2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)

Measureability of functional size in Agile software projects: Multiple case studies with COSMIC FSM

Tuna Hacaloglu
Department of Information Systems Engineering
Atilim University
Ankara, Turkey
Informatics Institute
Middle East Technical University
Ankara, Turkey
tuna.hacaloglu@atilim.edu.tr

Onur Demirors
Department of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
onurdemirors@iyte.edu.tr

***Abstract*— Functional size measurement (FSM) has been used in software engineering for decades as a main driver for estimation and significant input for other various project management activities throughout the project life span. To apply FSM accurately at the early stages of software development process, especially for estimation purposes, functional user requirements need to be available in detail as required by the adopted FSM method. However, in agile software development, requirement specifications, in general, are kept minimal. For this reason, the adjustment of the requirements to the necessary granularity level has been articulated as one of the barriers preventing the diffusion of FSM practices among agile teams. In this paper, we take a closer look at this problem in order to investigate the usability of FSM and to reveal FSM related challenges empirically through case studies on real agile projects from different software organizations. This study also provides a snapshot of agile organizations in terms of requirement specification and estimation related practices**

Keywords*— *agile, functional size, COSMIC

I. Introduction

Effort estimation has been among the focal points in various studies as regards agile software development literature. Even though agile is a mature paradigm now, when viewed from the effort estimation perspective, it is still dominated by subjective estimates; such as use case points and story points [1]. Despite their frequent usage, especially story points have not always been an appropriate instrument for estimation purposes due to their subjective nature, thus causing difficulties in terms of reusing prior estimates [2]. Since they are assigned based on the estimators' subjective judgment, when agile teams utilize story points the used measurement scale becomes unique and denominated to that specific team [3]. When there is a need for a measure across projects, functional size measurement (FSM) becomes more prominent because it is measured based on standardized methods [4] rather than being an estimate produced as a result of subjective judgments which can quickly vary according to the estimators' perceptions and experience [2]. Objective size measurement allows for establishing estimation models, benchmarking, internal process improvement and project governance [5]. In addition to being the input for project estimations; through functional size, organizations can keep track of their projects' scope change, reach an agreement between the supplier and acquirer, and assess and improve their processes by normalizing their performance and quality measures [6].

However, the applicability of FSM in agile software development projects has not been regarded as a straight-forward approach. The reason is that, in agile methods, requirements are generally kept as short as possible, and written in a limited scope for main functions alone [7], and supported with tacit knowledge. In turn, such knowledge is acquired and developed by the team during the daily/ weekly meetings according to the agile model it adopts. Agilists value these face-to-face communications [8]. Yet, not all this knowledge finds its way into documents; as a consequence, requirements are not written in a form containing detailed information. Based on a literature review study by Hacaloglu and Demirors in [9], it is observed that researchers, for example [11], emphasize the absence of detail and lack of formalism limiting the applicability of FSM practices, such as those in COSMIC FSM where the lack of details do not allow measurer to identify required measurement components; for example, "data groups" in COSMIC FSM. In a study by Salmanoglu, Ozturk, Bagriyanik, Ungan and Demirors [10], the need for adjusting the template of analysis documents in a form containing an equal level of abstraction is emphasized. Still, agile teams perceive such efforts costly and, as a result, avoid them [11]. In 2010, Buglione and Trudel [5] stated that agile practitioners tend to resist replacing story points with FSM because they think out on the grounds that they are familiar and successful with story points and do not want to spend time for an unfamiliar method, which, they believe, does not increase the quality and timeliness of the project.

In light of this background, the purpose of this study is to explore the applicability of FSM in agile software development projects empirically. We present the results of a series of case studies to understand the applicability of an ISO-compliant FSM method, COSMIC FSM, in agile settings. More specifically, we focus on the difficulties of the measurement process and the compatibility of the model with agile methods.

The intended goal is that by identifying agile-specific problems in applying FSM methods, we can contribute to the development/improvement of methods that better fit agile philosophy. This article is structured as follows: section 2 presents the related literature. In section 3, the research methodology including the case study is described. In section 4, the results obtained in terms of the challenges of applying the COSMIC method in agile requirements artifacts are presented and discussed. In section 5, we offer validity threats and in section 6, we present the conclusion and future works.

II. Related Literature

In the literature, there exist studies appraising the importance of measuring software size and demonstrating

978-1-7281-3421-5/19/$31.00 ©2019 IEEE
DOI 10.1109/SEAA.2019.00041

the success of incorporating functional size in effort estimation in agile software development projects. Some representative recent studies reporting the success of COSMIC- size based effort estimations are as follows: in 2014, Ungan et al.[12], based on a case study, found that COSMIC-based measurements provide more accurate results than the story point- based estimations. In 2016, Commeyne et al. [2] in another case study, found that the effort estimation model built with COSMIC Function Point (CFP) was better than the model built with Story Points (SP); the Mean Magnitude Relative Error (MMRE) value of SP was more than the MMRE value of CFP. In 2017, Salmanoglu, Hacaloglu and Demirors [13] presented the results of three independent case studies comparing the effort estimation models created with CFP and story points in agile projects, and reported that models based on CFP yield more successful estimation results.

Despite the presence of a handful studies demonstrating the success of functional size based effort estimations in real-life settings, agile practitioners have some critics that converge on the views of the nature of requirement documents that do not comprise essential content for FSM [11], and the fact that of agile practitioners tend not to easily give up practices that are already in use [5].

The applicability of FSM methods has been assessed in non-agile projects in several studies from various dimensions. For example, Turetken et al. [14] conducted a multiple case study on the impact of individual assumptions of measurers while conducting FSM on a Software Requirements Specification (SRS) document containing requirements, UML use case diagrams and detailed descriptions, activity diagrams, and simple class diagrams for data model. The authors emphasized the fact that FSM methods have well-written manuals and guidelines. Nonetheless, it is still possible for individual measurers to have subjective judgments when they face unfamiliar projects. According to the authors, for COSMIC FSM, identification of object of interests (OOI) and their data groups were difficult for the measurers. Turetken et al. [14] pointed out that the data model given was not detailed enough to determine OOI, and concluded that the requirements need to be written in a well-defined manner for COSMIC FSM. Another problem that is stated was the determination of functional processes when there is entity abstraction. Authors [14] suggested that there should be clear guidance in the measurement manual for this kind of cases.

Top et al. [15] further investigated the role of the measurer on the accuracy of FSM results. The authors concluded that accuracy of measurement is impacted by the measurers' individual cognition of measurement practice and the clarity of guideline. Ignorance of sub-type entities and transient OOI were among the other encountered problems where cascade delete problem was also observed.

Later, Trudel and Abran [16] assessed the usability of COSMIC FSM with inexperienced measurers, and used the COSMIC FSM to assess the quality of the SRS document, on which the measurements are done. The important thing is that SRS document used in the experiment, was prepared in detail with the introduction, scope, purpose as well as high-level description of software, list of features and functions, user characteristics, user interfaces with prototypes, functional requirements and many more information guiding the measurer as to what to measure. As a result of this study, the authors concluded that different measurers found different results and had difficulties on the correct identification of functional processes, data movement, data groups; missing data groups, data movement and duplicate data movement.

In 2011, Robiolo [17] investigated the simplicity of three measurement techniques including FPA, COSMIC FSM and Path from the IT Practitioners' perspectives, and found that Path is the easiest one in terms of measurement.

In the literature, there exist few studies assessing the applicability of FSM in agile software development projects. For example, Desharnais, Buglione and Kocaturk [18] assessed the usage of COSMIC FSM method in agile software project planning, where the applicability of the method is demonstrated using a fictional example. The authors addressed the evaluation of the quality of requirements document by using the checklist provided within the scope of the guideline for assuring the accuracy of measurements [19] in order to assess the accuracy of the result. Perhaps, the closest study is done by Desharnais, Kocaturk and Abran [20], where the authors focused solely on the quality assessment of documented user stories similar to the previous study with COSMIC FSM using the checklist in [19]. The authors rated the quality of a document having eight user stories in changing detail levels in three iterations.

To contribute to the literature, we explore the applicability of COSMIC FSM in six real projects from three different organizations. Each has a different requirement documentation style; which can be useful in exhibiting whether the approach of agile practices can indeed conform to FSM method.

III. Reasearch Methodology

Exploratory case study research method is used in the present work. The details related to the case study including the design, case descriptions and case realizations are presented in the following sub-sections.

A. Case Study Design

In the present paper, multiple case studies are conducted on different projects from different organizations, as stated earlier, to explore and assess the usability and suitability of FSM practices in agile software development projects. COSMIC FSM has been adopted as a measurement method for our purpose, and we aim to answer the following research questions:

RQ1.What types of problems are encountered during the application of COSMIC FSM process in agile projects?

RQ2. What measurement components are remained unidentified, thus affecting the accuracy of measurement?

Case Selection Criteria: The main criterion to include an organization or project in the study is whether software development is performed with agile methods. Furthermore, we aimed to choose organizations operating on different business domains or having projects in different business domains provided that the measurer is familiar with domain concepts, and there are different type of projects such as business applications, distributed systems etc., and application types such as desktop, mobile and web. We also looked for the availability of requirements specifications in the form of user stories, use-case scenarios, etc. The aim was to select cases having different requirement documenting style. Another initial criterion was the agility level of organizations using an agility assessment model such as the one proposed by Ozcan-Top and Demirors [21]; but since we found that it would not be feasible to conduct an agility assessment in every organization, we postponed the application of this criterion for future work.

Data Collection Procedure: The measurer aims to collect and record the following data within the scope of the case studies: Type of the application (web, mobile, desktop), business domain of the project, actual effort spent, agile methods used in the project, content types of requirement items documented (such as user story, use case, non-functional task, bug, etc.), functional user requirements, functional size of the project and a list of difficulties faced during the measurement.

Measurement planning: The size of the projects is planned to be measured by an external measurer who has never worked on any such projects before. In this way, the measurer will have the opportunity of being equidistant to each project and have an objective and similar kind of assumption regarding all the projects. COSMIC FSM is an ISO-compliant method that can be applicable during any stage of a project; independent of the technical details and known as a hundred percent repeatable method [3]. The measurer has 1 year of experience in measuring and, prior to any measurement within the present case study, the measurer has examined the related guidelines, including COSMIC Measurement Manual [22], Guideline for Sizing Business Application Software [23] and Guideline for Agile Projects [3], along with case studies developed by the COSMIC Group such as "C-Reg" Course Registration System [24] and ACME Car Hiring [25].

B. Description of Cases

To perform the case studies, we communicated with three different software development organizations which will be denoted as "*Organization 1", "Organization* 2", and "Organization 3" throughout the text.

Organization 1: Organization 1 develops web and mobile applications. They use Scrum and KANBAN methodology simultaneously. Especially, when there is not a sufficient number of requirements to define an epic, in order to acquire requirements in a fast manner, they benefit from KANBAN. The software development team is composed of five individuals. The team prefers and pays attention to document the requirements in the form of user stories. It also conducts meetings before sprints, at the beginning of which there are only user stories. During the meeting, they write technical details and assign estimations. In the past, these estimations were based on ideal days, whereas today they take the form of story points (SP). Team often prefer to document pessimistic estimations. The team is a self-organizing entity and adopt an experimental approach by preferring to explore different perspectives in order to find the best alternative. For example, earlier they used to conduct two daily meetings; one in the morning and the other one in the evening. These morning-evening daily meetings would often last 10 and 15 minutes, respectively.

Nowadays, evening meetings are cancelled as the same topics were repeated and discussed as the morning meeting. They found morning meetings more valuable. Also, in the past, they were estimating the user stories based on duration with ideal days. Nowadays, they prefer to experience estimation with SP. While assigning SPs the team determine a criterion such as "In one day, a person can complete a maximum of 3 SPs". One of the team members explained the reason for transition from ideal days to SP as follows: "It is difficult to produce an accurate estimate such as '8 hours'. This can create fear among the developers". First, the team estimate with SPs and then, translates it into hours. All these experimental decisions are made as a result of the three-week retrospective meetings, in which they discuss what they planned before and what they have realized. They adopt test-driven development. We collected 4 completed projects and measured them with COSMIC FSM. The team uses an issue-tracking tool to manage user stories and enter the actual effort spent on them. After each story is completed, the actual effort spent to complete it is entered into the issue-tracking tool. The actual efforts cover the implementation and testing activities related to each story. However, there exist independent social and technical tasks, to which an effort value is also assigned and which remain out of the scope of this study. On the other hand, there some user stories, whose effort values are not entered. The projects and the efforts spent to complete them are given in Table 1.

TABLE I. PROJECT DESCRIPTION OF ORGANIZATION 1

Project	***Platform***	***Domain***	***Status***	***Actual Effort in person-hour***	***CFP***	***Team' s estimation***
P1	Web	Online retailer	Completed	386,5	309	Estimate based on duration
P2	Web	Online retailer update[a]	Completed	241,3	115	Estimate based on duration
P3	Web	B2B E-commerce	Completed	363,5	207	No estimate entered
P4	Mobile	Mobile retailer	Completed	71,7	154	Estimate based on duration

[a] The update project has been measured independent of the previous version. It was also defined as a new project in the issue tracking tool.

Organization 2: Organization 2 is an information technology and consulting company. In their projects, they adopt agile software development with Scrum. The team works based on two-week sprints and conducts both sprint planning meetings and daily meetings. Sprint planning meetings are done before the sprints and last approximately three hours. Daily meetings last 15-30 minutes, and care is taken not to exceed this limit. The stories are prepared as one-sentence descriptions before each sprint. First, they write epics; then, they divide these epics into stories, and later, into tasks during the sprints. Every interaction in the system is written in an atomic way similar to a test scenario. Also, acceptance criteria, pre-post condition, normal flow and mock-up screens are available. The team writes stories considering the activity flow and screens related to them. The organization uses a tool for team collaboration. Mock-up screens for prototypes are prepared and for effort estimation in the company, story points are adopted. They do not prefer to assign more than 21 SP to each task because. The team considers a user story having an SP that is more than 21 as a complicated story which should be divided into manageable portions. One of the team members stated that there have been complaints about the relativity of the story point approach; where a typical story point can sometimes correspond to two days or half a day. It is also stated that there is no one-to-one relationship between the story point and the effort. The same member also complained about the effort never been predicted correctly.

We chose two completed sprints about a software development project through which firms will be able to pursue the processes of their applications for an official dealing over web. The project is a business web application. For database operations, the company uses an open-source database management system. The sprint details and the effort spent for each sprint is given in Table 2.

TABLE II. SPRINT DESCRIPTION OF ORGANIZATION 2

Sprint	***Platform***	***Domain***	***Status***	***Actual Effort in person-hour***	***CFP***	***Team' s estimation***
S1	Web	Official Dealing application	Completed	27	89	Estimate based on story points
S2	Web	Official Dealing application	Completed	45	225	Estimate based on story points

Organization 3: A single project was collected from an organization in Australia related to a business web application in a finance domain. There exists a brief explanation about the project including its major features. In addition, the team wrote use cases in a fully-dressed format describing the actor, use case description, main flow as step-by-step interaction of users with the system, post-conditions as output and exceptions. In the documentation, the prerequisite chain between the use cases is also provided. The team supported the use cases with screens. It kept the project details in an online collaborative working tool. Presently, the project is ongoing, From the organization, we collected the actual effort spent for each user story. The completed part of the project lasted 225 person-hours. The effort distribution of use cases is given in Table 3.

TABLE III. USE CASE AND EFFORT DISTRIBUTION OF ORGANIZATION 3

Use Case	***Status***	***CFP***	***Actual Effort in person-hour***
Use case 1	Completed	3	40
Use case 2	Completed	5	20
Use case 3	Completed	14	45
Use case 4	Not completed	10	0
Use case 5	80% completed	24	50
Use case 6	Completed	9	20
Use case 7	80% completed	18	15
Use case 8	Completed	13	0 (included in Use case 6)
Use case 9	Completed	18	0 (included in Use case 7)
Use case 10	Not completed	10	0
Use case 11	Completed	9	35
Use case 12	Not completed yet	6	0

C. *Realization of Functional Size Measurements*

To perform the case studies, ISO-compliant COSMIC FSM method [22], whose success has been proven in effort estimation, is chosen to be applied in projects collected from Agile organizations.

This COSMIC FSM method is based on the identification of data movements: for this aim, the method suggests to identify functional user requirements (FUR), the functional processes (FP) in these FUR and the data movements (DM) in the FP [22]. DM is defined as a flow of a data group (a group of attributes), describing a single object of interest (OOI) [23]. In other words, for a movement to be labeled as a data movement, it needs to be associated with a single OOI and be convey data related to that OOI and DM can appear in four type of forms, namely Entry (E), Exit (X), Read (R) and Write (W) [22].

Typical COSMIC FSM process that is described in the measurement manual [22] involves following steps shown in Figure 1:

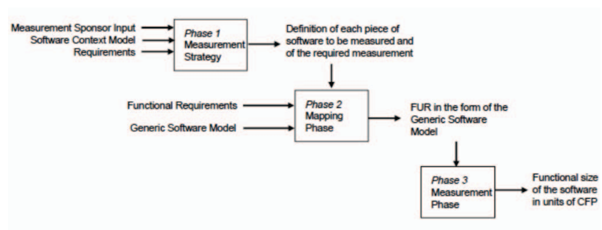

Fig. 1. COSMIC FSM Phases [22].

As a first step of the measurement, functional user requirements are identified in the projects. The dataset collected was made up of user stories, use cases, bugs, non-functional requirements, technical implementation tasks, business rules, social tasks (such as meetings, working a particular technique, etc.).

IV. RESULTS AND DISCUSSION

In this part, we present the results of the case studies by mainly describing the applicability of COSMIC FSM in different agile software development projects. After determining the parameters in measurement strategy phase, a typical COSMIC FSM mapping process involves the mapping of FUR into, Functional Processes, Object of Interests and Data Groups. This mapping stage is followed by the measurement step with the identification of respective data movements [3]. We describe the COSMIC FSM application process within the scope of distinguishing these components and the characteristics of the requirement specification style of the agile teams affecting this identification.

A. Assessment of the availability of project descriptions

Except the project carried out by of Organization 3, in none of the projects observed, is a description of the intended system provided. Also, there is no context diagram describing the interaction of the system with their functional users. This issue complicates the comprehensibility of the system under measurement and other systems that interact with it at first glance. In other words, the boundaries of the system are not apparent to the measurer.

B. Assessment of the data analysis process

There is no data model in any of projects. This fact complicates the identification of object of interests (OOI), which is a critical concept in COSMIC FSM. The reason is that data movement which is the base functional component is defined as a movement conveying single data group describing a single OOI [22]. Therefore, to count a data movement, the measurer needs to know the data groups and related OOIs. To identify OOI, data analysis methods such as Relational Data Analysis (RDA), UML Class Diagram and Entity/Relationship (ER) Analysis methods are suggested and emphasized in the guideline [23]. Accordingly, entity-type in ER, class in UML and Third Normal Form (3NF) subject in RDA models are usually prospective OOIs and, therefore, useful to identifying persistent data groups [23]. When we look from the perspective that data modeling has not been created in any of the cases observed, such an effort for measurement purposes only may be perceived as an overhead and might not be adopted among agile teams. In addition, this kind of models, for example 3NF subject creation in RDA, are concepts belonging to the relational database models. Today, technological enhancements on cloud computing and systems involving big data analytics have caused the emergence of non-relational database models such as NoSQL databases [26]. Hence, relational models are being replaced, or used in conjunction with non-relational databases. Especially, Banerjee & Sarkar [26], based on a study by Cattell [27], reported that contemporary agile applications involve flexible schema and horizontal scalability for read/write operations, which may be dispersed on many servers. In addition, in [28], it is specified that non-relational models become more appropriate for agility to adapt the "quick schema iterations and frequent code pushes". Furthermore, Desmarets [29] points to an existing perception that data modeling is incompatible with agile, and argues that creating logical models which require normalization might not be a productive step for NoSQL databases. In addition, Hsieh [30] previously stated that the physical data model is now focused instead of logical data model by NoSQL practitioners.

In the sprints of Organization 2, data groups are provided as a list in the documentation different from other organizations observed.

Identification of OOI in agile projects can be discussed in four more perspectives: firstly, as stated in [23] the data modeling methods do not attempt to identify all the OOIs within the scope of measurement; especially, transient data groups cannot be identified via these methods.

Secondly, even though some of the data groups are known, since not all of them are given, the measurer faces difficulty in understanding which OOIs are involved. Consequently, it remains difficult to conduct the measurement and identify distinct OOIs due to the lack of data model. This fact is already mentioned by Hussain et al. [11] who pointed out that without a full list of data-groups and data models, requirements are not sufficient to identify data groups to conduct COSMIC FSM. In addition, data groups are mostly abstract and are in their initial states. In the view of the authors of the present paper such data groups can be improved through further iterations, still these improvements might not be reflected within the documentation.

Thirdly, the identification of data groups and OOIs has also been stated in the literature among the problems related to measurement, even in non-agile software development projects, which have comparatively more details in terms of requirements. See, for example, [14].

As a fourth issue, in the guideline [23], it is stated that entities having "many-to-many" (M:N) relationship are candidate OOIs. However, in some user stories of Project 1, it is observed that there is not enough information to understand the cardinality between entities and, in the same way, identifying entities participating in M: N relationships.

Compliant with our observations, other studies from the literature [14, 15] demonstrate that identification of the OOI concept is not an easy step in the measurement process and, to our view, which requires further improvement.

C. Assessment of the characteristics of requirement artifacts

In the projects of Organization 1, user stories are the only artifact that convey requirements. In them, the human users and their intentions are specified. The user stories are generally written in the following format:

"As a user, I should be able to do task X."

To be able to apply COSMIC FSM, we concentrated on the user stories. An observed situation in Organization 1 is the tendency of team members to write social tasks (e.g., having a meeting with the customer) and technical development tasks (e.g., making a connection to system XX) in the user story format, too. In other words, not only the requirements but also all the tasks related with the creation of software are written in the user story format. In addition, the team members estimate these other tasks in the same manner with the functional requirements. Therefore, one observation related to estimation is that the estimated effort is not only derived from the functionality of the project. Any effort requiring any task is estimated with story points, and this can be the reason behind its high adoption. The requirement artifacts of Organization 2 were also user stories, but these stories are supported with mock-up screens.

In the projects of Organization 3, the organization documented their requirements in a fully dressed use-case format. The steps to accomplish the use cases are documented as step-by-step main flow, along with the actor, prerequisites, post-conditions and exceptions.

D. Ease of mapping the user story into single functional process

To map a user story into functional process, there exist two issues to consider:

First, according to the COSMIC Measurement Manual [22], a requirement has the necessary level of granularity for accurate size measurement if its functional processes and their sub-processes are identifiable. Second, it is further suggested in the COSMIC Guideline for Agile Projects [3] that "a user story should define a single functional process (p.18)" and, when it is written in the form of a Message Sequence Diagram, measuring its COSMIC size is trivial. However, in the projects of Organization 1, it is seen that there exists a set of requirements that is not written in this described level of granularity. Instead, there exists varying levels of abstractions in the requirements specifications. Some of the encountered user stories are either too big (comprising more than one FP) too abstract to identify individual FPs or too low level to describe a step in the requirement. For these type of abstractions, neither the individual functional processes nor their data movements are identifiable.

Another common problem observed in the projects measured in Organization 1 is the lack of sub-processes of FPs explaining the required details to understand the FPs data movements. In addition, in these type of writing style, the data groups describing the OOI and distinct OOI are not easy to identify. Therefore, when only user stories are considered, the difficulty of applying FSM increases. However, in projects of Organization 1, some stories have sub-tasks related to physical implementation, unit testing information facilitating the comprehensibility of the data groups and OOI, especially naming the implementation classes and their data attributes. For cases similar to Organization 1, a CRUDL format of requirement specification can be used while documenting the requirements. This can also facilitate the comprehensibility of the requirements; for example, to lay bare the different capabilities of functional users, such as different authorization levels, etc. This problem has been overcome to some extent with availability of mock-up screens in Organization 2. The use of these screens facilitated the comprehensibility of the sub-process identification, enhanced the understanding of the data groups involved in the FP through the screen. In the mockups, briefly written stories are supported with input labels, text fields which facilitate the understanding of data groups entering to and exiting from functional processes where navigation labels ease the identification of individual functional processes within the story.

Similarly, in the project of Organization 3, the data groups movements are made more visible with the availability of mockup screens where sub-processes are best understood compared to the other projects, due to the fully-dressed use case format adoption in this project.

Incorporation of mock-ups into the requirements, increased the comprehensibility of processes and facilitated the measurement process. Not only the pre-implementation, but also post-implementation artifacts can be used to extract functional user requirements [22]. This fact related to mock-ups led us to comparing the pre-implementation FSM results with those of post-implementation measurement. For this aim, we studied Project 4's delivered system, which is available in AppStore[1]. We downloaded the application and assessed each screen for the availability of the functional processes. Then, these processes were measured with COSMIC FSM. The implemented application has an 89 CFP score where its pre-implemented version has 154 CFP. We can explain that the reasons behind the decrease of the CFP from pre-implemented to post-implemented system can be either or both of the followings:

In the requirements set of Project 4, there are functional users other than customers. However, in the implemented version we could only access the customer roles in this mobile application. Therefore, we could only measure customer-related functional processes. In the pre-implementation version, only 128 out of 154 CFP belongs to the functional size from the customer perspective. Furthermore, in the pre-implementation requirement set of the project, there were two stories related to the customer whose actual efforts were not entered. This may be an indicator of the possibility that some functionality has not been implemented.

One observed benefit of measuring post-implementation artifacts can be the visibility of error/confirmation messages, which may not be stated in the pre-implementation requirement set, thus likely to be ignored by the measurers. Another problem encountered in the projects of Organization 1 was related with the system-type functional users. From the requirements document, the measurer can understand that Organization 1 is interacting with another software, from which it receives services. In some stories, whether the data is saved in persistent storage or sent to this software is not clearly specified. This issue creates ambiguity in the correct identification of the type as well as the number of data movements.

V. VALIDITY THREATS

One of the validity threats can be related with the interpretation of the data collected. Requirement artifacts that are in the form of user stories, use cases, and mockups are collected and taken as input for FSM. To overcome the interpretation variance, the measurer contacted project representatives to gain more in depth insight. In the end, certain artefacts were eliminated from the measurements.

Another threat to validity can be the sampling of organizations. An accurate sampling should be done to increase the results' generalizability. The authors performed the case studies in projects having different characteristics and collected from different organizations using different styles of documentation.

To ensure reliability, the measurement is done in accordance with the COSMIC FSM method, and various measurement supporting guidelines stated in section 3 have been investigated and referred to. In this way, if later another researcher replicates the present research, then he or she will be able to apply the same procedures and, hence, obtain similar results.

[1] https://www.apple.com/tr/ios/app-store/

VI. Conclusion and Future Works

In our previous study [9], we explored the usage of size in agile projects. One of the main findings of the study was the negative perception of agile practitioners' regarding the incorporating FSM in agile software development. Especially, the difficulty in the adjustment of FUR for measurement has been stated as a demotivating factor to use FSM.

Motivated by this finding, the case studies examined in the present paper determine the difficulties encountered while an independent measurer evaluates FSM in different real agile software development project. As a general finding, during the case studies, it is seen that COSMIC FSM in agile requires a lot of assumptions. Also, in all cases some of the major components of COSMIC FSM cannot be precisely determined from the available contents and usually there is no description of the context and it is difficult to understand the relationships among the requirements. In addition, there was no data model in the projects and not all the data groups were specified. In some cases, the identification of functional processes was problematic; more specifically, this was the case with the quantity and the type of data movements. The need for data modeling, which is seen less important in some of today's emerging approaches, is an important issue to consider for further studies.

On the other hand, the supporting of requirements with mockup screens and detailed specifications steps facilitated the measurement. We can summarize our findings as suggestions for requirements practice for organizations that wish to measure COSMIC FSM in Agile settings to describe step-by-step interactions between the users and systems, identifying data groups in details, and providing data models to extract OOIs.

It is also an important finding of this study that teams spend a lot of time for daily, weekly and retrospective meetings. If and how objective measures, such as COSMIC FSM change durations should be studied further. In addition, considering that every organization interpret the agile philosophy in their own way and has different agility levels, we need to conduct further studies to understand fitness of best agile practices and objective measures.

Some of these findings have similarities with these findings in traditional Waterfall based projects. Further evidence is needed to more precisely separate problems unique to agile approaches. The authors plan to extend the present study to understand the usability of measurement results for different purposes, such as effort and quality prediction.

References

[1] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development: a systematic literature review," Proceedings of the 10th International Conference on Predictive Models in Software Engineering, 2014, pp. 82–91.

[2] C. Commeyne, A. Abran, and R. Djouab, "Effort Estimation with Story Points and COSMIC Function Points-An Industry Case Study," Software Measurement News, vol. 21, no.1, pp.25-36, 2016.

[3] E. Berardi, L. Buglione, J. Cuadrado-Collego, J.M. Desharnais, C.Gencel, A. Lesterhuis, L. Santillo, C. Symons, S.Trudel, "Guideline for the use of COSMIC FSM to manage agile projects," Common Softw. Meas. Int. Consort., 2011.

[4] A. Abran, J.-M. Desharnais, M. Zarour, and O. Demirors, "Productivity-based software estimation models and process improvement: an empirical study," Int. J. Adv. Softw, vol. 8, no. 1&2, pp. 103–114, 2015.

[5] L. Buglione and S. Trudel, "Guideline for sizing agile projects with COSMIC," Proc. IWSM/MetriKon/Mensura, Stuttgart, Ger., 2010.

[6] B. Ozkan, O. Turetken, and O. Demirors, "Software functional size: For cost estimation and more," in European Conference on Software Process Improvement, 2008, pp. 59–69.

[7] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," in WET ICE 2003 Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003., 2003, pp. 308–313.

[8] A. De Lucia and A. Qusef, "Requirements engineering in agile software development," J. Emerg. Technol. web Intell., vol. 2, no. 3, pp. 212–220, 2010.

[9] T. Hacaloğlu and O. Demirörs, "Challenges of using software size in agile software development: A systematic literature review," IWSM/Mensura'18, September 18–20, 2018, Beijing, China, 2018.

[10] M. Salmanoğlu, K. Öztürk, S. Bağrıyanık, E. Ungan, and O. Demirörs, "Benefits and challenges of measuring software size: early results in a large organization," in 25th International Workshop on Software Measurement and 10th International Conference on Software Process and Product Measurement, IWSM-Mensura, 2015.

[11] I. Hussain, L. Kosseim, and O. Ormandjieva, "Approximation of COSMIC functional size to support early effort estimation in Agile," Data Knowl. Eng., vol. 85, pp. 2–14, 2013.

[12] E. Ungan, N. Cizmeli, and O. Demirors, "Comparison of functional size based estimation and story points, based on effort estimation effectiveness in SCRUM projects," in Proceedings - 40th Euromicro Conference Series on Software Engineering and Advanced Applications, SEAA 2014, 2014.

[13] M. Salmanoglu, T. Hacaloglu, O. Demirors, and C. C. S. Concepts, "Effort Estimation for Agile Software Development : Comparative Case Studies Using COSMIC Functional Size Measurement and Story Points," ACM Int. Conf. Proceeding Ser., 2017.

[14] O. Turetken, O. O. Top, B. Ozkan, and O. Demirors, "The impact of individual assumptions on functional size measurement," in Software Process and Product Measurement, Springer, 2008, pp. 155–169.

[15] O. O. Top, O. Demirors, and B. Ozkan, "Reliability of COSMIC functional size measurement results: A multiple case study on industry cases," in 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, 2009, pp. 327–334.

[16] S. Trudel and A. Abran, "Functional size measurement quality challenges for inexperienced measurers," in International Workshop on Software Measurement, 2009, pp. 157–169.

[17] G. Robiolo, "How simple is it to measure software size and complexity for an it practitioner?," in 2011 International Symposium on Empirical Software Engineering and Measurement, 2011, pp. 40–48.

[18] J. Desharnais, L. Buglione, and B. Kocaturk, "Improving Agile Software Projects Planning Using the COSMIC Method," in workshop on Managing Client Value Creation Process in Agile Projects (Torre Cane, Italy, 2011.

[19] COSMIC, "Guideline for Assuring the Accuracy of Measurements, v 0.92." Common Software Measurement International Consortium, 2011.

[20] J.-M. Desharnais, B. Kocaturk, and A. Abran, “Using the cosmic method to evaluate the quality of the documentation of agile user stories,” in 2011 Joint Conference of the 21st International Workshop on Software Measurement and the 6th International Conference on Software Process and Product Measurement, 2011, pp. 269–272.

[21] Ö. Özcan-Top and O. Demirors, “Application of a software agility assessment model–AgilityMod in the field,” Comput. Stand. Interfaces, vol. 62, pp. 1–16, 2019.

[22] COSMIC, “The COSMIC Functional Size Measurement Method Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2017) Version 4.0.2,” Common Software Measurement International Consortium, 2017.

[23] A. Lesterhuis and C. Symons, “Guideline for sizing business applications software version 1.3,” 2017.

[24] COSMIC, “Course Registration (‘C-REG’) System Case Study Version 2.0,” Common Software Measurement International Consortium, 2015.

[25] COSMIC, “ACME Car Hire Case Study v1.0.1,” Common Software Measurement International Consortium, 2018.

[26] S. Banerjee and A. Sarkar, “Modeling NoSQL databases: from conceptual to logical level design,” in 3rd International Conference Applications and Innovations in Mobile Computing (AIMoC 2016), Kolkata, India, February, 2016, pp. 10–12.

[27] R. Cattell, “Scalable SQL and NoSQL Data Stores,” ACM SIGMOD Rec., vol. 39, no. 4, pp. 12–27, 2010.

[28] “NoSQL Databases Explained.” https://www.mongodb.com/nosql-explained. (Accessed: Mar. 30, 2019).

[29] P. Desmarets, “Data Modeling is Dead…Long Live Schema Design!,” 2018. https://www.dataversity.net/data-modeling-dead-long-live-schema-design/#. (Accessed: Mar. 30, 2019).

[30] D. Hsieh, “NoSQL Data Modeling,” 2014. https://www.ebayinc.com/stories/blogs/tech/nosql-data-modeling/. (Accessed: Mar. 30, 2019).