Product Delivery Improvement in a Software Factory Contract Applying Learning Curves

Francisco Valdés-Souto[1] [0000−0001−6736−0666], Daniel Torres-Robledo[2], and Hanna Jadwiga-Oktaba[1]

[1] National Autonomous University of Mexico
Science Faculty
CDMX, Mexico
fvaldes@ciencias.unam.mx
hanna.oktaba@ciencias.unam.mx

[2] National Autonomous University of Mexico
Research Institute in Applied Mathematics and Systems
CDMX, Mexico
dtorres@ciencias.unam.mx

Abstract. In software development, the management of standardized metrics are not as frequent as it should be, which encourages the immaturity of software engineering. Currently, few companies use standards for the software functional size measurement (i.e. COSMIC); however, an increase in the adoption of this practice is emerging, derived from the need to have greater certainty, both in the estimates and in the management of their projects.

A problem faced by companies that already use standardized metrics is knowing formally what proportion of improvement can be required of suppliers as they gain more experience as the time of the customer-supplier relationship passes.

This paper presents a proposal to determine the learning ratio of a supplier in order to request improvement of the productivity factor (*PDR*) with which the supplier has been worked in previous cycles through a real case study in the Mexican industry, using the learning curve theory.

Keywords: COSMIC · Learning Curves · PDR · Productivity · Estimation · PDR Improvement.

1 Introduction

It is known that the functionality of a software product can be measured using some functional size measurement method (*FSMM*), for example, *COSMIC*

ISO/IEC 19761. The possibility of measuring size allows us to also measure productivity in software development.

When a software provider is contracted for various sequential periods to develop products in the same or similar context, know-how is obtained about the problem domain, the development context and the technologies utilized. This can also be observed in individual projects, since for large projects higher productivity is observed than for small projects[6][15]. When this situation occurs, economies of scale are gathered and production cost is reduced, that is, the effort required to develop one software unit is less than in previous periods because the know-how acquired, which means that an increase in productivity derived from learning gathers, while more know-how more increase in productivity, however, this is not linear and tends to a limit.

After having hired a software provider for a certain number of periods, it is natural that the client wants to have an improvement factor that impacts a reduction in cost or increase in productivity related to a better knowledge of the context obtained through the time from the provider. However, there is difficulty in defining this factor formally for the projects in subsequent periods, since it must be well-founded and consistent value, otherwise, it could compromise the success of the projects.

The learning curve theory is a tool for to estimate the recurring costs in a production process and is based on the common observation that, by repeating a task, it can be completed in shorter periods of time or required less effort to be developed, allowing to determine the learning level in repetitive activities, which serves to make estimates[7].

In the literature reviewed, this learning factor is found in different areas of the industry, such as in microenterprises[5], manufacturing[12] and construction[8], where analysis of learning curves has been used to estimate improvement expected over time.

However, in the literature reviewed for the software development, a formal and well-founded way to estimate the productivity improvement factor in subsequent projects has not been identified; having a mechanism to do so is useful for companies that contract software developments for consecutive periods.

This article presents a case study to determine the degree of learning considering different periods in which a supplier worked under the same context by applying the learning curves. Using the *PDR* applied in previous periods by the supplier, the idea is to determine a reliable improvement factor through a formal mechanism such as the learning curves theory.

The article is organized in the following manner: Section 2 mention the background of the measurement of the functional size of software using COSMIC. Section 3 shows the theoretical bases of the learning curves and the different approaches for these theories. Section 4 is the case study of this paper. It shows the data and the calculations to use the estimating lot cost using unit theory approach to estimate the *PDR*, which is possible and achievable by the provider, and quality criteria for the model obtained. Finally, the conclusions are presented, and the *PDR* expected for the first semester of the year 2020.

2 Measurement of the functional size of software using COSMIC

Functional size is the only current standard measurement of the software [14]. However, there are currently two generations of *FSMM*, with *COSMIC ISO/IEC 19761*[1] being the only second-generation *FSMM*, which was generated based on the *ISO/IEC 14143* standard and the experience of the first generation methods[18]. Which implies that it solves most of the problems presented by these methods, such as the management of concepts existing at the time that the first generation methods were created and currently those are not in use, the scope of application of the methods, unpractical measurement scale, in addition to having a reduced domain of application.

The COSMIC method was initially accepted by *ISO/IEC JTC1 SC7* as an International Standard in December 2002. The current version is *ISO/IEC 19761*: 2011 Software Engineering - COSMIC - A Functional Size Measurement Method (actually *ISO/IEC 19761*)[1]. All the rules, principles and examples to perform functional size measurements using the COSMIC methodology[3].

3 Learning Curves

Learning curves analysis is a theory that allows us to estimate recurring costs in production processes[7].

In the theory of learning curves, the dominant factor is the direct work required to complete the task or product. This is based on the observation that completing a task several times generates learning for the person who performs it, allowing them to finish this task in less time the next time it is made.

There are two predominant theories:

- the Unitary Theory (UT)[10], and
- the Accumulated Average Theory (AAT)[10]

Both theories are based on the fact that; if a task is repeated several times, the experience is gained, allowing the tasks to be performed in a shorter time for the next iteration[7], that is, increasing productivity.

The main differences between these theories depend on the data required to carry out the analysis; for context with considerable variations in costs or design, it is recommended to use the AAT since it allows reducing the estimation risk. On the other hand, UT is recommended to be used when the available data are accurate.

Considering this, for this paper, we will use the Unitary Theory, which defines two types of approaches, estimating unit cost and estimating lot cost.

3.1 Estimating unit cost using Unit Theory

If there is learning in the production process, the cost of a 2k unit is equal to the cost of unit k times the slope of the learning curve (J. R. Crawford 1947). This means that for an 80% Learning Curve, there is a 20% cost reduction each time the number of units is doubled, for example, unit 2 is 80% the cost of unit 1 and the cost of Unit 4 is 80% the cost of Unit 2.

The following equation defines the learning curves using Unit Theory:

$$Y(x) = A * x^b \tag{1}$$

Where:

- $Y(x)$ is the cost of unit number x.
- A is the cost of the first unit.
- x is the unit number x.
- b is a constant which represents the slope of the learning curve.

To apply the equation 1, it is necessary to have the cost data for each unit produced.

3.2 Estimating lot cost using Unit Theory

However, because the cost of production for each unit is rarely reported, a viable option is to estimate the production cost in lots.

To estimate the production cost in lots, equation 1 is used; however, it is necessary to adjust the information of each lot. Considering that the analysis of learning curves requires knowing the unit number and the cost associated with each one, therefore these values will be represented by the lot midpoint (LMP) and the average cost per unit within it (AUC). The following operations are performed to calculate these values.

Calculating the exact LMP value is an iterative process, but it can be easily estimated using an approximation, as shown in equations 2, 3, and 4.

For the first lot:

$$If\ lotSize < 10,\ then\ LMP = \frac{lotSize}{2} \tag{2}$$

$$If\ lotSize \geq 10,\ then\ LMP = \frac{lotSize}{3} \tag{3}$$

For all subsequent lots:

$$LMP = \frac{F + L + 2\sqrt{F \times L}}{4} \tag{4}$$

Where:

- F is the first unit number in a lot.
- L is the last unit number in a lot.

Since the initial and final unit values for each lot are cumulative, we must find the F and L values of each one in order. For example, for 3 lots of 50 pieces each, the first lot goes from unit 1 to 50, the second from 51 to 100, and finally, the third goes from unit 101 to 150.

Finally, to calculate the AUC value of each lot, divide the number of units in the lot by the cost of developing it, as shown in equation 5:

$$AUC = \frac{TotalLotCost}{LotSize} \tag{5}$$

4 Case Study

4.1 Method

To conduct this case study in a real world context, the methodology proposed by Runeson[9] was followed, which consists of five steps (table 1): study design, data collection, evidence collection, analysis of collected data and results report.

Table 1. Methodology to conduct the case study.

1 Study design	**Q1:** How to estimate the PDR in the next semester? **Object of study:** 21 projects developed along 4 semesters
2 Data collection	**Data of interest:** size of the software, effort
3 Evidence collection	**Evidence metrics:** *CFP*, *PDR*, Productivity
4 Analysis of collected data	**Quantitative analysis:** calculus of Lot Mid Point, Average Unit Cost, slope of learning curve
5 Study reporting	**Report of the case study:** methodology and results

4.2 Case Study

The analysis of the learning curves theory in this document has the objective to obtain the learning degree that formally may be requested for subsequent periods from a supplier to construct software projects. This objective is pursuit based on the productivity previously shown.

In this case study, the analysis will be performed on information from a contract between a Mexican government entity from the energy sector and a software provider. Due to confidentiality issues, the name of the entity will not be mentioned in this document.

Every certain period, approximately two years, this entity contracts software factories to develop the project it requires. The current provider for the software development service was contracted for two years, 2018 and 2019, and the contract was extended for the first half of 2020.

To analyze the productivity that can be required of this factory for the first semester of 2020, the entity provided information on 21 projects developed in the first two years by the provider. This information contains, for each project developed, the year and semester in which it was developed, identifier, required effort, and functional size, measured in *CFP*[3] (Table 2). The functional size was acquired utilizing the *EPCU* approximation approach, as was defined by the Experts Guide for Early Software Sizing with COSMIC[17]; that is why the numbers are no integer.

Derived from the fact that the data provided can group the projects by the semester in which they were developed, the unit theory approach using lot costs is considered in this case study. Each semester contains a batch of certain number of functional size units.

Two concepts widely known in the Operations Research theory will be used to allow the modeling of the productivity of the provider: Productivity Factor (*PDR*) and Productivity[4].

The *PDR* represents how many Work Hours [WH] are required to develop a functional size unit [CFP], and its units are given by [WH/CFP]. In contrast, Productivity represents how many [CFP] are implemented by [WH], and its units are given in [CFP/WH], which is the inverse of the *PDR*.

It is worth mentioning that the approach to determine the *PDR* in the first two semesters was not entirely correct, since the entity, due to poor advice, defined that all the projects that were developed in this period of time would be estimated with a determined fixed *PDR* in 34 [WH/CFP], which was defined using expert judgment, that is, without formal support derived from the analysis of historical data as recommended by the best practices. For the second year two semesters, 2019, the COSMIC methodology was correctly implemented using the reference database of the Mexican Association of Software Metrics[2], so these projects have a variable *PDR*, then the acceptance of the estimates and, consequently, the validation of the *PDR* was developed by the entity based on the definition of an estimation validation process, such as that proposed in [16].

With the data in Table 2, the learning curve analysis will be performed using the *UT* to estimate cost per lot, considering each semester of development as one, so it is necessary to perform the *LMP* and *AUC* calculations as mentioned in section 3.2. Table 3 shows the data resulting from these calculations.

Figure 1 shows the curve generated using the data from Table 3, in the "Y" axis (*AUC*), the average cost of each unit per lot is presented. In the "X" axis (*LMP*), the midpoint of the lot was considered.

A transformation over the values of both axes is made using the Natural Logarithm (ln) function to make the data more linear in order to perform a linear regression[7]. The result values are shown in Table 3. When graphing ln(AUC) for the "Y" axis and ln(LMP) for the "X" axis, we obtain the graph shown in figure 2. It is important to note that the line on the graph has a negative slope, which shows the learning process.

[3] CFP: COSMIC Function points, the unit of measurement of the COSMIC method.

Table 2. Information provided on the projects developed by the software factory.

ID	Semester	WH	PDR	CFP
1	1	667	34.00	19.61
2	1	660	34.00	19.41
3	1	583	34.00	17.14
4	1	574	34.00	16.88
5	2	509	34.00	14.97
6	2	381	34.00	11.20
7	2	363	34.00	10.67
8	2	257	34.00	7.55
9	3	19157	18.28	1047.97
10	3	14187	18.10	783.81
11	3	7324	16.28	449.87
12	3	6731	16.07	418.85
13	3	5772	15.99	360.97
14	3	2495	15.89	157.01
15	3	1389	15.78	88.02
16	4	12666	15.50	817.16
17	4	2275	15.37	148.01
18	4	17276	15.32	1127.67
19	4	27407	15.23	1799.54
20	4	2335	14.97	155.97
21	4	6880	13.36	514.97

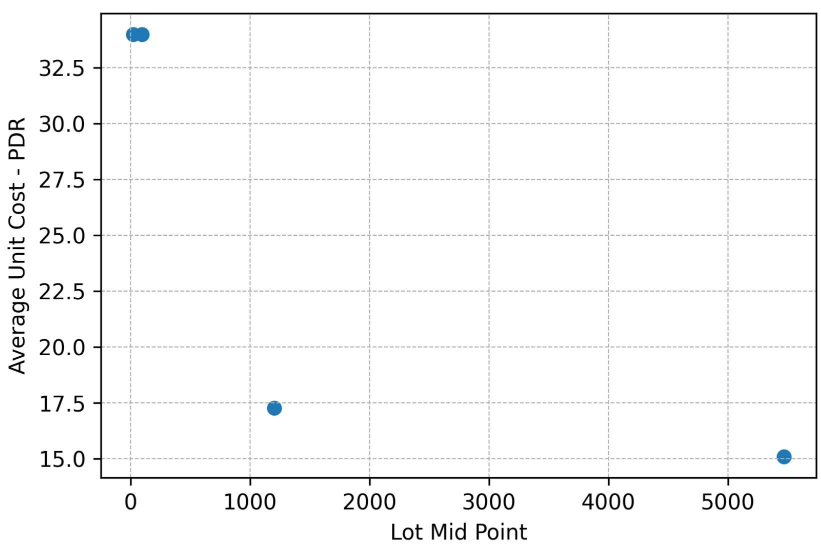

Fig. 1. AUC(PDR) vs. LMP.

Table 3. Result of the calculations made to analyze lot cost with Unit Theory.

Lot	Units [CFP]	WH	PDR (AUC)	First Unit	Cumulative	LMP	ln(AUC)	ln(LMP)
1	73.05	2484	34.00	1.00	73.05	24.35	3.52	3.19
2	44.41	1510	34.00	74.05	117.47	94.51	3.52	4.54
3	3306.53	57055	17.25	118.47	3424.00	1204.07	2.84	7.09
4	4563.34	68839	15.08	3425.00	7987.34	5468.27	2.71	8.60

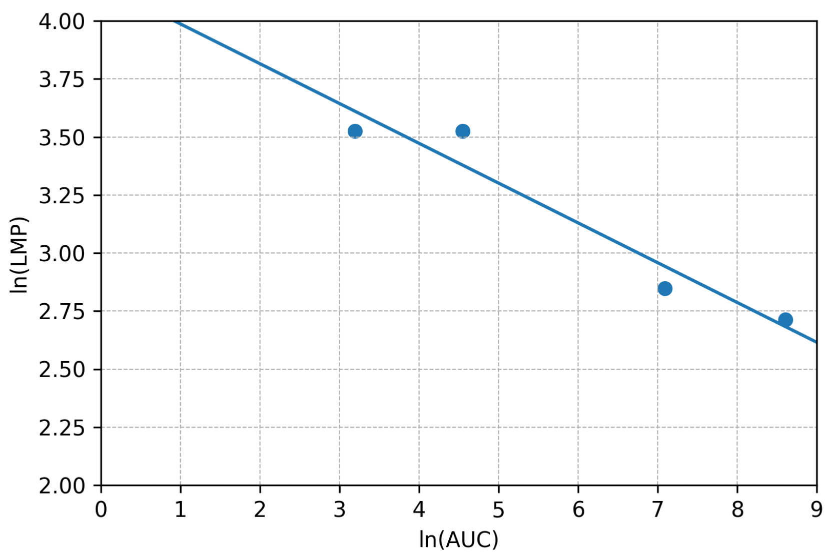

Fig. 2. ln(PDR) vs. ln(LMP) using the transformed data from Table 3.

The values of the constants that define the equation of the line contained in figure 2 can be found by performing a linear regression using the data in Table 3, which results in the following equation:

$$Y(x) = -0.171x + 4.1581 \tag{6}$$

Where:

- Y is the AUC
- x is the LMP

However, these results are found in units of natural logarithm (ln), since the equation that we really have is:

$$ln(PDR) = -0.171 * ln(LMP) + 4.1581 \tag{7}$$

To transform back this equation, the exponential function needs to be used in each side; as a result, the following equation:

$$PDR = 63.951 * LMP^{-0.171} \tag{8}$$

In this equation, we can see that the slope of the learning curve is:

$$2b = 2^{-0.171} = 0.887 = 88.7\% \tag{9}$$

This is the equation that best models the production environments for the data set in Table 2. This shows that there is a learning curve of 88.7%; in consequence, there is a productivity increase of 11.3% in each time it is doubled the amount of *CFP* developed.

Now it is possible to solve the equation to know the average cost per unit of any future lot using its *LMP* value, which can be obtained once the first and last unit are known.

As a validation exercise, the functional size to be developed during the contract extension was estimated. The average size of the projects developed in the four previous semesters was considered as the estimated value for the extended period. The value was 1997 [*CFP*].

From this data, the values for lot 5 (first semester of the year 2020) are calculated, which are the first and last unit, the lot midpoint (*LMP*), and equation 8 is used to calculate the *PDR* of the lot. Finally, since PDR = [WH/CFP], the amount of [WH] is given by multiplying the *PDR* [WH/CFP] by the functional size in [CFP].

Table 4 shows in column 4 the PDR for each lot from the past periods, while column 5 shows the estimated value for that lot using equation 8. In the last line, there is a lot number as 5, this corresponds to the first semester of 2020, in column 1 the average of [CFP] developed from the previous periods was considered as the last batch (1997 [CFP]) size, for this value the PDR estimated using the equation 8 also was calculated. In the last column (6) the Magnitude of Relative Error (MRE) was calculated considering the real value and the estimated value for the four initial lots, considering the real PDR (column 4) and the estimated PDR (column 5).

Table 4. Result of the calculations made to analyze lot cost with Unit Theory.

Lot	Units	WH	PDR	PDR Estimated	MRE
1	73.05	2484.00	34.00	36.99	0.08
2	44.41	1510.00	34.00	29.32	0.13
3	3306.53	57055.00	17.25	18.95	0.09
4	4563.34	68839.00	15.08	14.62	0.03
5	1997.00	26839.02	-	13.44	-

Using the column (5), the quality criteria for estimation was evaluated to analyze the robustness of the model, which are Mean Magnitude of Relative

Error (MMRE), MRE Standard Deviation (RMS) and the Prediction level at 25% (Pred 25%) shown in Table 5.

Table 5. Values of calculated quality criteria, MMRE, RMS, and Pred (25%).

Quality Criteria	Value
MMRE	0.08
RMS	2.91
Pred (25%)	4.00

Based on the information in Table 5, it can be mentioned that there is an average relative error of 8.8%, with a standard deviation of 2.913, and all the points are within the 25% prediction level. Observing the *MRE* value from Table 5, and considering the prediction level quality criteria, all the points are within the prediction level Pred(9.8%). That is, all the estimations present a relative error equal or below to 9.8%; in this sense, we can expect that the estimate for period 5 of the *PDR* to be used is 13.44 [WH/CFP] ±9.8%.

After graphing the values in Table 4, it is observed in figure 3 that the estimated values (green) for the known periods have the same behavior as the original values (blue), with an MMRE of 8.8% (Table 5).

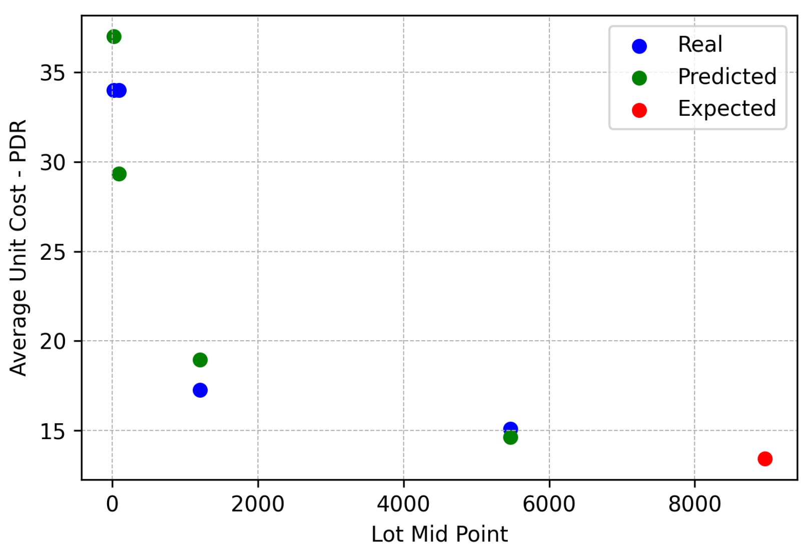

Fig. 3. In blue, the original values and in green those estimated by equation 8, also in red the expected value for the first half of 2020.

So we have a model that reliably represents the learning curve defined by equation 8, which has a learning rate of 88.7%. It is possible and achievable by the provider, since it was formally obtained using an analysis of Learning Curves and based on the observed productivity of previous development periods.

5 Conclusions

A problem faced by companies that have the same provider for several periods of time, is knowing formally what proportion of improvement could be expected because the know-how acquired during the service.

This paper presents a proposal to determine the learning ratio of a supplier in order to request an improvement rate of the productivity factor (PDR) based in the previous cycles through a real case study in the Mexican industry. The approach utilized to determine the degree of learning is the learning curve theory.

The analysis of Learning Curves for this case study has shown that there is learning throughout each period in which different projects were developed. This shows that productivity improves over time, showing a learning rate of 88.7%, representing a productivity increase of 11.3%.

With these results, we can estimate the effort in [WH] to produce a $[CFP]$ unit, that is, the PDR for the next batch, once we have the estimated size of the software to be developed in that period.

In order to estimate the expected functional size to be developed in the lot of the first semester of the year 2020, the average of the functional size from the lots referring to the years 2018 and 2019 was considered. The expected functional size for the first semester is 1997 $[CFP]$, using the equation 8, we can expect a PDR of 13.44[WH/CFP] $\pm$9.8%.

The case study presented has only few data to analyze, so the future work is to look for the bigger software projects data sets to repeat the analysis and compare the results.

6 Limitations

To analyze the productivity of a software development company, other variables must be included, such as personnel turnover and production interruptions[7], where the improvement factor is affected negatively.

References

1. The COSMIC Functional Size Measurement Method: Measurement Manual. v. 4.0.2 edn. (2017), http://www.cosmic-sizing.org
2. Asociación Mexicana de Métricas de Software. https://www.amms.org.mx/ (2020), [Online; accessed 5-June-2020]
3. COSMIC Sizing. https://cosmic-sizing.org/ (2020), [Online; accessed 5-June-2020]

4. Abran, A.: Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers. John Wiley & Sons, Hoboken, NJ, USA, 1 edn. (2015)
5. Flores, M.T., Lagarda, A.M.: Aprendizaje en microempresas de Baja California, pp. 95–116. Universidad Autonoma de Baja California (2011)
6. Jones, C.: Impact of Software Size on Productivity. ISBSG (2013)
7. Mislick, G.K., Nussbaum, D.A.: Cost Estimation Methods and Tools. Wiley, 1 edn. (2015)
8. Ralli, P., Panas, A., Pantouvakis, J.P., Karagiannakidis, D.: Comparative evaluation of learning curve models for construction productivity analysis. In: Panuwatwanich, K., Ko, C.H. (eds.) The 10th International Conference on Engineering, Project, and Production Management. pp. 347–358. Springer Singapore, Singapore (2020)
9. Runeson, P., Host, M.: Guidelines for conducting and reporting case study research in software engineering. Springerlink (2008). https://doi.org/10.1007/s10664-008-9102-8
10. Stewart, R.D.: Cost Estimating. John Wiley & Sons, Haarleem, Netherlands, 2 edn. (1991)
11. Team, C.P.: Cmmi for development, version 1.3. Tech. Rep. CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (2010), http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9661
12. Towill, D.R., Cherrington, J.E.: Learning curve models for predicting the performance of AMT, pp. 195–203. Springer-Verlag London (1994). https://doi.org/10.1007/BF01754598
13. Tuckman, B.W., Jensen, M.A.: Stages of small-group development revisited. Group & Organization Management **2**, 419 – 427 (1977)
14. Valdés-Souto, F.: Creating a Historical Database for Estimation Using the EPCU Approximation Approach for COSMIC (ISO 19761). Universidad Popular Autónoma de Puebla (UPAEP), México, Puebla, 4th edition of the international conference in software engineering research and innovation (conisoft'16) edn. (2016)
15. Valdés-Souto, F.: Impacto del Tamaño de Software en la Productividad para la Industria Mexicana de desarrollo de Software. Asociación Mexicana de Métricas de Software (AMMS), México, CDMX (2018)
16. Valdés-Souto, F.: Validation of supplier estimates using COSMIC method. CEUR Workshop Proceedings, Haarleem, Netherlands (2019)
17. Vogelezang, F. (ed.): Early Software Sizing with COSMIC: Experts Guide. The Netherlands, 2 edn. (February 27, 2020). https://doi.org/10.13140/RG.2.1.4195.0567
18. Vogelezang, F., Heeringen, H.v.: Benchmarking: Comparing Apples to Apples, pp. 205–217. Apress, Berkeley, CA (2019). https://doi.org/10.1007/978-1-4842-4221-6_18