A Tool for Automating Sizing in Agile Development Using the COSMIC Method

Bruel Gérançon, Sylvie Trudel
Department of Computer Science
Université du Québec à Montréal (UQAM)
Montréal, Canada
e-mail: gerancon.bruel@uqam.ca, trudel.s@uqam.ca

***Abstract*— Agility is one of the industry's most widely used software development approaches. It lies in the fact that an agile development project is supposed to deliver the functionalities for the product owner as soon as possible. However, automating sizing in agile development remains difficult. The software's functional size measurement methods are challenging to scale for agility. In the industry, managers and scrum masters use empirical methods to estimate the size of user stories manually. One of the pitfalls of this approach is the limited collection of data in agile projects, which makes it challenging to carry out statistical analysis to better estimate the value of appropriate efforts for the subsequent iterations of the projects. This paper presents a tool for automating sizing in agile development using the COSMIC Function Point (CFP) from User Stories written in natural language. Our tool integrates a set of techniques in Natural Language Processing (NLP), which semantically identify the triplet (subjects, predicates or verbs, objects) from items of product backlog (User Stories written in natural language) and automatically quantifies the number of verbs (data movement) which refers to the functional size. Afterward, we applied a set of rules in COSMIC to identify the types of data movement.**

Keywords - Agile; User Stories; Triplet; Natural Language Processing (NLP); Automation.

I. INTRODUCTION

The measure of the functional size of agile development projects plays an important role in software engineering. It allows project managers to establish reliable estimation and productivity models [11][18]. In other words, it is a key factor that allows for estimating the effort, the cost of developing software products, and performing an analysis of the performance of the software development team [9]. Moreover, the techniques used in agile development for writing software specification documents do not facilitate the automation of the functional size of agile development [9]. The software requirements are written in natural language and do not contain technical and specified details [10]-[12]. For this reason, it would be important to propose a new approach that facilitates measuring the functional size of agile development. How could the triplets approach help automate the functional size of agile projects?

In this article, we will review the primary technique used for writing software requirements in agile development in Section II. Section III will focus on the estimation technique in agile development. Subsequently, we will present the limitations of this technique and evaluate the possibility of automating agile developments using COSMIC. Section V will describe our proposed triplets approach for automating sizing in agile development using the COSMIC Function Point (CFP) from User Stories written in natural language and previous automation work realized with the triplets structure. Section V will introduce our new tool that automatically measures the functional size of agile development projects using the COSMIC method. Finally, we will present in Section VI the results of our research and its limitations.

II. TECHNIQUE FOR WRITING REQUIREMENTS IN AGILE

In agile development projects, the requirements are often written as structured User Stories. A User Story consists of a few lines of text describing a functionality software must offer to allow an actor or user to achieve a specific objective [4]. One of the significant advantages of this technique is that it is centered on the system user [3][4]. An often-used User Story description format is the following:

As a "user role"
I want "this feature or functionality"
So that I can “benefit or business value, or business reason”

III. ESTIMATION IN AGILE DEVELOPMENT PROJECTS

In this section, we present the user story points technique for estimating agile development projects, the limitations of this technique, and the sizing of User Stories with the COSMIC method.

A. *User Story Points*

In agile development, User Story Points (USP) are considered an estimated relative level of the effort required to complete a User Story [1][3]. Estimation is important because it allows the project manager to identify which requirements to prioritize for each iteration and whether these requirements or User Stories could be completed during the iteration [7]. Automating the measurement of the functional size of agile projects is a priority for managers and agile teams. Most agile development projects measure their requirements in user story points [2]. As part of this measure, agile teams commonly use the Fibonacci sequence to size stories (1, 2, 3, 5, 8, 13) to assign a value combining size and complexity so that this value reflects the effort to achieve the product backlog item [1][2][13][14]. The development team considers the number of User Story Points as the average size.

B. *Limitations of User Story Points*

The limitations of User Story Points for measuring agile development projects are that it is not possible to standardize their value from one project to another or from one organization to another, as this value is subjective and specific to the development team that assigned it [2][6][7]. Also, the User Story Points do not represent a measure of functional size but rather an effort estimate [1][3][8].

C. *Sizing User Stories with the COSMIC Method*

Many published works demonstrate that it is possible to manually apply a functional size measure on the items in the product backlog. For example, Trudel and Buglione [8] proposed a Guideline for sizing Agile Projects. Desharnais and al. [19] used functional size methods, such as COSMIC method, to estimate Agile User Stories. Angara et al. [6] present related work on linkages between User Stories and the COSMIC method. Furthermore, from the literature consulted, only a few tools automatically allow measuring the functional size of agile development projects (items in the product backlog) using the COSMIC method. Therefore, this paper aims to describe a tool for automating sizing in agile development projects using the COSMIC method.

IV. THE TRIPLETS APPROACH AND PREVIOUS WORK

In this section, we present the triplet approach proposed for estimating agile development projects and the previous work of automation of functional size with this approach.

A. *The Triplets Approach*

The triplets approach is a model that defines and represents the software requirements as a triplet [9][15]. Each triplet comprises a trio of concepts, such as (subject, predicate, object). The subject represents the functional user interacting with the system; a composite predicate represents the use case scenario; an atomic predicate represents the events the functional user triggers. The object represents a software component [9][10][12].

B. *Previous Works*

In our previous work, we used the triplet approach to automate functional size measurement of use cases [9]. We developed a tool for automatically generating triplets from use cases written in natural language, specifically in English or French, and calculating the functional size of the software to be measured [9]. Indeed, we decided to adapt our automation tool from requirements written as structured User Stories to measure agile development projects.

V. TOOL TO AUTOMATE AGILE PROJECT SIZING

Section II indicates that product backlog requirements are written in natural language as User Stories. We developed a tool that automatically performs the functional size measurement on the User Stories of a product backlog. Our tool integrates a set of Natural Language Processing (NLP) techniques, which semantically identifies the subjects, verbs, and objects from User Stories written in natural language. In such a perspective, we presume that a software requirement written as a User Story refers to an actor (subject) that triggers an action or a system operation (verb or data movement) on an object. Afterward, we applied several rules in COSMIC for identifying the verbs that correspond to a type of data movement (Entry, Exit, Read, Write) and quantifying the number of verbs (data movement), which refers to the functional size [9]. In the description of this User Story: “As a visitor, I want to search a product,” the tool identifies “user” as the subject, “search” as the verb, and “product” as the object. In other words, the tool targets the triplet structure (subject, predicate/verb, object). In [9] and [10], we assumed that the writing of software requirements in agile, specifically in the form of User Stories, can be done with predicates of two arguments ***f* (*x*, *y*).** The predicate is expressed by a verb, which corresponds to the data movements, system operations, or methods of the object, which will be triggered following an external stimulation [9][10][15]. The “**x**” variable or subject of the action represents the user or actor, while the “**y**” variable is the object of the action. The objects represent the software classes that will be implemented. In this case, our tool automatically identifies the potential software components and methods (data movements or system operations) that will be implemented in each iteration of an agile development project. Figure 1 presents the data model built by our tool from the items in the product backlog to determine the functional size in COSMIC Function Points (CFP).

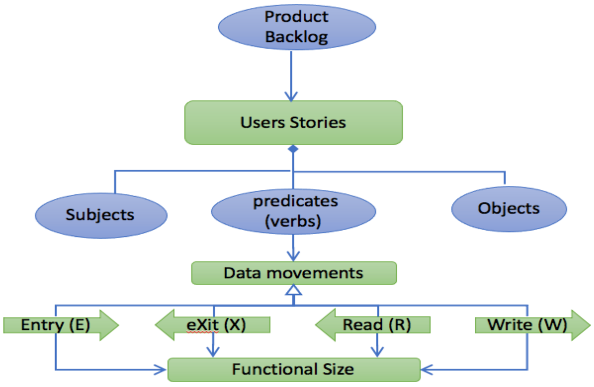

Figure 1. Data Model built by our tool.

A. Evaluation and Validation of Results

We tested the tool with two (2) agile development projects for which the items in the product backlog are written in the form of User Stories. First, human experts certified with the COSMIC method manually measured the functional size of the two (2) projects according to the measurement manual rules [16][17]. Second, we compared the results presented by the tool to the experts' manual measurement results, which are published and available on the COSMIC website. The research showed that our tool offers automated results consistent with the manual results, with an average accuracy of 95.97%. It was found that automated counting yields different results compared with manual counting (a difference of 4.03%). To identify the source of the discrepancies, we examined the software requirements documents (description of the User Stories) for both projects. After analysis, we identified the following main factors behind the discrepancies:

- The tool fails to determine data movements for the following Use Stories: “As a user, C-Reg requests Course Catalog to send Course Offering data”; “As a user, C-Reg requests Course Offering data (with number of students enrolled, etc.) from the Course Catalog.” This is because the tool identifies the verb “requests” as a noun, not a verb. The tool fails to determine data movements for these User Stories: “As a user, C-Reg sends The Professor’s selected Course offerings to the Catalog,” and “C-Reg sends Professor’s qualifications and department to Course Catalog to retrieve.” The tool does not identify the verb “sends.” This project was challenging to measure since there were a lot of unnecessary details in the description of User Stories while the User Story description format was not wholly respected. User Stories in this document are also described with passive verbs, but some User Stories are described according to the recommended standard format with active verbs. The two examples of scenarios of User Stories whose verbs are conjugated in the passive form are respectively, “Validated Course Offering IDs are sent to the Course Catalog so that it can maintain the count of Students for each Course Offering”; “Student’s Schedule items are marked ‘enrolled’ and made persistent on C-Reg’’.
- For project #2, a difference of 1 CFP was recorded between manual and automated measures for the total number of data movements (|83 CFP - 84 CFP|). The additional COSMIC Function Point (CFP) obtained by the tool occurs following a duplicate. We summarized in TABLE I the automatic size of User Stories obtained from the tool.

TABLE I. AUTOMATIC AND MANUAL SIZE OF USER STORIES

Project	Manual Functional Sizing	Automatic Functional Sizing	Accuracy
Case#1	109	95	93.14%
Case#2	83	84	98.80%
Total	**193**	**178**	**95.97%**

B. The Usefulness of Functional Size Automation for Agile Development

One of the benefits of measuring the functional size of agile development projects is that it allows project managers to establish reliable estimation and productivity models, provided the effort and time data is of good quality [11][18]. Studies and experiments have shown a strong correlation between functional size and effort and between functional size and project duration [18]. When the functional size of agile projects is known early, it would allow managers to perform an analysis of the performance of the software development team, such as development cost, productivity, and delivery rate [18]. This is admittedly a weak point of most agile methods.

It is important to mention that automated counting of agile development projects is advantageous when there are many requirements from real-life projects. Indeed, automating the measurement process of agile development is helpful because it allows measuring faster. For example, in our experiment, for a series of projects totaling 362 CFP in size, a COSMIC human expert took about 268 minutes to measure the functional size, i.e., to apply the COSMIC method and determine the functional size [10]. Other human experts would have taken around 362 minutes, a little over one minute per CFP [12]. As for the tool developed, it took approximately two (2) minutes and 3 seconds to upload the document that contains the User Stories descriptions, determine the functional size, and identify the types of data movement [10] [15]. By comparing the manual measurement effort to that of automated measurement, we found a significant difference in favor of automated measurement [10].

VI. CONCLUSION AND FUTURE WORK

This paper proposed a new tool designed to automate the functional sizing of agile development projects from the items in a product backlog. This tool can effectively identify the subjects, predicates or verbs, and objects derived from User Stories and quantifies the number of data movements, which refers to the functional size. Thus, the validation of our tool needs to cover the potential cases. In the future, extensive testing will be performed to improve the tool's efficiency. Also, we will integrate a machine learning module, which allows the tool to learn to identify the data movement for the User Stories that are not described according to the recommended standard format.

REFERENCES

[1] S. Murat, T. Hacaloglu and O. Demirors, “Effort estimation for agile software development: Comparative case studies using COSMIC functional size measurement and story points”, *Proceedings of the 27th International Workshop on Software Measurement and the 12th International Conference on Software Process and Product Measurement*, pp. 41-49, 2017.

[2] T. Fehlmann and A. Gelli, “Functional Size Measurement in Agile Development: Velocity in Agile Sprints”, *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, pp. 200-204, 2023.

ISBN: 978-1-68558-089-6

[3] I. K. Raharjana, D. Siahaan, and C. Fatichah, “User stories and natural language processing: A systematic literature review”, pp. 53811-53826, *IEEE*, 2021.

[4] K. Beck and D. West, “User Stories in Agile Development”, In Scenarios, Stories, Use Cases: Through the Systems Developments Lifecycle, 2004.

[5] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, “Agile software development methods: Review and analysis” *arXiv preprint arXiv:1709.08439*, 2017.

[6] J. Angara, P. Srinivas Prasad, and G. Sridevi, "Towards Benchmarking User Stories Estimation with COSMIC Function Points-A Case Example of Participant Observation" *International Journal of Electrical & Computer Engineering*, pp. 2088-870, 8, 2018.

[7] J-M. Desharnais, L. Buglione, and B. Kocatürk, “Using the COSMIC method to estimate Agile user stories”, *Proceedings of the 12th International Conference on product focused software development and process improvement*, pp. 68-73, 2011.

[8] S. Trudel and L. Buglione, “Guideline for sizing Agile projects with COSMIC” *Proceedings of International Workshop on Software Measurement*, 2010.

[9] B. Gérançon, S. Trudel, R. Nkambou, and S. Robert, “Software functional sizing automation from requirements written as triplets.” *International Conference on Software Engineering Advances, Barcelona, Spain, ISBN: 978-1-61208-894-5 16*, pp. 23–29, 2021.

[10] B. Gerançon, “Design and implementation of a triplet formalization technique for the automation of functional size measurement from software requirements written in natural language”, Ph.D. thesis, University of Quebec at Montreal (UQAM), 2022). [Online]. Available from: https://archipel.uqam.ca/15762/1/D4212.pdf, [Retrieved: October 2023].

[11] S. Trudel, “The COSMIC ISO 19761 functional size measurement method as a software requirements improvement mechanism”, Ph.D. thesis, École de Technologie Supérieure (ETS), Montreal, 2012.

[12] B. Gérançon and S. Trudel, “Improving Quality of Software Requirements by Using a Triplet Structure”, *International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, IEEE, Computer Society, ISBN-13: 979-8-3503-2028-2*, pp. 1884-1888, 2022.

[13] T. Hacaloglu and O. Demirors, “Measurability of functional size in Agile software projects: Multiple case studies with COSMIC FSM” *In 2019, 45th Euromicro conference on software engineering and advanced applications (SEAA)*, pp. 204-211. IEEE, 2019.

[14] R.K. Mallidi and S. Manmohan, “Study on agile story point estimation techniques and challenges” *International Journal of Computer*, pp. 9-14, 2021

[15] B. Gérançon, R. Nkambou, S. Trudel and S. Robert, “A cognitive model for supporting software functional sizing automation from requirements written as triplets”, *The 20th International Conference on Software Engineering Research Practice, Springer Nature, Las Vegas, USA, ISBN-13: 978-1-6654-6218-2 (IEEE)*, pp. 362-373, 2022.

[16] A. Abran *et al.*, “The COSMIC Functional Size Measurement Method Version 5.0”, [Online]. Available from: www.cosmic-sizing.org, [Retrieved: August 2023].

[17] M. Downing, M. Eagles, P. Hope, and Ph. James, “C-Reg Case Study v1.0.1”, August 2018. [Online]. Available from: https://cosmic-sizing.org/publications/acme-car-hire-case-study-v1-0-1/, [Retrieved: August 2023].

[18] A. Abran, “Software project estimation: The fundamentals for providing high-quality information to decision-makers”, John Wiley & Sons, 2015.

[19] J. Desharnais, L. Buglione, and B. Kocatürk, "Using the COSMIC method to estimate Agile user stories", *Proceedings of the 12th international conference on product focused software development and process improvement*, pp. 68-73, 2011.