

**Manuela PETRESCU,
Emilia Loredana POP**

*Department of Computer Science, Faculty of Mathematics and Computer Science,
Babes-Bolyai University, Cluj-Napoca, Romania*

COMPLEXITY METRICS IN DEVELOPING APPLICATIONS AND A METHOD FOR COST STANDARDIZATION

*Case
Study /
Original
Research*

Keywords

*Cost Standardization;
Digitization;
Complexity metrics;*

Abstract

The digitization process has grown exponentially in the last decade, and does not show any signs of slowing down, as more and more money are invested in automatic processing, in e-government services and e-divestment. In this context, the problem of cost standardization appears for digital application acquisitions performed by the public services or by economic entities that are connected to the states. In the public auctions, especially those that are funded with European money or public money, the price for digital applications exceeds the average rates, and sometimes the developed applications do not meet the expectations or are not able to cope with the data volumes. Thus, there is the need of having a price standardisation for the digitization process, and here is suggested a method to find a solution for this problem. In this paper, a method that takes into consideration economical, architectural, licensing, complexity factors in order to compute a quotation for a product is proposed. This method is a tool that can be useful for the service providers and the beneficiaries and help them understand the implications of their decisions, of the different workflows in terms of final cost.

INTRODUCTION

According to a study made by Transparency International, billions are lost due to corruption, and Romania is not an isolated case. For example, more than 38.6 billion is lost each year in Romania due to corruption; this value reflects the high corruption in the public administration, 68% of Romanian citizens believe that their lives are daily affected by it (Ifla, 2021). A large part of this money is lost in the digitization process. Two relevant examples from Romania: A huge corruption case was related to the digitization of the CNAS (National Health Insurance Company) that signed a contract for a digital health system that should have been delivered in two years and should cost 119 million EUR. The project took six years to be implemented and cost over 300 million EUR (Petrescu and Karamitsios, 2019; Phillips, 2003). An analysis of this contract reveals deficiencies, for example the provider did not have any penalties for delaying; the penalties were defined only for CNAS. In case of early contract termination, CNAS should support by itself all the costs.

There are a lot of functional problems – as the system was not designed to support many users and is not performing as expected (The Standish Group, 2020). The e-Romania site that can be found at www.portaleromania.ro, took more than two years to implement and cost more than 11 million Euros (Vasilache, 2018). How was possible to have such an outcome?

Toomas Hendrik Ilves, ex-president of Estonia, one of the most digitised countries according to DESI (Digital European Society Index), concluded in an interview that in fact going digital is not about technologies, human resource, or budget allocations, it's about "political will, policy, laws and regulations, in that order". To all these works, there are need laws that underpin the system, and so, the digital identity should be defined and then, should be set out the regulations to avoid the abuses (Hendrik, 2018).

The abuses can be corrected by standardization, due to this fact, correctly estimating the price of an application is a major problem and has an impact in the state's digitization process.

BACKGROUND

Current status

The digitization process is growing in an exponential pace, as more and more companies, economic entities and states are digitizing their internal processes and flows in order to ensure a more effective communication, information transparency, and nonetheless to provide 24/7

services for their citizens (Estonia's case is relevant: more than 90% of the services offered by the state are digitised and everything except marriages, divorces and real estate transactions can be performed online) (Petrescu, 2020).

Other example is given by the CHAOS Report (The Standish Group, 2020) which studies more than "50,000 in-depth *software* project profiles of the previous five years" from all over the world and concludes that there is a need for a better planning, cost and effort evaluation as 19% of the number of projects failed, 50% were challenged and only 31% were successful (The Standish Group, 2020).

All the applications and e-services implemented come with a cost, in terms of money and the effort invested; but most of the time, a standard for the real cost of an application is hard to compute as it depends on a set of factors such as: the application type, the complexity, the hardware solutions, the licensing, humans' resources, and so on. The analysis revealed the absence of methodology solutions, the problem of developing quantitative measures, as the complexity metrics that are hard to use for a complex application. These issues are encountered in all types of applications, web-based or mobile. The plan for this paper is not to pursue research in project management, but to examine how the main metrics from project evaluation should be. The project planning, the risk management and the resource allocation were documented in digitization documentation and guidelines (Ifla, 2021; Nokes, 2007; Phillips, 2003).

There are papers that tried to summarize different approaches for monitoring digitization internationally and others paid attention to a specific type of application. In many approaches, the value of digitization is associated with providing access to data, to large collections or to preserve them; "Analysis revealed that most reports were exploratory or pilot studies performed without any concern for the regular collection of data in the future. This resulted in the absence of sound methodology solutions based on digitization definition in most cases." (Manžuch, 2009). Due to this fact, the research papers and the guidelines were elaborated for these specific types of applications, used mainly in the libraries, museums and archives and many reports and researches for digitising institutions relied on specific institutional typologies or application types (Kockum and Dacre, 2021; Princeton University Library, 2021; CLIR, 2001). There are papers that present a comparative analysis of effort for different estimation in mobile applications (Pandey, Litoriy and Pandey, 2020), others investigate the optical character recognition (OCR) technology (different resolution settings and binarization levels in order to obtain a high accuracy (Stančić and Trbušić,

2020), some papers expose the relationships in forms of maps, charts, reports and globes, or others use reality capture technology (Wang and Cho, 2014; Wong, Ge and He, 2018), which involves the use of 3D laser scanning and photogrammetry for acquiring accurate geometrical/spatial information. The main idea is that besides the complexity of an application, the effort estimation techniques must add the know-how variable.

Constraints and difficulties

Due to a huge variety of applications, as the number diversity of the factors that can influence the outcome of a project, a metric that would fit all the projects is hard to be found. The first metric proposed by Halstead was based on computing the number of operators and operands, and even if the metric was useful when it was designed, it is deprecated, irrelevant and hard to apply today (Al-Qutaish and Abran, 2005). Today's application reached out a new limit, as millions of lines of codes used to build it. For example, Windows XP 2001 has more than 30 million lines of code, Apache OpenOffice has more than 20 million lines of codes (Desjardins, 2017), without mentioning the fall back: any functionality can be written more concise or using more lines of code. Due to this fact, other metrics should be evaluated. Another metric designed in the same period by McCabe was the cyclomatic complexity, measuring the number of linearly independent paths through the program's source code (McCabe, 1976). Additionally, the functions points were defined as a "unit of measurement" for the amount of business functionality that was provided for a user or for other applications, several recognized standards were based on the function point, and the latest one, named ISO/IEC 19515 was proposed in 2019 by Object Management Group (OMG).

As these metrics reached their limit, a metric for Object-Oriented Design was proposed in 1994, introducing six complexity metrics: coupling between object classes, lack of cohesion between methods, weighted methods per class, response for a class, depth of inheritance tree and number of children classes (Chidamber and Kemerer, 1994).

There can be at least one more complexity metric that should be added to these metrics: the degree of parallelism; the parallelism in an application is influenced by the architecture and it is correlated with the number of users/threads that can use that application in parallel.

MAIN RESULTS - PROPOSED METRICS

Most of the applications that are developed today for providing national e-services are more complex than the applications developed in the past, and there are a variety of factors that influence the cost

of a project and can increase the risk of a project. Due to this fact, the risk management related papers and methodology were studied and some solutions were proposed in order to decrease it (Albrecht, 1979; Alexsandro, Robson and Jorge, 2021; Lavanya and Malarvizhi, 2008).

Providing an estimation cost for an application without knowing that it is architecture, the hardware or the licensing costs it is like playing the roulette, as the licensing cost for similar applications and features can have a huge impact. If the hardware cost is relatively easy to determine, and the prices between the same components from different manufacturers are on the same scale, this is not the case with licensing cost for libraries or for different components. For example, the cost ratio per processor for an Oracle vs SQL Server (MS) installation is 2.56 (Burlinson, 2015), (Oracle is more expensive), and adding the cost for the other features such as compression, Query Store (MS) vs Tuning Pack (Oracle), Availability Groups (MS) vs Active Data Guard (Oracle), the final total cost can increase the a ratio of 12.6 (the total cost for MS was computed to \$109,980.00 versus \$1,388,000.00) (Blake, 2016).

Number of users' metrics

The expected number of users of an application is important as the order of the number and can have an impact on the performance; a larger number of users implies a higher degree of parallelisation: from data storage needed for each user to network capabilities: amount of data sent back and forth to the server for each user (size, format), and to processing power. The expected number user should be considered equal to the maximum number of the concurrent users. Due to the fact that the metric is proposed especially for e-services, the expected number of users can start from a few thousands to millions. The number order is correlated with the complexity, so it is proposed a metric based on the size: number of users' complexity $NoUC$, where:

$$NoUC: X \rightarrow Y,$$

$$x \in \mathbb{R}, \text{ where } \mathbb{R} \text{ is the set of real numbers,} \\ y \in Y \mid 1 < y < 10, y \text{ is the order of the number of users.}$$

$$NoUC(x) = 1, \text{ if no is less than } 100000 \text{ users,}$$

$$NoUC(x) = y, \text{ if no is greater than } 100000 \text{ users,} \\ \text{where no is number of users.}$$

The function was proposed as there should not be a huge impact on the application when the user's number increase from 100.000 to 200.000 users, but there could be a difference when the number is

increased from 10.000 to 100.000 users. The same approach is valid for a gap between 1 and 2 million users compared to the gap from 100 thousand to 1 million. Once an application was designed to accommodate 10 million users, it can easily accommodate one more million.

Transactions Complexity

The next discussion is about the transaction's complexity there are some factors that stick out: the objects that are treated (updated/added/deleted) with an accent to the number of their attributes and their complexity and the type of the transactions.

Regarding the number of the attributes for an object, there are discussions regarding their number, as there is not a definite conclusion regarding the recommended number. Are 20, 30 or 40 attributes too many? When it's recommended to reorganize them into a new class? The type of these attributes should be taken into consideration, as an attribute having a predefined type (String, Integer, so on) is easier to deal with compared to an object or a reference to a user/application defined object. The complex attribute types are related to the coupling between object classes metric, the reason why the proposed function is a recursive one. However, it should take into account that some of the objects can be lazy instantiated, that's why a factor $c \in [0..1]$ should be added.

Each tuple of ten or less than ten general attributes number will be notated with n and equal with $Math.floor(numberOfAttributes/10)$. Each tuple of ten or less than ten attributes defined as an object (or reference) to an application defined object will be notated with $tc(n)$, where tc is a function, in which the result value should be correlated with the complexity of the referenced objects. Next it is considered $tc: X \rightarrow Y$ a set of objects in the application and Y be a subset of positive real numbers. Then

$n = Math.floor(o.numberofAttributes/10)$
is for the attributes defined as basic types, and
 $n1 = o.numberofAttributes$
is for the attributes that are objects or references to objects, and consequently it is obtained
 $tc(o) = n + c * tc(n1); c \in [0..1]$.

Sensitive data metric

The information related to an application can be open data, and everyone can have access to it. Inside of it, there can be sensitive data, when specific users can access specific information. In the application where sensitive data is involved, security protocols (for example assigning roles and rights for the different user and/or user categories should be implemented) and the steps involved are added to the complexity. This metric is not referring to privacy metrics as defined in (Clifton, 2009), "Measures to determine the susceptibility of

data or a dataset to revealing private information", but it measures the complexity added by the security and data protection mechanism.

Let r be the number of roles and rights, so for each sensitive object (category), the complexity can be defined as a function, and s be the number of sensitive data categories, where $s \in \mathbb{R}$, and \mathbb{R} is the real number set. If the object is not a sensitive data object, the value of s would be 1. The metric is defined as a function,
 $f: DataSet\ o \mid sensitive\ objects \rightarrow \mathbb{R}$, with $s \in \mathbb{R}$, and consequently it is obtained

$$f(o) = s * tc(o) \mid s \in \mathbb{R}, o \in DataSet\ o \mid o\ is\ a\ defined\ object.$$

Transactions or flow volumes

The transaction volumes are strictly correlated with the performance of a system and are independent of the number of users, as one user (an institutional one) could generate a large number of transactions. The proposed metric is a linear function, with a growth rate / pace similar to one from the number of user's function. By considering the previously defined function $tc(o)$ and the sensitive data metric, it is also defined:

$$fy: X \rightarrow Y, \{x \in X \mid x\ is\ the\ number\ of\ transactions, y \in Y \subset \mathbb{N}, y\ is\ the\ size\ order\ of\ the\ number\ of\ transactions\}$$

$$fy(o) = y * s * tc(o) \mid s \in \mathbb{R}, o \in DataSet\ o \mid o\ is\ a\ defined\ object.$$

Flow complexity for CRUD operations

The complexity of the CRUD (Create, Read, Update and Delete) operations is related to the operation type, as a read operation is usually "less expensive" and can be easily parallelized compared to a write or a delete operation. Correlated with the previous metric, the following cases appear (n , $n1$, $fy(o)$) were previously defined):

- Read operations:
 - For simple attributes the complexity is defined as $fc(o) = n * r$;
 - For complex attributes / related objects the complexity is defined as $fc(o) = n1 * r * fy(o)$;
- Write operations:
 - For simple attributes the complexity is defined as $fc(o) = n$;
 - For complex attributes / related objects that imply coupling, the complexity is defined as $fc(o) = n1 * fy(o)$;

where r is read complexity, $r \in [0..1]$.

Business complexity

Business complexity is defined as the number of states (NS) an object has to go through before it reaches the final state. For example, creating an invoice could have the following state that can require manual intervention: Create the invoice - status *created*, Approve it (check if the merchandise can be delivered) – status *approved*, Payment initialised - status *inPayment*, Payment complete – status *paid*, Merchandise delivered - status *complete*. Business complexity is an important metric as correlated to the coupling between object classes parameter can influence in an exponential way the total complexity of the project and implicit, it's cost.

If the coupling between the objects, the average depth of inheritance tree, the complexity of crud operations, the business complexity *BC* can be formalised as a function with exponential growth.

Next it is considered *ObjNo* the number of objects defined in the application, and consequently, the formula is:

$$BC = \text{sum}(NS(o) * fc(o) * fy(o)) / ObjNo.$$

Testing complexity

Testing complexity (*TC*) is tightly related to the business complexity (*BC*), when the business complexity grows, the number of tests performed is increasing, and so is their complexity. The testing is an important phase in the developing an application and it should be treated with the importance it deserves; improved software quality translates into strongly positive return of investment (ROI) (Capers, 2011).

$$TC = BC * t, t \in [0..1].$$

$$TC = t * \text{sum}(NS(o) * fc(o) * fy(o)) | ObjNo, t \in [0..1].$$

Developing language and know-how

The programming language used for developing an application has an impact as there is one cost to develop a web-based application using java, react or php and other thing is to develop in COBOL for example. Related to the developing language is the human resource know-how issue, and when estimating a cost for an application, the average programmer's cost (as it differs depending on location) should be considered. As these parameters are related, there is a need to denote them with *CP* (cost per programmer). In the end, there are two other costs that should be taken into consideration: hardware cost, installation cost and training cost. The training cost can differ a lot depending on the application, on its complexity and how its *UI* (user interface) and the flows were designed. Nonetheless, the maintenance cost can get to a 20% - 30% per year from total project cost

in the first years of usage. All these costs will be summarized as additional costs *AC*.

The final cost formula should take into consideration all the metrics proposed before and should also be compared to the cost of similar applications if there is available data. The comparison process is known as parametric estimation (*PE*). Parametric estimates are based on historical data, comparing similar applications and calibrating them with current development cost. The final formula for the cost of a project (*FFCP*) should be based on the project's complexity, human resource and should also include additional costs, and so the final formula should be

$$FFCP = ((NoUC(x) + BC + TC) * CP + AC) * PE.$$

So, the cost of a project will depend on various parameters and there is a need to analyse each aspect to obtain the correct and precise final value. Due to this cost value, improvements can be given and also an optimization can be proposed. Consequently, one should pay attention to each detail, allocate enough time and resources (human, social, work, and so on) for getting the best result for a project cost and for the involved team.

CONCLUSION AND FUTURE WORK

When estimating a project, there are relatively fixed costs (as hardware cost or licensing) but the developing cost is variable and is tightly related to the application complexity. However, analysing the complexity for a project should take into consideration multiple metrics and factors, including the human factors. Most of the time the human factor that performs the estimation takes decisions based on the previous experience. However, the human factor impact is harder to measure as each person's know-how is different and so are their estimations, but in a team, some average metrics can be used. The best approach when estimating a project will be to use metrics independent of human factors, metrics that can be validated against older projects. So, this paper had proposed how to calculate the cost for a project by using metrics.

The future work will imply a task to find the values of the parameters for different applications based on their functionality (presentation sites, data storage and presentation for museums for example, simple applications and complex applications) and to determine the correlation between the growth of the variable in the proposed formulas and the complexity of each type of application. The final goal is to determine in a more reliable way the cost of each application type, to establish cost

standardization, thus allowing a better cost prediction.

REFERENCE LIST

- [1] Albrecht, A.J. (1979). Measuring Application Development Productivity. Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, IBM Corporation, 83-92.
- [2] Alexandro, S.F., Robson, L., & Jorge, L.V.B. (2021). A risk prediction model for software project management based on similarity analysis of context histories. *Information and Software Technology*, 131, doi:10.1016/j.infsof.2020.106497.
- [3] Al-Qutash, R., & Abran, A. (2005). An Analysis of the Design and Definitions of Halstead's Metrics. Proceedings of the 15th International Workshop on Software Measurement (IWSM'05), Québec, Canada.
- [4] Blake, T. (2016). Comparing SQL Server vs. Oracle License Cost. Retrieved from <https://seniordba.wordpress.com/2016/09/01/comparing-sql-server-vs-oracle-license-cost/>.
- [5] Burleson, D. (2015). Costs of SQL Server vs. Oracle, Oracle Database Tips. Retrieved from http://www.dba-oracle.com/t_cost_sql_server_vs_oracle.htm.
- [6] Capers, J. (2011). Economics of Software Quality. Addison-Wesley Professional.
- [7] Chidamber, S.R., & Kemerer, C.F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6), 476-493.
- [8] Clifton, C. (2009). Privacy Metrics. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. Retrieved from https://doi.org/10.1007/978-0-387-39940-9_272.
- [9] Desjardins, J. (2017). How Many Millions of Lines of Code Does It Take?. Retrieved from <https://www.visualcapitalist.com/millions-lines-of-code/>.
- [10] Hendrik, I.T. (2018). E-stonia Takes Off. Retrieved from <https://www.imf.org/external/pubs/ft/fandd/2018/03/trenches.htm>.
- [11] Ifla, (2021). Guidelines for Digitization Projects for collections and holdings in the public domain, particularly those held by libraries and archives March 2002. Retrieved from <https://www.ifla.org/files/assets/preservation-and-conservation/publications/digitization-projects-guidelines.pdf>.
- [12] Kockum, F., & Dacre, N. (2021). Project Management Volume, Velocity, Variety: A Big Data Dynamics Approach. *Advanced Project Management*. 21(1). Retrieved from SSRN: <https://ssrn.com/abstract=3813838> or <http://dx.doi.org/10.2139/ssrn.3813838>.
- [13] Lavanya, N., & Malarvizhi, T. (2008). Risk analysis and management: a vital key to effective project management. PMI® Global Congress 2008, PA: Project Management Institute.
- [14] Manžuch, Z. (2009). Monitoring digitisation: lessons from previous experiences," *Journal of Documentation*, 65(5), 768-796.
- [15] McCabe, T.J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, SE-2(4).
- [16] Nokes, S. (2007). *The Definitive Guide to Project Management*. 2nd Edition," London (Financial Times / Prentice Hall).
- [17] Pandey, M., & Litoriy, R., & Pandey, P. (2020), Validation of Existing Software Effort Estimation Techniques in Context with Mobile Software Applications. *Wireless Personal Communications*. 110, 1659-1677.
- [18] Petrescu, E.D., & Karamitsios, Y. (2019). Corruption in Romania: a European affair. Retrieved from <https://www.euractiv.com/section/justice-home-affairs/opinion/corruption-in-romania-a-european-affair>.
- [19] Petrescu M. (2020). Digitalization process of public services. Romania's example. 13th Joint Conference on Mathematics and Computer Science, 2020, ELTE, Hungary (MACS 2020). Accepted.
- [20] Phillips, J. (2003). *PMP Project Management Professional Study Guide*. McGraw-Hill Professional.
- [21] Stančić, H., & Trbušić, H.Ž. (2020). Optimisation of archival processes involving digitisation of typewritten documents. *Aslib Journal of Information Management*, ahead-of-print. doi:10.1108/ajim-11-2019-0326.
- [22] Vasilache, A. (2018). De ce nu mai poate fi accesat portalul de 12 milioane de euro e-România: Ministerul Comunicațiilor susține că este performant, dar neinteresant pentru cetățeni și autorități [Why the 12 million euro e-Romania portal can no longer be accessed: the Ministry of Communications claims that it is efficient, but uninteresting for citizens and authorities]. Retrieved from <https://economie.hotnews.ro/stiri-telecom-22395924-nu-mai-poate-accesat-portalul-12-milioane-euro-romnia-ministerul-comunicatiilor-sustine-este-performant-dar-neinteresant-pentru-cetateni-autoritati.html>.
- [23] Wang, C., & Cho, Y.K. (2014). Automatic as-is 3d building models creation from unorganized point clouds. *Construction Research Congress*, 2(3), 917-924.

[24] Wong, J.K.W., Ge, J., & He, S.X. (2018). Digitisation in facilities management: A literature review and future research directions. *Automation in Construction*, 92, 312-326.

Other Sources

[25] Princeton University Library (2021). Best Practices for Digitization: General Guidelines and Information. Retrieved from <https://library.princeton.edu/digital-collections/general-guidelines-and-information>.

[26] Council on Library and Information Resources (CLIR) Washington, D.C. (2001). *Building and Sustaining Digital Collections: Models for Libraries and Museums*. Retrieved from <https://www.clir.org/wp-content/uploads/sites/6/2016/09/pub100.pdf>.

[27] The Standish Group (2020). *CHAOS Report 2020*. Retrieved from <https://standishgroup.myshopify.com/>.