of critical importance as the allocation of work and coordination to reach the product goals through individual systems (which are developed in geographically distributed sites) depends on this phase. Additionally, the organization is facing a lot of challenges in this phase. Lastly, as it is a new product, none of the BUCs have reached the verification phase and beyond yet (see Figure 1).

The teams primarily responsible for the specification phase at BUC level are located in the chosen site. One of the system development team (responsible for BSUC level analysis) is also collocated at this site. Thus, we were able to cover the entire specification phase with coverage of both the service i.e. BUC and system i.e. BSUC level in the chosen site.

Thus, due to the scale of the product and reasons listed above it was not possible to start with an end-to-end VSM of the entire development process. However, after the completion of this study, VSM was conducted for several of the individual sites to get an end-to-end perspective.

3.3. Data collection

The two main sources of data were: the six workshops that were conducted as part of VSM and the corporate data on work items. This data included release plans, number of BOPs, BUCs and BSUCs their status, time spent in each phase, priority, number of impacted systems, initial size estimates, categorization as functional or quality requirements. We relied on extensive note taking, pictures of the white board and collection of yellow stickers for collecting data during the workshops. Notes from the researchers were corroborated and compiled into reports before the subsequent meeting where a summary was presented.

The first two authors of the study were acting as VSM facilitators who were responsible for data collection, moderating workshop sessions, analyzing the collected information and presenting the results. The third author served as an expert on FLOW and observed and discussed the process that was followed during the VSM.

One practitioner served as the "VSM Manager" who had the upper management support and could sanction the resources and time necessary to successfully complete the VSM activity. Five other practitioners were involved who were present during the workshops. These practitioners were chosen due to their knowledge and experience of the process and also their stakes in it.

Table 2 lists the roles and experience of the participants in the VSM workshops. The participants were chosen with the scope of the process and diversity of roles in mind for the following reasons: 1) No practitioner alone knows all the pertinent details for the end-to-end process 2) their involvement helps to develop a consensus on the current way of working, 3) identifying challenges as even if there are individuals well versed with the process they may not be aware of the challenges that the people working in the phase have to face, 4) prioritizing of challenges, otherwise we may sub-optimize by concentrating on a phase and not considering the end-to-end target 5) help to get the commitment on improvement actions and plan.

Furthermore, data about progress tracking for BUCs and BSUCs was extracted by individual system owners and was given to the researchers in spreadsheets. A relational database was created to reconstruct the links between BUCs and their break down at system level BSUCs. Lastly, examples of BUC and BSUC output were analyzed to triangulate the practitioners' opinion and perception of the quality of artifacts in terms of their content and level of detail.

Table 2: Participants of the VSM workshop

Role and responsibilities	Experience with the product	Experience with the company
Line manager, part of core team, system management	1.5	14
System manager, working on service-level analysis and system design	1.5	19
Program manager, budgets, priority settings, staffing requests	0	21
Architect, System design, requirements	3	20
Project Manager	0.6	14

To answer RQ1.3, in the last workshop, the practitioners were asked to filled out a questionnaire (with eight questions see Figure 11) anonymously. A seven point Likert scale was used to collect their feedback on statements that attempted to assess both the process for conducting FLOW-assisted VSM and its outcome. Table 3 maps the questions to the aspect being evaluated.

Lastly, a follow-up meeting was conducted where the product development team presented the status regarding implementation of the identified improvements. This provided another reflection point for assessing the practical usefulness of the FLOW-assisted VSM application in the case company.

3.4. Validity threats

This section reports the validity threats to the findings of this study and what measures were taken to alleviate them. The limitations of the study are also discussed here.

Reliability related threats influence the repeatability of a study e.g. dependence of the research results on the researchers who conducted it would diminish the reliability of a study [22]. To avoid inaccuracy in captured information (information and views captured during the workshops), two researchers took notes while the third researcher was moderating the workshops. These notes and their interpretation were triangulated among the researchers. Also, the results of the previous workshop were presented to the practitioners at the start of each

Table 3: The aspects evaluated in the questionnaire.

Question	Evaluated aspect	Perspective
Q1, Q2, Q3, Q4	Outcome	VSM + FLOW
Q5, Q6	Process	VSM
Q7, Q8	Process	FLOW

subsequent workshop to identify any misinterpretations by the researchers.

Internal validity is dependent on researchers' awareness or ability to control the extent of a factor's effect when investigating a causal relation [22]. Internal validity of the results was increased by the involvement of multiple practitioners having various roles and responsibilities in the development process. Multiple data sources were used in this study, the expert opinion on challenges where possible was triangulated with actual process metrics e.g. the perception of under-estimation of required effort was validated by comparing the estimated time and the actual time spent on analysis. Similarly, number of closed BUCs and BSUCs were used to validate the frequent reprioritization and quality of initial analysis.

Ericsson promotes continuous improvement in software development at all levels. We are aware that some participants in VSM related workshops were participating in other ongoing improvement initiatives. Thus, there is a risk that the challenges and improvements identified in this study cannot be solely attributed to the intervention in this study. Their involvement in multiple initiatives would have indeed influenced some findings of this study as well. But due to the different focus of the initiatives, while there were some overlaps between the outcomes of the study, a majority of the participants agreed that the use of FLOW-assisted VSM lead to new insights into the process that they did not have before.

Construct validity reflects whether the measures used by researchers indeed represent the intended purpose of investigation [22]. The opinions of practitioners on the process and outcomes of the VSM were collected using a questionnaire. To reduce the risk that questions might be misunderstood, the researchers were present during this session to answer any clarification questions. Furthermore, to reduce the influence that our presence may introduce and to encourage honest opinions, practitioners were asked not to put any identification information on the filled questionnaires. Furthermore, using a semi-structured format, complementary themes were explored in a group discussion after the practitioners had filled the questionnaire.

External validity is concerned with the generalization of the findings of the study outside the studied case [22].

Both methodologies, FLOW and VSM, have been individually validated in industrial studies. While their proposed combination has only been studied here on one product, however, it has been shown that the use of FLOW successfully addresses the limitations of VSM. The context of the study, the process of combining FLOW with VSM and under what conditions this combination becomes useful has been defined in sufficient detail for other practitioners to judge the relevance for their unique context.

4. FLOW-assisted VSM

The conventional VSM notation [2] is insufficient to capture the details of the knowledge intensive software development processes. This limitation has not been addressed in the adaptation of VSM for product development VSM either where the additional information captured is just regarding the iterations [3]. Figure 2 illustrates the notation used by Poppendieck and Poppendieck [23] to draw a value stream map. For each activity, the time taken in processing (while a team or a person is actively working on a request) and the total calendar time spent on a request is used to calculate value-adding time. Similarly, the waiting times in backlogs between various activities are calculated and captured with this notation.

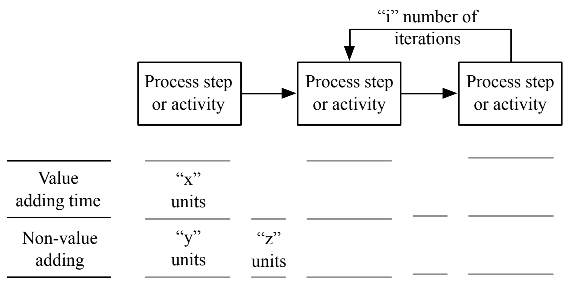


Figure 2: VSM notation (based on [23]).

As discussed in Section 2 FLOW is a methodology that systematically captures and visualizes channels for information flow and allows the identification of sub-optimal communication paths [12]. Its simple notation, ability to identify and visualize both documented and undocumented channels of information, make it a practical and relevant complement for use in conjunction with VSM.

The notation used by FLOW is presented in Figure 3. The notation is intentionally kept very simple. It is supposed to be used on a white-board or a piece of paper to discuss current and desired situations. An activity is symbolized by a rectangle with several incoming and outgoing flows. The rectangle acts as a black box and hides internal details of the activities and flows. It can be refined by another FLOW diagram [24]. The simplicity of the notation makes it more scalable compared to more complex modeling notations. A FLOW model

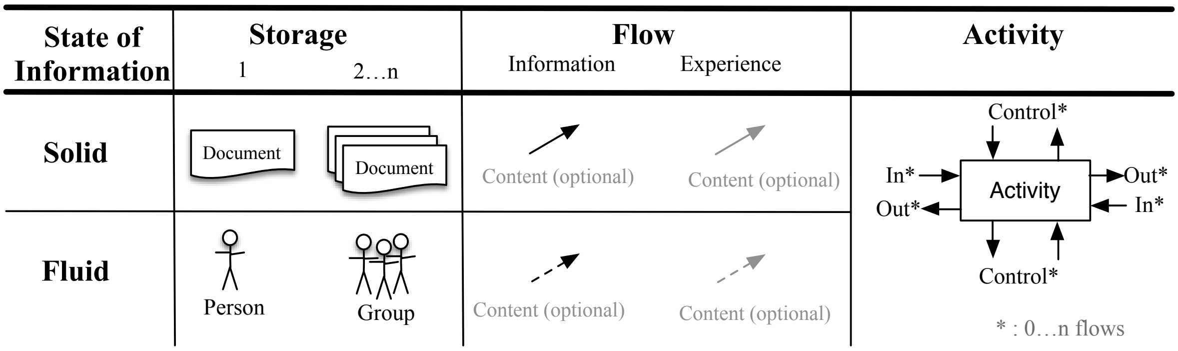


Figure 3: Symbols used in the graphical FLOW notation.

mainly depicts the network of information flow (people, documents, etc.) and the types of channels used.

The original FLOW notation as presented by Schneider et al. [21] and used in previous industry applications, uses slightly different symbols for documents and persons. A document is usually represented by the document symbol available in PowerPoint; a person is represented by a Smiley. Both symbols are readily available and do not require any specific editor. The stick figures and document symbols used in this paper are more familiar to the industry participants. This advantage exceeded the conformance with previous applications of FLOW.

VSM has five distinct steps; Figure 4 lists the steps and the main activities in each step (for details please see [1]). FLOW can be used in step S2, to elicit the CSM. However, we consider that it may not be necessary in every case to use FLOW for establishing a CSM. For example, if the challenges are related to long waiting times for items in backlogs, redundant reviews etc. then it may not be necessary to model different types of information, stakeholders and competence. So, depending on the challenges identified in step S3 (see Figure 4), we can decide whether the use of FLOW is justified. Thus, if there are challenges related to knowledge management or information flow, as was the case in this case study, then there is a utility of FLOW to identify the information needs, bottlenecks, broken communication paths and thus help in eliminating waste and creating a FSM in step S4.

In particular, the amount of information and its variety grows with the number of persons/sites involved, hence this has to be understood as well. Also, the information flows have to be understood to help explain why certain waiting times (i.e. the time when "work items" that move through the process are inactive) exist in a process.

In the context of VSM, we can benefit from FLOW in the following ways:

- First we can use it for eliciting the information flows in the product development process.
- Secondly, its graphical notation can be used to represent the current state map.
- Lastly, it can be used to identify and reason about changes in the current process to derive and document the FSM with improvements.

Five aspects listed in Table 4 are elicited for each activity in the FLOW elicitation technique. They refer to the incoming and outgoing flows of an activity (see Figure 3: activity symbol) and ask for the details of each flow.

In this research, we used FLOW to materialize the *Lean* principle of pull" for information-flow in the product development process. The idea is to use the pull-based approach starting from the down-stream phase and identifying what information and experience are required from an upstream phase of the process. In previous applications, the order of FLOW interviews did not consider the position of activities in the development process. Instead, the availability of interviewees was the primary concern.

Within FLOW interviews, the order of flows around one activity is usually:

- What are the activities you are involved in? Each one is detailed afterwards.
- What does this activity produce, and where do they go? The out-going flows respond to the pull of down-stream activities.
- What input is needed in order to produce this output, and where do you get it from? Very similar to data flow diagrams, there must be a source (incoming flow) of every information.

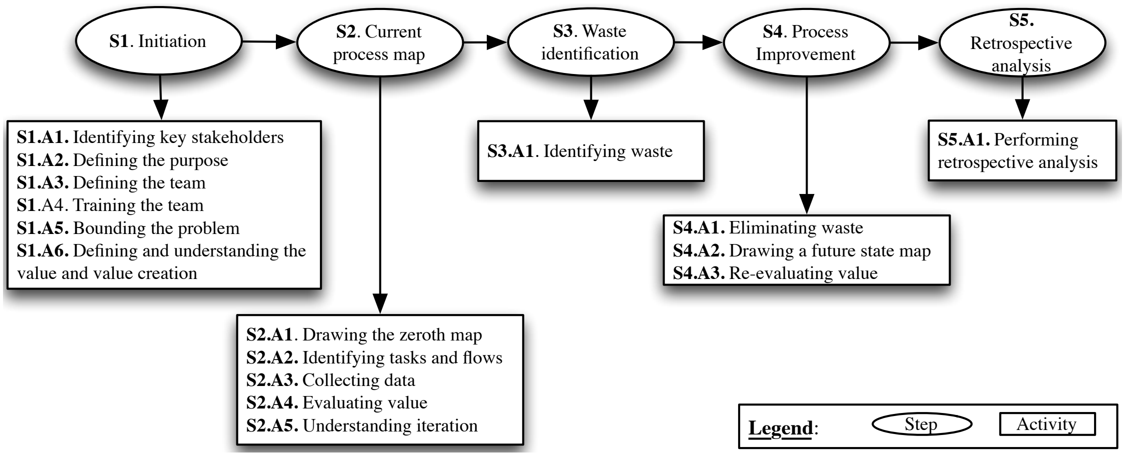


Figure 4: Overview of the VSM process based on [1].

- Who helps or directs the activity? The activity may be controlled and supported by checklists, guidelines, or experience. It is equally important to model where that experience is supposed to come from even if it is a person bringing this experience to the table (fluid experience).
- What tools or support do you need for performing the activity?

The questions are designed to crosscheck incoming and outgoing information with the receiver or source, respectively. Inconsistent answers indicate either inconsistent terminology (spec vs. requirements document), incomplete knowledge of the source or target, or misunderstandings among co-workers, e.g. as one of the practitioners said: *(I always thought our BUC would be used by system teams as high-level input to do a detailed analysis).*

While eliciting information for each aspect in Table 4 a distinction should be made between the current practice and the ideal practice e.g. when covering the content aspect, two sets of questions should be posed:

1. What information do you need? The aim here is to identify the desired state of the process.
2. What information do you currently get? This question aims to explore the current state.

Once this has been done for all the aspects of each of the activities in the CSM, one may compare the current state with the desired state to identify what is unnecessarily produced and what is missing. Similarly, having identified the personnel involved in each activity we can identify situations where there is an information overload on some key personnel. Other critical

Table 4: Elicitation tool based on FLOW used for each activity

Aspect	Questions
Tasks	What are the main tasks in this activity? e.g. write user stories
Roles	Who should be involved in this activity?
Output	What is the output of the activity? e.g. test cases
Intput	What is the input to the activity? e.g. user stories
Content	Details of the input required?
Who/What	Who has the information needed for this activity? What is the source of information?
Competence	What competence/knowledge sources are required to facilitate this activity? e.g. architectural expertise, knowledge repository

questions could include reflection on whether we want certain channels to be fluid or solid. The strengths and weaknesses of each type must be balanced, given that we are now looking at a globally distributed context with geographical and temporal distance can we avoid documenting the decision about the priority of BUCs in a local spread sheet.

5. Application of FLOW-assisted VSM

In this section, we report the results of applying the five step VSM process assisted by FLOW (as presented in Section 4).

5.1. Initiation

Since the product is relatively new (no BUCs have gone beyond the implementation phase) and very large, it was decided to reduce the scope of this VSM activity to the "specification" phase of the process. Another motivation was that all the necessary stakeholders for this phase were located in Sweden. The specification phase is responsible for formulating the work package, which takes the form of a *Business Use Case* (BUC) based on the business opportunities (BOPs) that have been identified (see Section 3.1 for an overview). The practitioners involved in the specification phase were also involved in a high-level analysis to identify dependencies, set priorities and indicate which individual systems would be impacted and therefore should be involved in further analysis and design of the BUC.

After negotiation with the product management the VSM goal for this product was to *"reduce the lead-time for end-to-end development process with a focus on improvements in the specification phase"*.

Besides the scale and other challenges listed above, we wanted to analyze the end-to-end development process to wasteful information that is not really required in the process downstream. So, we started with the "release" phases (see Figure 1 for an overview of the process) and wanted to "pull" the information needs from there onto the phases upstream. However, this proved too ambitious due to the size and complexity of systems involved in the product.

Therefore, we not only had to revert to the original scope of the VSM activity, but we also started with the first activity in the process that is the specification of a BOP and identified what information should it contain and what competence is required to define BUCs from it and we continued so on and so forth for rest of the activities in the specification phase.

We started the other way around as to refine the things, it was easier to start from the beginning. At the start, the information need is simpler and on a much higher abstraction level and that made asking what needs to be added, at least for this complex case work better.

5.2. Current state map

In first workshop with all the participants, an introduction to VSM, its process, prior experience of using it (especially in the case company), the goal for the current VSM activity and the rationale behind it were briefly presented. "VSM Manager" specifically related the company's goals and to the goal of the VSM and encouraged participation in the workshops as an opportunity to effect the future way of working.

After the introduction, we start with a very high-level view of the development process in the company (as discussed previous in Section 3.1 and shown in Figure 1). Then the participants were asked to draw the current state map on the board by following the work package i.e. a Business Use Case (BUC) through the process (in this case from "opportunity identification" till the development starts on the respective BUCs see Figure 1). With group participation, one typical BUC is followed through the process and the activities and tasks in the current process are mapped. The outcome of CSM from the first workshop is presented in Figure 5.

The process starts with the high-level analysis and breakdown of BOPs into BUCs by the core team. Then a team lead by a BUC author consisting of test manager, project manager and representatives from impacted system teams perform detailed analysis (called *BUC study*) on individual BUCs. A core team contact person is used if there is a need for further elaboration on BOP or to assess if the detailed BUC specification is in line with the market need. BUCs cover service level concerns that are inter-system interactions etc. These BUCs are then assigned to dependent system teams that perform *BSUC studies* to address system level concerns and ensure that various system development teams are able to develop and test in a coordinated manner. This is a typical process view that will be a starting point for VSM. However, in the next section when we look at the challenges (Table 5) this view is insufficient to address them.

5.3. Challenges

Both qualitative and quantitative information was used to identify challenges in the current process.

5.3.1. Expert opinion

Once the CSM was done, the participants were asked to reflect on the challenges that impede the product from reaching the goal of reducing the lead-time. To encourage participation, each practitioner was asked to write down at least three challenges on *"sticky notes"*. One by one, they were asked to read out the challenges they have listed, to give a brief description of the challenge and to place it on the CSM. The next participant would do the same and if there was an overlap with the already identified challenge they will be put the *"sticky notes"* together. This was supplemented by VSM facilitators' notes of challenges that were mentioned by the participants earlier while creating the CSM. An overlap here could have meant that something is perceived as a greater challenge than others. However, we knew from past experience that this only highlighted that a certain challenge was better known and talked about in the company.

Therefore, once the consolidated list of challenges was created, it was prioritized in terms of the impediment or hindrance it poses to achieving the stated goal.

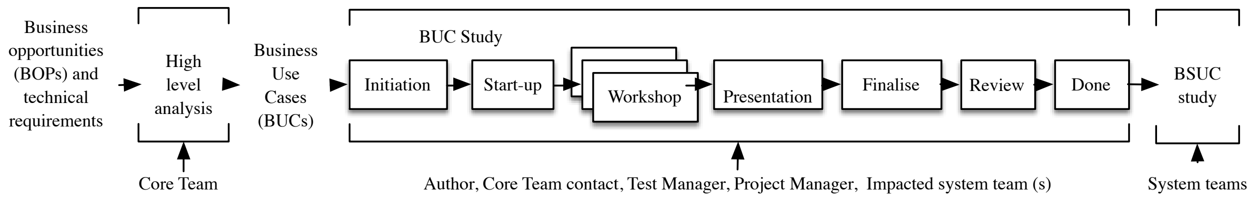


Figure 5: Current state map

Table 5: Perceived priority in terms of the hindrance these challenges pose to reach the stated goal.

ID.	Challenges	Votes
1.	Lack of high-level analysis and documentation of business opportunity analysis and breakdown to BUCs	12
2.	Lack of documentation/ guidelines/ template/ scope for system-level analysis	10
3.	Lack of a common understanding of what information and the level of detail should be the outcome of a service-level analysis	10
4.	Incomplete architecture models / specification / functional baseline	10
5.	Utilization of correct competence	4
6.	BUC not updated/ Neither treated as an alive document nor have an alternative document to track changes	2
7.	Lack of end-to-end BUC responsibility	1
8.	Incomplete information model	1
9.	Quality of business opportunity document (Requirements instead of Business Opportunities)	
10.	Challenges in communication with acquired companies having different maturity levels in the Telecommunication domain.	

The 100-dollar method was used to create a prioritized list of the challenges [25]. Each participant had ten dollars to spend on the list of the challenges, paying more for the challenges they believe are major hurdles to achieving the stated goal. This collective prioritization was useful to avoid any personal biases of participants and to get a consensus on what is collectively considered important. The challenges identified, and their relative priorities are listed in Table 5.

Moreover, a rudimentary root cause analysis was done for the top four challenges to understand the reasons behind the challenges. These are summarized in Table 6.

Given that most of the challenges were related to documentation, understanding information and competence needs it was evident that existing notation for describing the current state map (as shown in Figure 5) will be insufficient. Besides, the typical timing based analysis [26] [3], it was decided to utilize FLOW [12] that systematically captures information flows in the software development process. The current state map (as shown in Figure 5) was used as a list of activities that became input for FLOW where details of each activity were further elicited.

Lessons learned: If we find challenges in a CSM related to: documentation, lack of common understanding, or unsatisfied information needs, it points to the need to conduct information modeling in the context of value stream mapping analysis.

5.3.2. Quantitative analysis

Figure 6(a) shows the estimated time required for completing the analysis and the actual time taken to complete it. While Figure 6(b) shows the distribution of lead-times of various BUCs for the analysis.

By analyzing the lead-time for this phase, it was identified that unlike the typical initial estimate about the high-level study which should at most take 6 time-units in reality even the median value of actual time spent was higher than that with the worst lead-time of 23 time units (see Figure 6). This was consistent with the general perception (expressed in the workshops by practitioners), that the initial estimates are very inaccurate and underestimate the required effort.

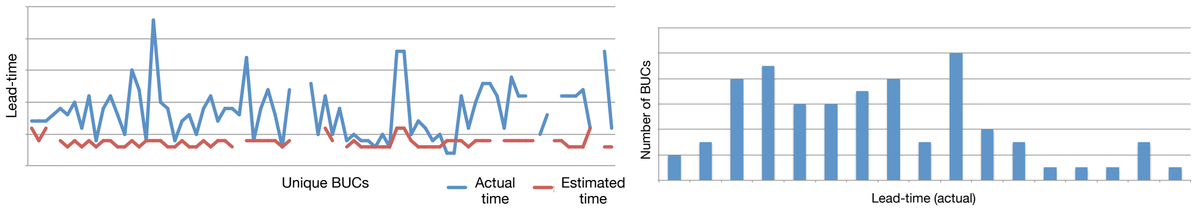
This may explain another observation about the current way of working, i.e. why the SPM team would not provide a prioritized list of BOPs and assign more work than the capacity of the analysis team (as expressed by practitioners in the workshops and also reflected in the long lead-times). Thus, the mismatch is because of the SPM team's expectation that the analysis is not an effort intensive task (as seen by the consistently underestimation of effort required as shown in Figure 6) while the reality is very different. Furthermore, the consistent under-estimation raise the question, whether the BUCs are broken down to the appropriate scope?

By looking at the release plan and current statuses of various BUCs we were able to identify two wastes:

1) Instead of *"build integrity in"* more immediate focus is on delivering functionality: very few i.e. less than 18% of non-functional requirements were assigned to the upcoming release while a majority was assigned to

Table 6: Challenges and the reasons behind them.

ID.	Challenge	Reasons
1.	Lack of high-level analysis and documentation of business opportunity analysis and breakdown to BUCs	Due to time constraint, there is a lack of documentation with over-reliance on the people in the core team. This sometimes also leads to inconsistent responses from within the core team.
2.	Lack of documentation/ guidelines/ template/ scope for system-level studies	Underestimated the need, had "hoped" that development can start without doing this level of analysis. Also, there is a lack of awareness that this analysis has to be done.
3.	Lack of a common understanding of what information and level of detail should be the outcome of a service-level study	It was decided early on that service-level analysis (BUC study) will be kept on a high-level. However, contrary to this directive the expectations of the system teams responsible for system development are different (they expected much more detailed level of analysis and specification). Between the service-level analysis which is relatively high-level and the system-level analysis which only covers the analysis for intra-system level details, a detailed analysis of inter-system level is missing for BUCs that require more than one system to deliver.
4.	Incomplete architecture models / specification / functional baseline	Underestimated the work, complexity and time it will take. Also, the value of having these for the product was underestimated to facilitate analysis and later on design and development. When reflecting on the causes, practitioners recollected statements like *"we will do it later"* or that the product is *"based on a legacy system so there is no immediate need"* for these documents. Another reason is weak governance i.e. to manage and enforce such standards based on the information needs.



(a) Comparison of actual and estimated time required for analysis

(b) Distribution of BUCs based on lead-time

Figure 6: Analyzing the estimated and actual time spent on BUC analysis

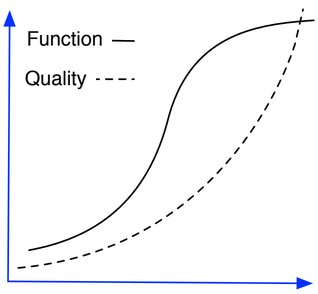


Figure 7: The trend in prioritization of Non-functional requirements

a later release. The approach is roughly visualized in Figure 7 where the current focus is on delivering functionality.

2) *"Partially done work"* instead of taking one item through the process: There were slightly fewer ongoing BUCs that were assigned to the subsequent two releases than the number of BUCs where some work has been done. For release 1 and 2: 45% of all the BUCs were assigned high-priority for development. However, only 47% of the BUCs being investigated belonged to this group. Similarly, 33% of the BUCs that have been analyzed did not belong to the high priority group. The reason for such a high number was mostly re-prioritization due to the changing market for this new product.

6. Results

In the following sections we have reported the results of the study in alignment with the research questions.

6.1. The use of FLOW interview-based technique (RQ1.1)

In this study, using the typical VSM process [1] the top four challenges identified by the practitioners were all related to information needs and flow in the organization (see Table 5). The existing VSM notation and method are incapable of investigating and improving such challenges. For example in this case, the level of detail in the current state map (see Figure 5) was insufficient to discuss solutions to the top four challenges. Therefore, we decided to use the FLOW notation and method to fill this gap in current adaptations of VSM.

The focus of the study was the specification phase and utilizing the *Lean* concept of "pull" [27] we wanted to identify what information will be required by downstream phases that must be created in the specification phase. The idea was to make a conscious decision in the upstream phases about aspects like which analyses to perform, in what detail, whether to document the results and what media to use for dissemination. Thus, avoiding any unnecessary analysis and documentation.

Therefore, to "optimize the whole" [23] we started with the release phase of the product and identified what activities are performed during this phase. Then using the FLOW elicitation template the inputs required for those activities were listed and relevant information like the sources and state were elicited. During this workshop, there were two main findings:

- First, we realized that applying the "pull" principle on the selected scope i.e. starting with the activities in the release phase and essentially identifying what needs to be documented and how in the specification phase did not work (see Figure 1 for an overview of the development process).
- Secondly, we found that eliciting the information and structuring it in the form of FLOW elicitation template (see Figure 8) on the fly did not work well in the workshop setting. It was not the template but rather the attempt to structure information while eliciting that became a problem in the workshop setting.

Although conceptually it made sense to identify the information needs in the last activity of the entire development process and then pull that required information from the preceding phase all the way to the start of the development process, but due to the following reasons it did not work in our case: 1) the scale of the product development (we lost most of the participants in the discussions, as such only the practitioners responsible for that phase were aware of the detailed tasks, inputs and outputs) 2) because it is a new product still under development and none of the BUCs have reached even the service level testing yet, we realized that soon it became an exercise of assuming what information needs will exist and what and how the required information is assumed to be delivered.

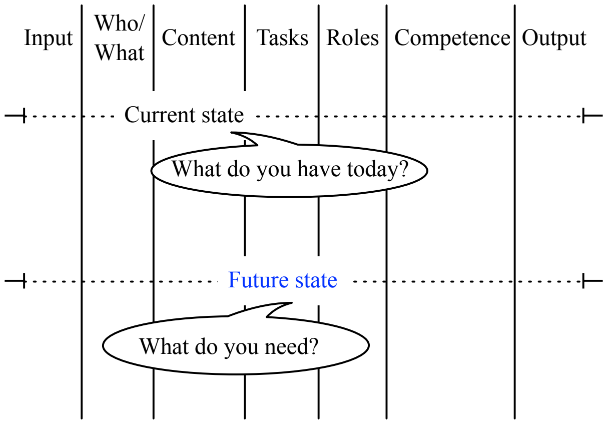


Figure 9: Elicitation of information based on FLOW template in a workshop setting.

The reason for the second observation we consider is that structuring the information on the white-board in the FLOW template form was not adding value. In a workshop setting, it became very difficult to moderate e.g. some people were now listing outputs and someone mentioned an input and the discussion started on that.

Based on these two observation, we made two changes to our approach, we decided to focus only on the scope of the current VSM i.e. the "specification" phase instead of the end-to-end development process and we used FLOW in the background, guiding the elicitation and to structure and report the findings and resulting process maps (made using the FLOW notation) to the practitioners.

Therefore, taking the activities in the current state map (see Figure 5) one by one we elicited the six columns on a white-board as depicted in Figure 9. Having elicited this information we used FLOW notation to visualize it. Thus, researchers acted as the interface between FLOW templates and practitioners thus avoiding the need to train practitioners with the FLOW templates or methodology.

6.2. *Improvements and the future state map (RQ1.2)*

Based on the analysis reported in Section 5.3 the following considerations were highlighted when creating the future state map and action plan for improvements. Section 6.2.1 lists the contributions of a typical artifact flow done as part of VSM and Section 6.2.2 identifies the contribution of performing information flow analysis using FLOW as suggested in this study.

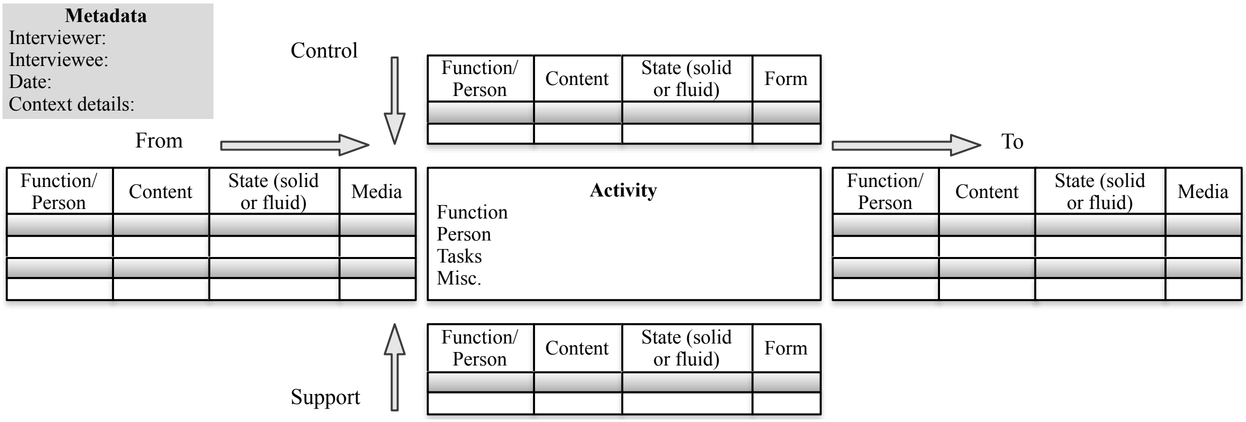


Figure 8: Interview-based elicitation form used in FLOW

6.2.1. *Improvement considerations derived from typical VSM analysis*

The data on waiting times and productive time was not collected in the company for this product therefore the typical time-line analysis (as shown in Figure 2), which is done as part of VSM activity could not be performed. However, the analysis was guided by the principles of Lean and what is typically considered waste in *Lean* software development [4]. In the remainder of the section, we highlight the principles and wastes that led to insightful discussions and improvement proposals for the process.

According to Poppendieck and Poppendieck [4] the two ways to reduce cycle time for a process are to achieve a steady arrival rate for work and to look at how the work is processed and remove variability in the processing time.

For this product, we found that the requirements were handed over in large batches and without adequate prioritization. The perception was that there are a high number of service-level analyses (BUC studies) being done in parallel and it was seen in the data as well (see Section 5.3.2). There were as many ongoing and completed service-level analyses that were not even highly prioritized as the ones with a high priority. This was done without taking into consideration the capacity of the organization. Therefore, the current push based approach should be replaced with a pull driven approach that takes into account the capacity of the development organization and the priority of the business opportunity.

One indication of too many ongoing studies is reflected by the dissatisfaction with the quality of information about the breakdown of BOPs to BUCs (see Table 5). Another indication is the sheer overwhelming feeling that the teams have of working with service-level analyses (BUC studies). Yet another quantifiable measure is the fact that none of the BUCs have reached the system-test or the service-test levels in the product development life-cycle.

When seeking explanation for the reasons behind certain challenges, it was noticed that a majority of the challenges were due to the secondary importance given to the documents necessary to ensure the non-functional attributes of the system. For example, in Table 6, when asked about the reasons why architectural model and functional baseline were not created in time, the reasons include that the value was underestimated or that there was a time pressure. A similar mindset towards quality was noticed in the release plan where functional requirements were prioritized over non-functional requirements (see Figure 7).

Furthermore, the consistent under-estimation (see Figure 6(a)) raised the question, whether the BUCs were broken down to the appropriate scope?

For the future, the key improvements identified were as follows:

- Release frequently and take in smaller batches [4]. “Do not keep piling up work that cannot be used immediately” [4]. This ensures a continuous flow of requirements instead of batches/chunks of business opportunities
- Improve prioritization of business opportunities and BUCs to address the issue of parallelism
- Efficiently deliver new BUCs to the development organization (*pull driven*) also helps to address the issue of parallelism. No need to keep piling up BUCs (through hastened BUC studies) if the development organization does not have the capacity to implement them.

6.2.2. Improvements derived from the use of FLOW

Lesson learned: The current state map (see Figures 1 and 5) did not allow us to analyze the process in sufficient detail to address the identified challenges in the process (see Table 5). With these process descriptions, we could have modified the process a bit on a high-level, however, improving it would not have been possible without the awareness of additional information in each step of the process.

With the improvement suggestions (as discussed in Section 6.2) and key challenges that were related to information needs (see Section 5.3) in consideration, the FLOW methodology was used. Using the high-level list of activities (see Figure 5) as input, the FLOW methodology helped to systematically elaborate and visualize the state and storage of information, and also the information and competence required to perform these activities. This facilitated to identify causes for certain issues where the flow of information was broken and to identify concrete improvement actions.

Figure 10 shows the outcome of the VSM activity using the FLOW notation. This figure is not intended to explain the development process at Ericsson, rather the intention is to give a flavor of the size and complexity of the process being analyzed. Also, any practitioners applying the proposed approach can see what can be expected as an outcome from it. The improvements in the process are coded in blue color and are listed below:

- The information required to define a business opportunity (BOP) was established
- Prioritized list of BOPs based on the market window by the product management team.
- Acceptance criteria for an initial BUC definition to become sufficient for detailed analysis were established. This entailed that the core team should provide a statement about the scope, list of impacted systems. A non-technical description of the high-level value, i.e. what do we want to achieve with this BUC. Furthermore, analysis of the information model and architecture should be made available for service level analysis (BUC study).
- Involving product managers after initial analysis to validate if the understanding of BOP is correct and the direction of BUC specification is aligned with the expectation in the BOPs.
- Identified what is missing in the detailed BUCs that needs to be added for system teams to perform system level analysis.
- Added an activity between service level analysis and system level analysis that should cover the inter-system concerns in more detail for BUCs that require more than one system to deliver. The BUC analysis lead along with the impacted system-teams is required to develop an initial specification of inter-system interfaces and a preliminary schedule for development.
- It was decided to involve the *Cross Functional Teams* early on in the BSUC study activity.
- Detailed list of tasks that should be performed as part of the system-level analysis (BSUC study) were detailed as well.

Furthermore, with the visualization of overall information flows other challenges that were not visible before were highlighted. For example, it was observed that the prioritized list of BUCs was not available to all the sites and was maintained in a spreadsheet. Similarly, given that the quality of architecture description was considered an issue, it was visible that the resources (only two persons whom are assigned to communicate with all the system teams and to maintain/update the architecture description apart from their other commitments being part of the core-team) were insufficient.

Figure 10 shows the final version of this analysis. The items in blue are what have been identified as improvements in the current process, which included: assignment and clarification of roles, establishment of new activities, artifacts and details of their content and the need for competence.

This simple representation provided sufficient support to analyze the process further and identify additional challenges, e.g. deciding about what information should be documented and where it is more optimal to rely on face-to-face communication.

Compared to the current state map (see Figure 5) that was drawn without the use of FLOW methodology, it is visible that these improvements could not have been identified and conceptualized without the use of FLOW.

6.3. Practitioners' perspective (RQ1.3)

To elicit practitioners' feedback on the process and outcome of FLOW-assisted VSM a questionnaire was used. The questions along with the responses are depicted in Figure 11.

Regarding FLOW, overall the practitioners perceived that its use led to more insightful discussions and should be used at the company. No negative opinion was expressed regarding further its further use in the company for process improvement (see response to Q8). For question Q7 three practitioners agreed that the use of FLOW led to more insightful discussions while one practitioner disagreed with the statement.

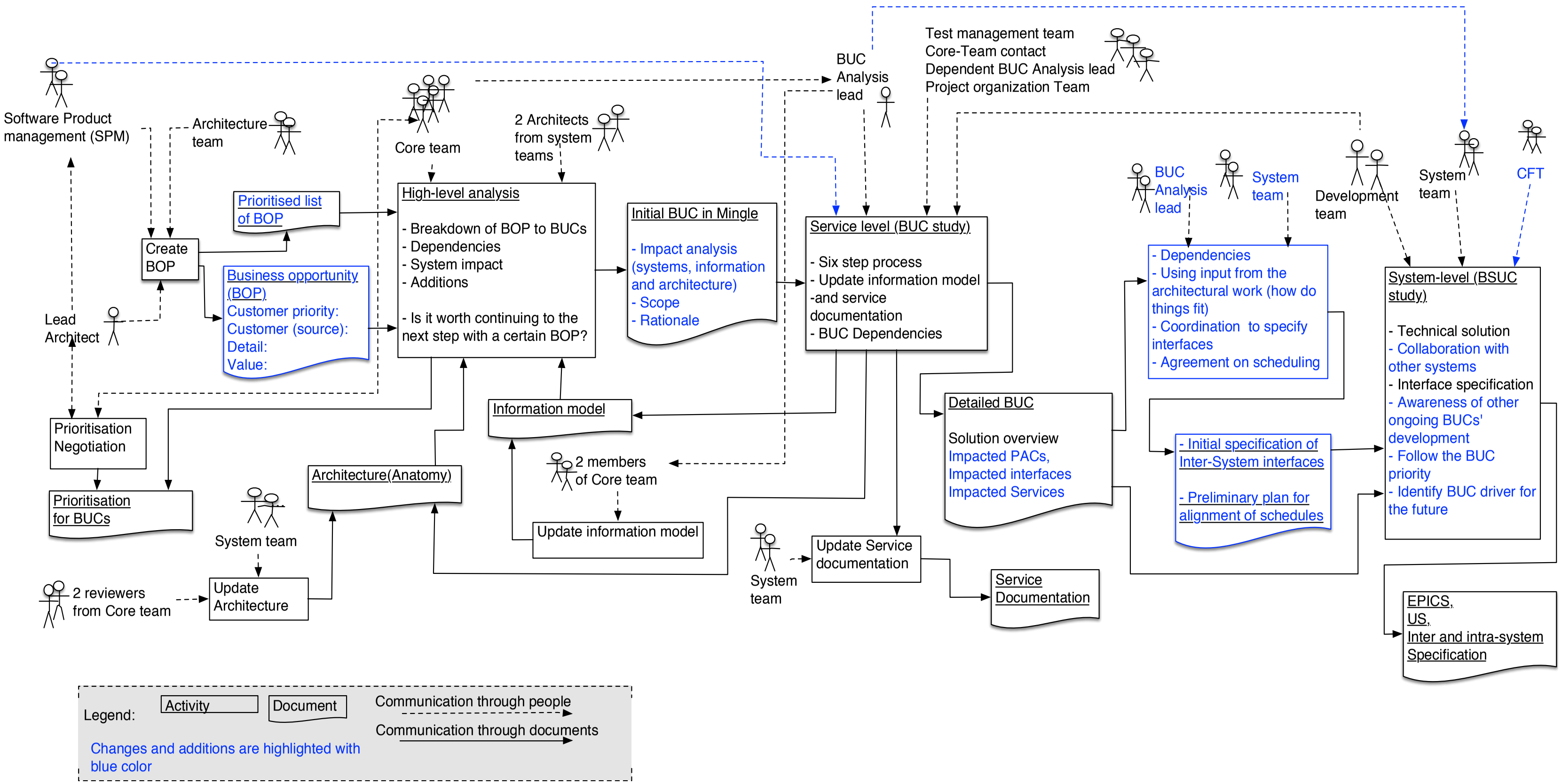


Figure 10: Future state map

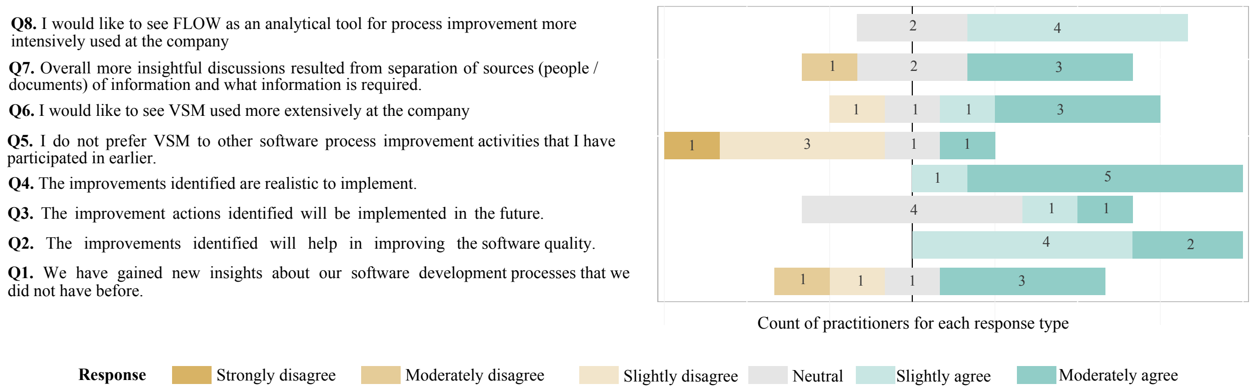


Figure 11: Feedback questionnaire results for the six practitioners.

While expressing their feedback on the outcome of use of FLOW one practitioner found the notation and the final outcome where the end-to-end process overview, artifacts (key information items), and stakeholders is visible on one sheet is really good *"Best outcome of the exercises"*. Another remarked, *"it is good to depict stakeholder interaction in the process"*.

Similarly, as seen in responses to questions Q5 and Q6 in Figure 11, the practitioners received VSM largely positively. Only one practitioner disagreed with the continued use of VSM and use of VSM as the preferred method for process improvement in the company. However, since no information about the other improvement activities that the participants may have previously been involved was elicited, and due to the anonymity of questionnaire respondents, we could not investigate the negative feedback further.

Questions Q2 - Q4 had no negative response from any of the practitioners showing an overall agreement that the improvements identified in the workshops will lead to improved software quality. The results also show confidence in the improvements being realistic to implement (see responses to Q2 and Q4 in Figure 11. Besides the agreement on the realistic nature and a likely positive impact of the improvements, only two practitioners think that they will be implemented in the organization.

Interestingly, in a follow-up at the company six months after the conclusion of this study (see Section 6.4) it was found that contrary to the skepticism expressed by the practitioners the improvements had been implemented in the organization.

There were mixed responses to the statement that they gained new insights into their development process (see question Q1 in 11). Half of the respondents agreed that they did gain new insights and two disagreed. It is noteworthy that this is still a positive result, as everyone needs to have the same understanding for such a large process. Thus, half of the participants gaining new insights is important to assure an end-to-end perspective where everyone knows what information they should provide, or shall receive and facilitate its flow.

One criticism expressed by a participant was that *"the improvements identified were less than what they had expected from the VSM activity"* she also commented that there was a *"high overlap"* during the workshops that were spread overtime.

Two practitioners commented that the workshop highlighted the break in the flow and brought focus to it. Other positive comments highlighted the "granular" level at which analysis was done and how it helped to focus on optimizing towards the overall flow instead of small improvements.

6.4. *Follow-up after six-months*

The practitioner who acted as the "VSM Manager" during this study conducted a follow-up and reported that almost all the improvements listed in Section 6.2 have now been implemented which address the top. Such as:

- "One slider" implemented in BUC study to give overall summary of business needs and impact on sub-systems – Challenge 1 in Table 5
- Assigned BUC responsibility for the entire lifecycle – Challenges 6 and 7 in Table 5
- To deal with parallelism related problems, a constraint on number of BUCs being developed concurrently has been introduced
- Mechanism to deal with dependency, structure on how to handle cross-PAC dependencies is now in place. Interface descriptions are to be defined before concluding a BUC study – Challenge 3 in Table 5

- service and system models and governance for these models have been established – Challenge 4 in Table 5
- BSUC study process has been established – Challenge 2 in Table 5

Furthermore, having seen the value that VSM analysis can bring they have also collecting data on backlog size and waiting time and it has been connected to the generic measurement dashboard in the company. The process is still not "pull driven" which is a completely different way of thinking and is not intuitively obvious. We are often used to thinking in processing batches. Having said that, the changes that have been implemented to lay the groundwork, e.g. having the quantitative data on waiting times that will now be collected, will help to motivate the transition in the teams.

Typically, companies are hesitant to make an investment in change, unless they are fully convinced of the benefits. Since, almost all the improvements identified in this study have been implemented it shows their confidence in the process that was followed in the study and its outcomes.

7. Reflections

FLOW methodology was used to guide elicitation and its notation was used to structure information and represent the process using it. The simple notation was easy to adopt for us and practitioners shared the same opinion about the process. It helped us to identify points that needed further clarification and to pinpoint people and artifacts that may be potential bottlenecks in the process.

Key lessons from using FLOW to assist VSM in this case are the following:

- The scope cannot be too broad in such a large product, instead of the entire end-to-end process start with individual phases and still have practitioners covering various perspectives from the overall process.
- The maturity is a factor, and people had a tendency to fall back on the old processes they already know and put them into the future state map (i.e. hard to think on a "green field").
- The notation should not be enforced on the participants, this is something that the researchers should do on the background.
- Even for a very large-scale process, we could accurately capture the future state map on a level that the practitioners agree and commit to the findings, and this could be represented on a single A3 (i.e. the whole process)
- The companies having a similar structure (multiple products, need for break-down from abstract to more detailed and then allocating to sub-systems - SoS approach - can benefit from some learnings with regard to how to actually structure the process)
- Information that was elicited in just a few workshops could most likely not have been gathered by keeping on the abstraction level of the current state map, and that the distinction between information flow types etc. helped to critically appraise the process and identify bottlenecks e.g. the new activity that is added to bridge the information gap between service level analysis and system-level analysis (see Figure 10).
- Overall, the experience indicates that the approach scales. This was shown with the ability to capture sufficient detail of the process in six workshops to capture the current way of working, analyze it, and reach consensus on what challenges to address and how to address them for a large scale product (for a discussion on the scale, see the context description in Section 3.1).
- Based on the experience from this case, the combination of VSM with FLOW can be useful in cases where the development process is not well defined, if there are challenges related to information exchange, or if the organization is involved in distributed development.

Having diversity in participants that gave a sample of various stakeholder roles from the teams involved in the specification phase of the product development process at the company enabled a systemic end-to-end view of the specification phase and in general the concerns that later phases may have. Their participation ensures that they understand the problems and challenges in other areas that they may not directly be working in. Also, it helps to get a buy-in from stakeholders who may need to do additional work in the future to improve the quality and reduce the lead-time for another process that consumes their output (although it may not reduce their lead-time). In short, it helps to achieve the end-to-end perspective within the specification phase and avoids the pitfalls of sub-optimizing in silos, although we could not reach the same goal for the entire product development process.

Using *"sticky-notes"* to identify challenges ensured participation and gave everyone the opportunity to bring up the challenges they face and consider as a hindrance. Furthermore, by using a variant of the 100 Dollar method for prioritization [25] where each participant has an equal opportunity to highlight what they consider is crucial to address if the organization is to reach the stated goal. In this study, it was seen that although some challenges generated a lot of discussion, but they were not high-up on the prioritized list.

Quantitative analysis (see Section 5.3.2), not a lot of data was available as the process was not well defined and it took considerable effort to extract data from different sites and to we had to manually connect the data from different silos. Yet the utility became apparent not only to triangulate the qualitative input from the practitioners (e.g. the amount of parallelism in the process) but also to identify new challenges that they had not identified (prioritization of functional requirements over non-functional requirements).

Apart from the participation in workshops and making data available to researchers no further investment was required from the practitioners. From the organization, the investment takes the form of ensuring that the practitioners are available for the workshops. For researchers, taking on the role of facilitators the investment includes understanding the context, familiarizing with the technical jargon, data analysis, reporting of results and conducting the workshops. In this study, it took approximately 16 hours of workshop time to arrive at the FSM.

7.1. Reflections from the FLOW expert

From the perspective of FLOW, the combination with VSM brought a number of insights: In previous cases, FLOW had been used in a series of interviews. In each interview, one person was asked about his or her concrete activities, and their output and input information. In between two interviews, the graphical FLOW model was updated and extended accordingly. In several cases, flows between two activities were not described consistently from both sides. Other flows were not mentioned at all, leading to "dangling" flows, or obviously unclear flow of information. All those cases triggered focused in-depth questions in the next interviews. Thus, graphical flow models were not shown in interviews, but used in the back office to prepare and focus subsequent interviews. Final FLOW models were only displayed to participants after the series of interviews. There was a summary workshop around the FLOW diagram that had emerged, with a number of observations by the researchers. Activities for improvement were derived in those final discussion workshops. Due to the size and time restrictions, bilateral interviews were not an option in the case described in this paper. There were too many parts to be discussed and too many people to be involved. Trying to fold a series of interviews into a single workshop (or maybe two workshops) did not work well, as outlined above. In particular, there was no opportunity to update a FLOW model and then reflect on inconsistencies, dangling flows, and the like. The adapted solution used here relaxed the scheme of asking for outgoing, incoming, controlling flows a little. Keeping those aspects in mind helped to elicit important information. Doing it in a more informal way than during typical FLOW interviews was advantageous for the workshop setting.

The state map in Figure 10 is large and rather complex, but not exceptional for a final FLOW diagram. Initial FLOW models tend to be very simple, covering only one or two activities and their related flows. The models grow with every person interviewed. Figure 10 was not updated iteratively after each interview, but drawn in one step after the workshops that served for eliciting information. Once the FLOW state map in Figure 10 existed, however, it was used in a very similar way as previous FLOW models in other cases during the final discussion workshop: Participants would point to it, step up to it, correct it, and refer to information flows without necessarily calling them information flows. The diagram helped focus discussions and gain a good overview. Such a large diagram is difficult to draw from scratch, i.e. from the textual notes alone. However, this adapted variant of information gathering and visualization was obviously more appropriate in this context. This new approach of creating a FLOW model will feed back into the FLOW methodology.

8. Conclusion

We have presented a process for conducting VSM that performs both the typical artifact flow analysis and additionally the information flow analysis. For latter, we have utilized the FLOW approach to overcome the limitations of existing VSM method and notation. The process for FLOW-assisted VSM and its outcome are presented in this paper to help other researchers and industry practitioners to apply it in their context.

A case study that was conducted to evaluate the usefulness of the combination for a large-scale product at Ericsson AB has contributed to answer the following research questions:

RQ: *How useful is the combination of FLOW and value stream mapping in large-scale software development?* In cases, such as the one reported in this paper, where the challenges are mostly related to communication and information flows, the use of FLOW methodology will be beneficial to provide a systematic approach to identify, visualize and evaluate information flows.

For new product development, where the processes are less mature and the product is still in its infancy, a typical quantitative analysis of waiting times for artifacts is often not possible and perhaps not as important. The more significant aspect is to establish a process that encourages taking the end-to-end perspective and improving the flow of information and artifacts through the process.

FLOW provided a systematic approach to elicit, visualize and analyze information flows. The lightweight approach with intuitive notation provided the necessary support missing in the current VSM to analyze information flow and effectively streamline the value stream

VSM provided a broader framework that brings a systematic approach of reflecting on the value stream to eliminate waste and identifying improvements. While FLOW was successfully used to elicit and model information flows. Furthermore, the artifact flow analysis, where the time-line for the value stream is assessed, which is typical in VSM, helps to quantify the implications of challenges in information flow.

RQ1.1: *Can the interview-based technique for FLOW be transferred directly to a larger group of people (in a workshop setting), if not, how can it be adapted?* As shown in Section 6.1 the content elicited in the typical FLOW interviews was found sufficient for information flow analysis. However, in a workshop setting the elicitation mechanism had to be adapted to the situation. Presence of multiple practitioners posed a challenge as they had multiple perspectives and understandings of the process due to their different roles in the organization. Instead, we used it to our advantage by putting the elicited information on a white board (as shown in Figure 9). This helped to develop consensus on both the current and the desired state of information flow in the process.

RQ1.2: *Which wastes and improvements could be identified with the combination of FLOW and VSM?*

Using VSM assisted by FLOW helped to create a specification process, identify stakeholders and artifacts, and detailed content of these artifacts. This was essential for the processes that consume the output of the specification phase. This approach identified improvement opportunities in the current process and helped visualize them in the future state map.

It was found in this study (see Section 6.2) that there were challenges and improvements that were uniquely identified through typical artifact-flow analysis (typically done in VSM) and information-flow analysis (atypical of VSM in software development).

RQ1.3: *How do practitioners perceive the process and outcomes for the VSM supported by FLOW?* In an anonymized questionnaire, practitioners responded favorably to the use of VSM and FLOW and would like to see its continued use in the company. They were all positive about the realistic nature and the likely benefits of the identified improvements on software quality. However, there was a fairly weak agreement on whether the use of VSM and FLOW generated new insights about their process.

Another significant benefit of FLOW notation was its simple yet powerful notation that was intuitive and even for a large scale product development process, an overview of the entire process could be represented on a single sheet of paper This was highlighted by the participants of this study.

Acknowledgment

The authors would like to thank Kennet Kjellsson from Ericsson AB, Sweden who facilitated this work. We are also grateful to all the practitioners for their participation and feedback. This work has been supported by ELLIIT, a Strategic Area within IT and Mobile Communications, funded by the Swedish Government.

References

[1] M. Khurum, K. Petersen, T. Gorschek, Extending value stream mapping through waste definition beyond customer perspective, Journal of Software: Evolution and Process 26 (12) (2014) 1074–1105.

[2] M. Rother, J. Shook, Learning to See: Value Stream Mapping to Add Value and Eliminate MUDA, spi edition Edition, Lean Enterprise Institute, Brookline, Mass., 1999.

[3] H. L. McManus, Product Development Value Stream Mapping (PDVSM) manual, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 1st Edition (2005).

[4] M. Poppendieck, T. Poppendieck, Lean software development: an agile toolkit, Addison-Wesley Professional, 2003.

[5] S. Mujtaba, R. Feldt, K. Petersen, Waste and lead time reduction in a software product customization process with value stream maps, in: Software Engineering Conference (ASWEC), 2010 21st Australian, IEEE, 2010, pp. 139–148.

[6] K. Stapel, E. Knauss, K. Schneider, Using FLOW to Improve Communication of Requirements in Globally Distributed Software Projects, in: Proceedings of the Workshop on Collaboration and Intercultural Issues on Requirements: Communication, Understanding and Softskills (CIRCUS), Ieee, 2009, pp. 5–14.

[7] M. Staron, W. Meding, Monitoring bottlenecks in agile and lean software development projects - a method and its industrial use, in: Proceedings of the 12th International Conference on Product-focused Software Process Improvement, PROFES'11, Springer-Verlag, Berlin, Heidelberg, 2011, pp. 3–16.

[8] K. Petersen, P. Roos, S. Nyström, P. Runeson, Early identification of bottlenecks in very large scale system of systems software development, Journal of Software: Evolution and Process 26 (12) (2014) 1150–1171.

[9] K. Petersen, M. Khurum, L. Angelis, Reasons for bottlenecks in very large-scale system of systems development, Information and Software Technology 56 (10) (2014) 1403–1420.
[10] A. Rausch, C. Bartelt, T. Ternit, M. Kuhrmann, The V-Modell XT Applied - Model-Driven and Document-Centric Development, in: Proceedings of the 3rd World Congress for Software Quality, 2005.
[11] A. Cockburn, Agile Software Development, Addison Wesley, 2002.
[12] K. Stapel, K. Schneider, Managing Knowledge on Communication and Information Flow in Global Software Projects, Expert Systems - Special Issue on Knowledge Engineering in Global Software DevelopmentSubmitted for consideration to.
[13] K. Stapel, K. Schneider, FLOW-Methode - Methodenbeschreibung zur Anwendung von FLOW, Tech. rep., Fachgebiet Software Engineering, Leibniz Universitt Hannover (2012). URL `http://arxiv.org/abs/1202.5919`
[14] N. Zazworka, K. Stapel, E. Knauss, F. Shull, V. R. Basili, K. Schneider, Are Developers Complying with the Process: An XP Study, in: Proceedings of the 4th International Symposium on Empirical Software Engineering and Measurement (ESEM '10), IEEE Computer Society, Bolzano-Bozen, Italy, 2010, best Paper.
[15] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, J. Still, The impact of agile practices on communication in software development, Empirical Software Engineering 13 (3) (2008) 303–337.
[16] S. Winkler, Information flow between requirement artifacts. results of an empirical study, in: Requirements Engineering: Foundation for Software Quality, Springer, 2007, pp. 232–246.
[17] B. Berenbach, G. Borotto, Metrics for model driven requirements development, in: Proceedings of the 28th International Conference on Software Engineering, ICSE '06, ACM, New York, NY, USA, 2006, pp. 445–451.
[18] T. DeMarco, Structured Analysis and System Specification, Prentice Hall, Englewood Cliffs, NJ, 1979.
[19] I. Kwan, D. Damian, M.-A. Storey, Visualizing a requirements-centred social network to maintain awareness within development teams, in: Proceedings of the First International Workshop on Requirements Engineering Visualization, IEEE, 2006, pp. 7–7.
[20] A. Wise, Little-JIL 1.5 Language Report, Tech. rep., Department of Computer Science, Univ. of Massachusetts, accessed: 2015-02-21 (2007). URL `http://laser.cs.umass.edu/techreports/06-51.pdf`
[21] K. Schneider, K. Stapel, E. Knauss, Beyond documents: Visualizing informal communication, in: Proceedings of the 2008 Requirements Engineering Visualization, REV '08, IEEE Computer Society, Washington, DC, USA, 2008, pp. 31–40.
[22] P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering, Empirical software engineering 14 (2) (2009) 131–164.
[23] M. Poppendieck, T. Poppendieck, Implementing lean software development: From concept to cash, Addison-Wesley, 2006.
[24] K. Schneider, D. Lbke, Systematic Tailoring of Quality Techniques, in: World Congress of Software Quality 2005, Vol. 3, 2005.
[25] P. Berander, P. Jönsson, Hierarchical cumulative voting (hcv) - prioritization of requirements in hierarchies, International Journal of Software Engineering and Knowledge Engineering 16 (6) (2006) 819–850.
[26] K. Petersen, C. Wohlin, Measuring the flow in lean software development, Software: Practice and Experience 41 (9) (2011) 975–996.
[27] J. P. Womack, D. T. Jones, Lean thinking: banish waste and create wealth in your corporation, Simon and Schuster, 2010.