COSMIC Sizing of RPA Software: A Case Study from a Proof of Concept Implementation in a Banking Organization

Francisco Valdés Souto[1][0000-0001-6736-0666] and Roberto Pedraza Coello[2] and Fabiola Cristina Olguín Barrón[3]

[1] National Autonomous University of Mexico
Science Faculty
CDMX, Mexico City, Mexico
fvaldes@ciencias.unam.mx
[2] National Autonomous University of Mexico
Research Institute in Applied Mathematics and Systems
rpedrazacoello@gmail.com
[3] Robot Digital TI
fabiola.olguin@robotdigital.mx

Abstract. During a software project lifecycle, the estimation of cost, effort, and schedule are essential aspects since they are used to allocate resources, plan product releases, and negotiate payments. It has been demonstrated that the dominant variable to estimate these aspects is the software functional size. One of the methods to measure software is COSMIC (ISO/IEC 19761).

The COSMIC organization has published guidelines to help the measurement process for some types of software applications like real-time software, SOA, business applications, etc. However, there is a lack of information on how to measure some new or trending technologies. One of these trending technologies is Robotic Process Automation, also known as RPA. Even though the RPA market has been growing rapidly, there has not been an approach of using formal metrics to measure this kind of software.

This paper illustrates the process of measuring an RPA software using the COSMIC method.

Keywords: Robot Process Automation, RPA, COSMIC, ISO/IEC - 19761

1 Introduction

Companies naturally are looking to be more efficient and save costs. This kind of situation causes technological innovations to arise. Robotic Process Automation (RPA) is one of these innovations which, in terms of return on investment, most case studies show positive results. Šimek & Šperka [1] claim several advantages in its application, generating the RPA market to grow and aiming to keep growing.

According to van der Aalst, Bichler & Heinzl [2], “RPA is an umbrella term for tools that operate on the user interface of the computer systems in the way a human would do”. Another definition is presented by Tornbohm & Dunie [3]: “RPA tools perform [if, then else] statements on structured data, typically using a combination of user interface interactions, or by connecting to APIs to drive client servers, mainframes or HTML code. An RPA tool operates by mapping a process in the RPA tool language for the software Robot to follow, with runtime allocated to execute the script by a control dashboard.” RPA tools generate software robots that perform tasks the same way a human does, with some configuration a robot could read emails, open PDFs file, enter data into ERP systems, etc.

RPA can be cataloged as a type of software implementing a type of functionality, maybe in a distinct way of traditional development. During the lifecycle of any software development project, including RPA, the use of formal metrics to estimate cost, effort, and schedule is relevant since these aspects are used to allocate resources, plan product releases and negotiate payments. However, currently in the software development activities, the lack of formal metrics and standards in the estimation of cost and schedule is a constant in the industry. The dominant variable to estimate effort, cost, or time is the software's size to be developed, as has been described by several studies [4,5].

The size of the software can be measured in many ways and at different points in a software project life cycle. Up to now, there are five standards of software functional size measurement methods (FSMM). The ISO/IEC 19761 COSMIC method is the only standard belonging to the second generation, including several domains of use, like MIS (Management Information Systems), real-time infrastructure, etc., and also solving most of the problems that come with the FSMM of the 1st generation.

The COSMIC method arises in the year 2000; until now, in the literature reviewed, there is no documentation about the use of the COSMIC method to measure or estimate RPA software, as could be reviewed in the knowledge base of the COSMIC site (www.cosmic-sizing.org).

Any approach that helps determine if the RPA software can be measured with COSMIC would be useful in order to generate more formal estimations and compare the productivity behavior of this type of project against the other types of software in distinct domains like MIS.

The article presents a case study that is organized as follows: Section 2 explains the characteristics of RPA and how the industry has grown. In this section, an overview of COSMIC (ISO/IEC 19761) is presented. Section 3 explains the RPA robot used in this paper, along with the Robot’s environment, the process that has been automated, and the application of the COSMIC method to measure the RPA functionality. Section 4 presents a summary of the study, validity threats and suggestions for future work.

2 Background

2.1 RPA

Redesigning old information systems or designing new information systems is often costly in an automation project. Companies have been trying to automate routine tasks and business processes for a long time, often without proper Return on Investment (ROI). This kind of problem caused RPA technology to arise [1].

RPA tools' aim is to reduce monotonous tasks on employees so that workers have more time for value-added tasks [6]. Although there is another approach where RPA can cause potential replacement of human resources [1]. Figure 1 shows how a back-office agent performed a specific process, and Figure 2 shows the same process with an RPA Robot replacing the back-office agent.

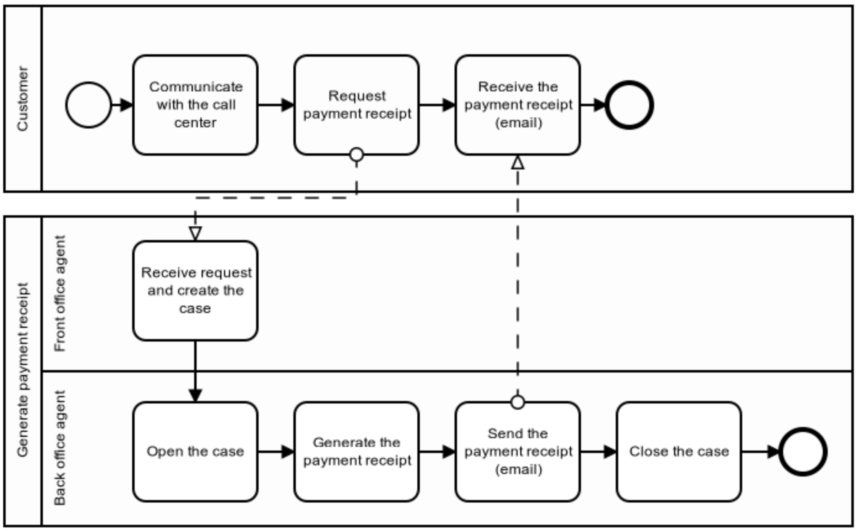

Fig. 1. Generate payment process before automation. Adapted from [6]

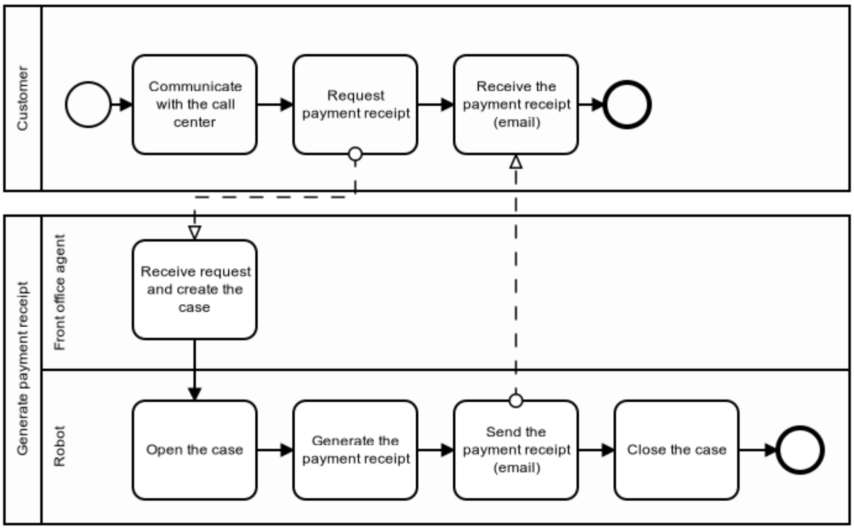

Fig. 2. Generate payment process after automation. Adapted from [6]

According to van der Aalst et al. [2] there are differences between RPA tools and other automation technologies. First, RPA uses an “outside-in” approach so that systems currently being used do not have to be redesigned. Rather than making changes to systems, humans are replaced by robots that identify applications on the screen and manipulate this software as a human does, in consequence the RPA robot acts like a functional user for the current systems. Secondly, one of the RPA objectives is to be robust regarding changes to the information systems. The user interface can suffer changes, but if the essential contents remain unchanged, the RPA software should adapt as humans do.

Furthermore, the relationship between the RPA Robot and humans is so that when a case turns out to be exceptional, the RPA Robot may hand over the case to a human. Syed et al. [7] also noticed that RPA is easier, cheaper, and faster to implement than other forms of automation.

Aguirre & Rodriguez [6] explain two more characteristics. RPA solutions do not require programming skills for software interface configuration. This is because RPA is set to work by just dragging, dropping, and linking icons, and RPA does not store any transactional data, so there is no need for a database.

There are a few papers, most of them presented in the form of a case study, that show the productivity and cost-saving results of implementing RPA. For example, in the case study presented by Aguirre & Rodriguez [6] an RPA Robot is implemented into a Business Process Outsourcing service provider, where an increase in productivity and capacity of approximately 20% was reported. Lacity, Willcocks & Craig [8] present a case study that shows one of the most positive results reported, where 300 Robots automated 25% of the company's office processes. These Robots perform the work of 600 people. The study also shows a return of investment of 200% for the first year after the implementation.

The positive results, along with the RPA trend, have caused the RPA market to grow. Forrester's estimated growth is that the RPA market will reach US $2.9 billion by 2021 and US $12 billion by 2023. There are approximately 15 main vendors of RPA solutions on the market [9, 10].

Considering the feature where RPA creates the functionality required easily by dragging, dropping, and linking icons, a natural source of cost savings could be the productivity increase, because the productivity could be defined as "the ratio of the number of products delivered by a process to the number of inputs" [11]. The need to measure the functionality of an RPA Robot is a fundamental issue to compare the productivity improvement with this type of project with the traditional developments and to define estimation models related to RPA projects.

2.2 COSMIC

In the software engineering field, various ISO/IEC norms have been developed oriented to measurement. Many years verifying various approaches showed that the concept of software size measurement based on its functionality, is a key attribute to the user and the technical team.

The ISO/IEC 14143 norm [12] includes a set of rules regarding size measurement in functionality units. For this kind of measurement, this standard proposes the following definitions:

"Functional size is defined as the size of the software derived by quantifying the Functional User Requirements" [12]

"Functional User Requirements (FUR) stands for a sub-set of the User Requirements describing what the software does, in terms of tasks and services" [12]

"Functional Size Measurement (FSM) is the process of measuring functional size" [12]

After about 40 years of improving various software FSM techniques, five of them (out of over 20) have been acknowledged by the ISO/IEC as conforming the rules laid down in the ISO/IEC 14143 norm and have taken on the form of the following standards: ISO/IEC 20926 (IFPUG), ISO/IEC 20968 (MKII), ISO/IEC 24570 (NESMA), ISO 29881 (FISMA) and ISO/IEC 19761 (COSMIC) [13]. Although the latter is the only one that belongs to the "Second Generation" of FSM methods, the others are also standards belonging to the "First Generation" [14].

COSMIC introduces its own homologated and standardized measure unit called Cosmic Function Point (CFP). 1 CFP represents the size of one data movement (Entry, Exit, Read, or Write). Therefore, functional size can be measured by counting the number of handled data movements (CFP). More information about the COSMIC method can be found in the COSMIC website (www.cosmic-sizing.org).

The official COSMIC website provides guidelines to apply the COSMIC method in specific situations. Some of the situations covered by COSMIC are real-time software [15], business applications [16], data warehouse and big-data software [17], service-oriented architecture [18], agile project [19], etc. There are studies [20] that show that it is possible to measure software size even in unguided situations or technologies like machine learning [4].

3 Case study: RPA proof of concept implementation for a banking organization

3.1 Methodology

This study is based on an RPA Robot that has been developed as a proof of concept (PoC). The Robot involves two companies, due to confidentiality issues, the name of the client company is not presented. The company Robot Digital TI (company A) was developing and providing RPA solutions to a second company (company B), a banking organization. Company A developed a project implementing RPA in the operations department of company B. Company A had communication with company B to understand the process that has been automated. When the process was well understood, a BPMN diagram that models the business process was created.

The input to develop this paper was the BPMN diagram (Figure 3), received from Company A; several interviews were developed with this organization's employees to understand each of the tasks. We also saw a demonstration of how the robot performs these tasks. With all this we had enough information to identify FUR at a functional process level, considering the RPA as a software that will act as a functional user for the other applications involved.

Having all the information allowed us to perform the 3 phases of the cosmic method following the principles from the COSMIC Software Context Model and the COSMIC Generic Software Model. Firstly, during the measurement strategy phase, we identified the FUR and the functional users of the robot. The mapping phase allowed us to identify the functional process derived from the FUR, along with data groups and the data movements of the functional process. Finally, in the measurement phase, the cosmic unit of measurement was applied to obtain the functional size of the software.

3.2 Robot's environment

The process named fixed-term deposit was chosen to be automated as a PoC because the activities that belong to this are repetitive, and the parts of the process are well structured. It is worth noticing that not all the variants of the fixed-term deposit nor all the activities of the process were automated. This Robot covers the simplest case of a fixed-term deposit.

The process diagram is presented in figure 3, showing 17 activities the operations department performed during the fixed-term deposit process. Out of these 17 tasks, 15 are automated by the Robot (marked with a green tick); the two remaining tasks (marked with a red cross) are not part of the PoC, so they are not taken into account for this case study. This process starts when, at a certain time of the day, the robot starts waiting for the business department to send a registration request.

The BPMN diagram presented in figure 3 shows the FTD business process of company B at a business level, that means, it does not have the granularity level to develop a measurement. Therefore, some subprocesses, like logging in to their systems, getting some specific data, downloading PDF files, etc., are missing in figure 3. The whole process, as it is done by the robot, is explained in section 3.3.

The RPA Robot has been developed using the "Automation Anywhere" tool and will be called FTD (acronym of "fixed-term deposit"). Three software pieces are essential to understand the Robot's functionality: the CRM System, the banking system, and a server that runs on CITRIX.

3.3 Applying the COSMIC method to FTD

According to the COSMIC website, the COSMIC Functional Sizing Methodology "is based on fundamental software principles, and is therefore suited to all types of software". Therefore, it can be applicable to measure RPA robots. COSMIC is based on representing the functionality of the software, that is mapping the functionality to four types of data movements that could apply to any software (Entry,eXit,Write,Read).

The approach we have taken to measure RPA robots is that when the robot manipulates the user interface, as a human would, by sending or receiving data groups, these manipulations are classified as entries and exits. According to the COSMIC manual, a functional user is a sender and/or intended recipient of data of a piece of software. When the robot writes and reads from other software's user interface, the RPA robot commits to the definition of functional user for the other software.

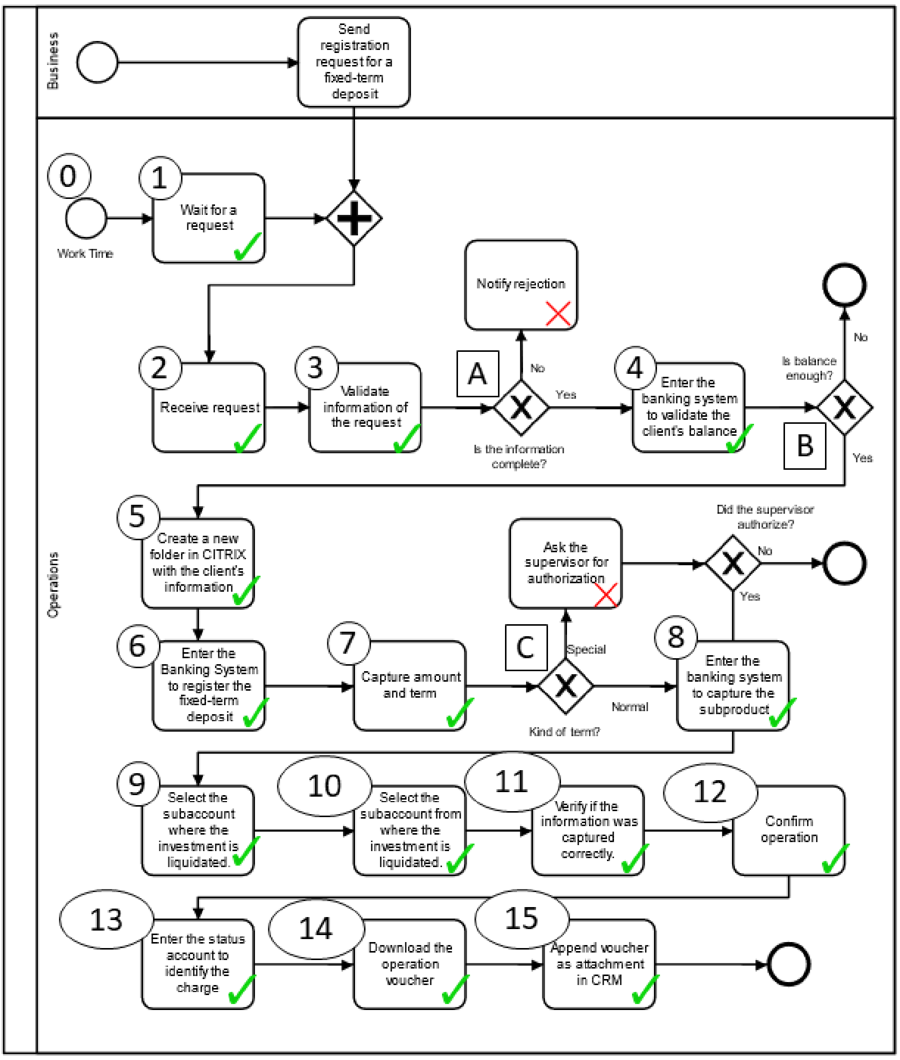

Fig. 3. BPMN diagram showing the fixed-term deposit process. The tasks that have been automated by the Robot are marked with a green tick. The tasks that have not been automated, therefore not taken into account in this case study, are marked with a red cross. The diagram has numbers in circles/ovals to identify the tasks automated by the robot, and it has letters in squares to identify decisions automated by the robot.

The robot reads and writes in the user interface, this could cause confusion when having, for example, an “Entry” type data movement and the verb “Read” on the subprocess description. To avoid this, it has been decided not to use the “write” and “read” verbs in the subprocess descriptions presented in Table 1, nor in the functional process description. Instead, when talking about the robot writing in the user interface, we use the verb “send”; when talking about the robot reading from the user interface, we use the verbs “request” and “receive”.

Below, Figure 4, presents the visualization of the RPA interacting with other software as defined by COSMIC Manual [20] and the comparison when the human functional user executes the same functionality.

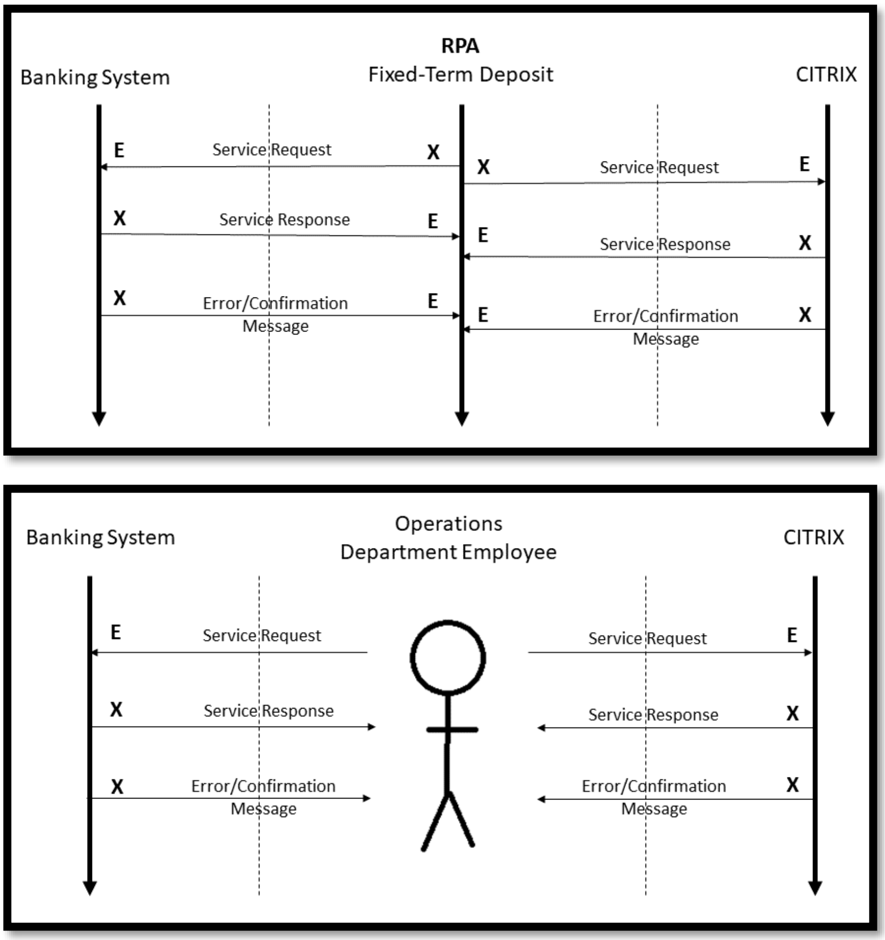

Fig. 4. Comparison between RPA functional user and human functional user when executing the same functionality

Phase 1: Measurement Strategy

The purpose of this measurement is to measure the FTD software using the Application Layer approach, that means “as a whole”, to enable the possibility to make a benchmark to compare the effort required to develop these applications against traditional web applications, and for estimation purposes. As presented in this paper, to show that the functionality provided by an RPA software can be measured using COSMIC (ISO/IEC 19761).

The scope of the measurement is the complete functionality of the FTD software, which only includes one functional process called “Fixed Term Deposit FUR”.

The COSMIC context diagram will be used in the strategy phase to help identify and show the functional users of the software to be measured. The context diagram, defined by the COSMIC method [20], of FTD software, is shown in Figure 5.

The FTD Robot has four functional users:

- **CRM:** The Customer Relationship Management System that company B uses.
- **Banking System:** The system that company B uses to do banking transactions and operations.
- **Citrix:** A server running on CITRIX operative system, where company B uploads some of their archives.
- **Clock:** The clock that notifies FTD to start working.

RPA does not store any transactional data; this leads to not having read or written data movements when measuring this kind of software.

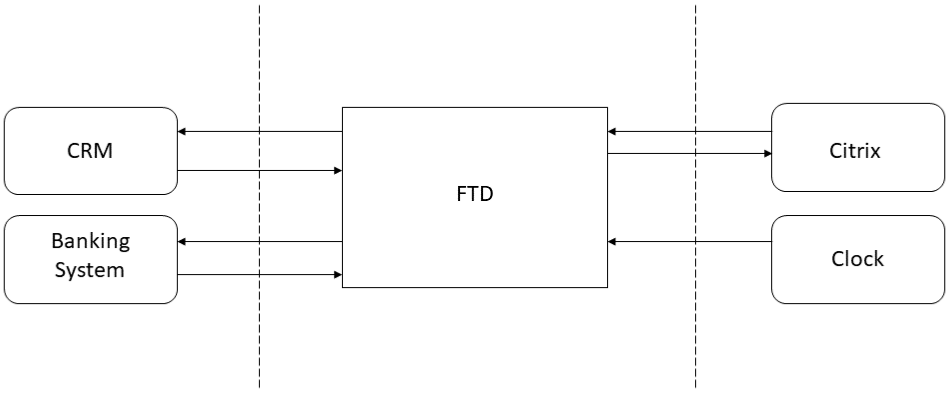

Fig. 4. Context Diagram of FTD Software

In this case, the tasks that the robot should do were presented with a BPMN diagram (Figure 3), which shows the business view of how the Fixed Term Deposit process works, along with a detailed explanation of how humans do this process. The BPMN diagram, along with the explanation, allows the development team to have FUR at a functional process level of granularity, considering that RPA uses an “outside-in” approach.

Functional Process – Fixed Term Deposit FUR

At a particular time, the Robot receives a clock tick to start the process. The Robot accesses the CITRIX server and then enters the CRM System. The Robot is requesting the CRM, every specific time, for a new client number to process. When there is a client number, the Robot receives it. The Robot requests the CRM for the client's request document by sending the client number. The Robot receives the client's request document (in PDF format) from the CRM.

The Robot obtains the data through the request document via OCR (Optical Character Recognition). Then, it verifies that the information supplied by the client is valid and saves the request document data in variables. It also sends the client's request document to CITRIX to be saved.

After sending the document to CITRIX, the Robot requests the banking system the client's information along with the account balance by sending the client number. The Robots receive the client's information along with the account balance. The Robot validates if the balance (included in the account data) is enough for it to continue the process. If this validation is successful, the Robot sends the term and amount to the banking system, after this, it sends the subproduct (all of these contained in the client's request document) to the banking system. It also sends the account data to CITRIX, so a new folder is created (the folder name has information from the account data).

Then it requests the client's subaccounts by sending the client data to the banking system. It receives the client's subaccounts from the banking system. The Robot selects and sends two subaccounts, the subaccount where the money generated by the investment will be liquidated, and the subaccount from where the funds for the investment will be taken.

After all this, the Robot requests the banking system for all the information saved by the system during this process (Client's request information). It receives all the information that the banking system has saved. The Robot verifies that the saved information is correct. The Robot confirms the operation, once the information is verified.

Following the operation's confirmation, the Robot requests the operation voucher (in PDF format) to the banking system by sending the client's information. The Robot receives the operation voucher (in PDF format). Then it requests the banking system for the account status by sending the client information. The Robot receives the account status from the banking system, then verifies that the fixed term deposit transactions are present in the account status. The Robot sends the operation voucher document to CITRIX to be saved. The Robot sends the operation voucher to the CRM to be saved as an attachment.

During all the process, the Robot sends CITRIX the instructions that it executes, so they are saved in a log.

Phase 2: Mapping Phase and Phase 3: Measurement Phase

Knowing the information presented in table 1, it is correct to say that the FTD functional size is 25 CFP.

Table 1 has 28 rows (1 row per subprocess) and 5 columns. The first column helps identify every subprocess. The second column shows the functional users related to each subprocess, from the RPA Robot’s viewpoint. The third column contains the subprocesses details. The fourth column shows the data groups presented in each subprocess. Finally, the fifth column shows the data movement types related to every subprocess.

ID	Functional User	BPMN Activity	Subprocess	Data Group	Data Movement Type
1	Clock	0	Receive a clock tick to start the process.	Clock Tick	E
2	CITRIX	1	Access CITRIX server.	Credentials Citrix	X
3	CRM	1	Access CRM System.	Credentials CRM	X
4	CRM	1	Request, every certain time, for a client number to process.	Client	X
5	CRM	1	Receive the client's number.	Client Number	E
6	CRM	2	Request the client’s request document (in PDF format).	Client Number	X
7	CRM	2	Receive the client’s request document (in PDF format).	Client Solicitude	E
8		3	Get the data from the request document via OCR. Verify that the information supplied by the client is valid and save the client’srequest data in variables.	n/a	n/a
9	CITRIX	3	Send the client’s request document to CITRIX to be saved.	Client Solicitude	X
10	Banking System	4	Request the client’s information along with the account balance.	Client	X
11	Banking System	4	Receive the client’s information along with the account balance.	Client, Account Data	E, E
12		A	Validate if the balance is enough to continue the process.		
13	Banking System	7,8	Send amount, term and subproduct.	Client Solicitude	X
14	CITRIX	5	Send the account data to CITRIX so it creates a new folder (the folder name has information from the account data).	Account Data	X

15	Banking System	9	Request the client's subaccounts.	Client	X
16	Banking System	9	Receive the client's subaccounts.	Subaccount	E
17	Banking System	9	Select and send the subaccount where the money generated by the investment will be liquidated.	Subaccount	X
18	Banking System	10	Select and send the subaccount from where the funds for the investment will be taken.	Subaccount	X
19	Banking System	11	Request the information that the banking system has saved during the process.	Control Command	n/a
20	Banking System	11,12	Receive the information that was captured and verify if it is correct. Confirm the operation.	Control Command	n/a
21	Banking System	14	Request to download the operation voucher (in PDF format).	Client	X
22	Banking System	14	Receive the operation voucher (in PDF format).	Operation Voucher	E
23	Banking System	13	Request the account status.	Client	X
24	Banking System	13	Receive the account status. Verify that the fixed term deposit transactions are present.	Account Status	E
25	CITRIX	15	Send the operation voucher to CITRIX to be saved.	Operation Voucher	X
26	CRM	15	Send the operation voucher so it is saved as an attachment in CRM.	Operation Voucher	X
27	CITRIX	n/a	Send the instructions executed so that CITRIX can save it in a log.	Instruction	X
				Total	24 CFP v5.0

Table 1. Measurement Table

It is worth noticing that row 27 has "n/a" in the BPMN activity column, this is because, even though the robot does this task, it is not related to any activity presented in the BPMN Diagram.

The COSMIC method is applied to get the functional size value of FTD by following the rule that says Functional Size = E + X + W + R. The functional size of the FTD software is the sum of all the data movements from the Fixed Term Deposit FUR functional process. The FTD software receives 8 Entries, sends 18 exits and has no writing or reading. This gives FTD a functional size of 24 CFP v5.0.

4 Summary and future work

The RPA market has been growing, and according to the market projections, it will continue growing. Thus, having a correct estimation for this kind of project will become more and more critical. Using COSMIC functional size to help estimate is a step forward to achieving this goal of having better estimations.

In this paper, an RPA robot from a banking operation was used to illustrate how the measurement of an RPA software can be made by using the COSMIC method, showing the appliance of the COSMIC method measurement process to this kind of software.

The use of the COSMIC method to measure the RPA software was successful. The FTD Robot has a functional size of 24 CFP v5.0. With the use of the COSMIC method, a productivity model for RPA projects can be generated, so that productivity can be compared with other kinds of software.

This paper obtains the FUR from a BPMN diagram generated by observing how a human does a certain task, which is the approach applied by the company that develops the RPA robot. The process should be well defined and monotonous, this leads to having certain facility to obtain the FUR at a functional process granularity.

4.1 Validity Threats

This paper presents a case study where the FUR were presented by a BPMN diagram because that is how company A works; there could be other ways to obtain FUR, but these other approaches are out of the scope for this work.

The way of making a BPMN diagram depends from person to person, so there can be tasks in the diagram that do not have any effect on the measurement process, or vice versa, tasks that are not explicitly shown in the BPMN diagram but do have effect on the measurement. This can cause confusion to the measurer and errors during the measurement process.

Some tasks caused some confusion, for example, task 6 in figure 7 does not have any effect in the measurement process. In addition, row 25 in table 1 says “Send the operation voucher to CITRIX to be saved”, but this task is not shown in the BPMN diagram, this FUR was only obtainable during the interviews performed to the employees of company A. Clear FUR are required to get a significant measure of the software using COSMIC, the same situation is presented in traditional software applications.

4.2 Future Work

As future work, it could be interesting to compare RPA technology to other kinds of automation technologies in terms of productivity (CFP/Effort). It is being mentioned that RPA projects are easier, cheaper, and faster to implement than other forms of automation, maybe because of the nature of how RPA software is developed (by dragging, dropping, and linking icons), and it has an “outside-in” approach. The comparison could approximate how much a software development company can save in costs by increasing productivity by developing automation projects using RPA instead of other automation forms.

Another proposal for future work is a case study where multiple RPA software are measured, and information about the projects of these software (effort, cost, time) is collected. This study could be beneficial to the companies that are providing RPA solutions, as this could show how to generate productivity models applied to this kind of software in order to estimate effort and costs, obtaining a correlation value between the functional size and these two aspects.

References

1. D. Šimek and R. Šperka, “How Robot/human Orchestration Can Help in an HR Department: A Case Study from a Pilot Implementation,” Organizacija, vol. 52, no. 3, pp. 204–217, 2019.
2. W. M. P. van der Aalst, M. Bichler, and A. Heinzl, “Robotic Process Automation,” Bus. Inf. Syst. Eng., vol. 60, no. 4, pp. 269–272, 2018.
3. C. Tornbohm and R. Dunie, “Market Guide for Robotic Process Automation Software,” Gartner, Inc., no. December, pp. 1–35, 2017.
4. A. Lesterhuis and A. Abran, “Cosmic sizing of machine learning image classifier software using neural networks,” CEUR Workshop Proc., vol. 2476, pp. 121–129, 2019.
5. P. Bourque, S. Oligny, A. Abran, and B. Fournier, “Developing project duration models in software engineering,” J. Comput. Sci. Technol., vol. 22, no. 3, pp. 348–357, May 2007.
6. S. Aguirre and A. Rodriguez, “Automation of a business process using robotic process automation (RPA): A case study,” in Communications in Computer and Information Science, 2017, vol. 742, pp. 65–71.
7. R. Syed et al., “Robotic Process Automation: Contemporary themes and challenges,” Comput. Ind., vol. 115, p. 103162, 2020.
8. M. Lacity, L. Willcocks, and A. Craig, “Robotic Process Automation: Mature Capabilities in the Energy Sector,” Outsourcing Unit Work. Res. Pap. Ser., no. October 2015, pp. 1–19, 2015.
9. C. Le Clair, “Robotic Process Automation: The 12 Providers That Matter Most And How They Stack Up Key takeaways,” Forrester Wave, 2018.
10. C. Le Clair, “Robotic Process Automation: The 15 Providers That Matter Most And How They Stack Up,” Forrester Wave, 2019.
11. A. Abran, Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers. 2015.
12. ISO; IEC, “Information technology — Software measurement — Functional size measurement — Part 6: Guide for use of ISO/IEC 14143 series and related International Standards (ISO/IEC 14143-6:2006(E)),” Iso/ Iec, 2006. [Online]. Available: https://www.iso.org/standard/60176.html. [Accessed: 27-Apr-2020].
13. B. Czarnacka-Chrobot, “Standardization of software size measurement,” Adv. Intell. Soft Comput., vol. 64, pp. 149–156, 2009.
14. A. Abran and C. Woodward, “Guideline on how to convert ‘First Generation’ Function Point sizes to COSMIC sizes,” no. November, pp. 0–53, 2016.

15. Lesterhuis, A., Londeix, B., Symons, C.R., “Guideline for Sizing Real-time Software,” 2016. [Online]. Available: https://cosmic-sizing.org/wp-content/uploads/2016/11/COSMIC-Method-v4.0.1-Guideline-for-sizing-real-time-software-v1.1.1.pdf. [Accessed: 19-May-2020].
16. A. Symons, C.R., Lesterhuis, “Guideline for Sizing Business Application Software - COSMIC,” 2017. [Online]. Available: https://cosmic-sizing.org/publications/guideline-sizing-business-application-software-2/. [Accessed: 19-May-2020].
17. A. Symons, C.R., Lesterhuis, “Guideline for sizing Data Warehouse and Big Data Software, v1.2,” 2018. [Online]. Available: https://cosmic-sizing.org/publications/guideline-for-sizing-data-warehouse-and-big-data-software-v1-2/. [Accessed: 19-May-2020].
18. C. R. Fagg, P., Lesterhuis, A., Rule, G., Ungerer, G., Galegaonkar, S., Natarajan, K., Santillo, L., Vogelezang, F.W., Jain, P., O’Neill, M., Symons, “Guideline for Sizing Service-Oriented Architecture Software - COSMIC,” 2015. [Online]. Available: https://cosmic-sizing.org/publications/guideline-for-sizing-service-oriented-architecture-software/. [Accessed: 19-May-2020].
19. S. Trudel and L. Buglione, “Guideline for Sizing Agile Projects with COSMIC,” in Proceedings of the IWSM/MetriKon/Mensura, 2010.
20. Common Software Measurement International Consortium (COSMIC), “Measurement Manual v5.0,” 2020.