Int J Digit Libr (2010) 11:141–153
DOI 10.1007/s00799-011-0073-9

DiCoMo: the digitization cost model

Alejandro Bia · Rafael Muñoz · Jaime Gómez

Published online: 1 October 2011

Abstract The estimate of digitization costs is a very difficult task. It is difficult to obtain accurate values because of the great quantity of unknown factors. However, digitization projects need to have a precise idea of the economic costs and the times involved in the development of their contents. The common practice when we start digitizing a new collection is to set a schedule, and a firm commitment to fulfil it (both in terms of cost and deadlines), even before the actual digitization work starts. As it happens with software development projects, incorrect estimates produce delays and cause costs overdrafts. Based on methods used in Software Engineering for software development cost prediction like COCOMO and Function Points, and using historical data gathered during 5 years at the MCDL project, during the digitization of more than 12000 books, we have developed a method for time-and-cost estimates named DiCoMo (Digitization Cost Model) for digital content production in general. This method can be adapted to different production processes, like the production of digital XML or HTML texts using scanning and OCR, and undergoing human proofreading and error correction, or for the production of digital facsimiles (scanning without OCR). The accuracy of the estimates improve with time, since the algorithms can be optimized by making adjustments based on historical data gathered from previous tasks. Finally, we consider the problem of parallelizing tasks, i.e. dividing the work among a number of encoders that will work in parallel.

Keywords Cost and time estimates · Digitization · Contents production · DL project management

This study is a substantially revised and extended version of a paper (with the title Estimating Digitization Costs in Digital Libraries Using DiCoMo) originally appeared in the Proceedings of the 14th European Conference on Digital Libraries (ECDL 2010).

A. Bia (✉)
Operating Research Center (CIO), Universidad Miguel Hernández de Elche, Elche, Spain
e-mail: abia@umh.es

R. Muñoz · J. Gómez
Department of Languages and Information Systems, Universidad de Alicante, Alicante, Spain
e-mail: rafael@dlsi.ua.es

J. Gómez
e-mail: jgomez@dlsi.ua.es

1 Introduction

Even after three decades since Barry Boehm presented the Constructive Cost Model (COCOMO) [1], the problem of accurately estimating software development costs is far from solved. In professional software development practice, just a few developers use software estimation methods other than expert judgement (which is basically an "expert's guess"), and when they do, the results are usually far from satisfactory [2,3].

This study discusses some of the reasons why cost estimation methods like COCOMO fail in practice in certain software-engineering applications, but may be accurate for other tasks, like predicting digitization times and costs, provided that we make the necessary modifications and customizations to the algorithm. By doing this, we have improved the accuracy of the estimates and widened its possible uses to other fields. Hence, we recommend the use of this type of algorithmic method for tasks other than software development. Later, we provide examples of production time and cost estimates obtained in this way at the MCDL[1] project [4,5].

[1] http://www.cervantesvirtual.com/.

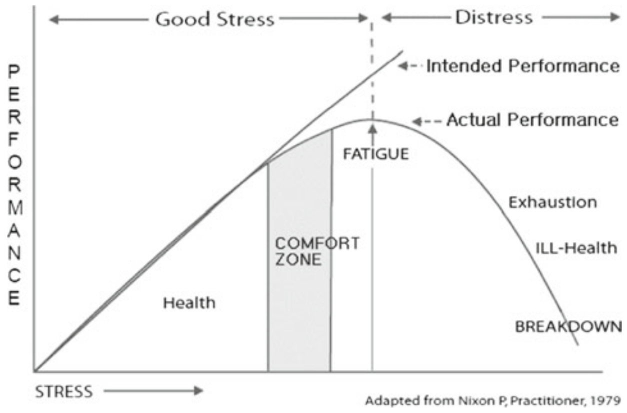


Fig. 1 Performance under Stress (Nixon's "Human Function Curve" [6])

1.1 Some previous remarks on the nature of time-and-cost estimates

Often we use the words prediction and forecast when referring to estimates. The nature and purpose of predictions and forecasts is different from estimates. In the case of predictions and forecasts (think of stock-exchange predictions or weather forecasts), we obtain some prediction values, and then wait for real events to happen and confirm the predictions, or not. In the case of an estimate, we should not wait for an event to happen, but instead, should work pro-actively to make it happen. This active nature is essential for profiting from estimates. An estimate is a target, a goal we have to fulfil, a reference or time frame to help us control our project. A good estimate is the time or cost objective within which a task **can be done** under **moderate pressure** (within the comfort zone) with **reasonably good quality**. A task can always be done in a longer time, or in a shorter time under exceptional stress, up to a point where it cannot be done (at least with the desired quality), or it causes burnout to team members (see Fig. 1). Therefore it is wise to think of estimates as **reasonable goals**, that will require some effort and control, and not as mere predictions. A good deal of risk management is also advisable to help accomplishing the estimated targets, without surprises.

1.2 Previous studies

The only algorithmic method we found [7] proposes the following linear formula to calculate the time required for texts: $\text{Hours}_{\text{textfile}} = 10 + X \cdot 0.412$, 10 being the fixed[2] training time in hours and X the number of text pages. Similarly, Bauer uses the following formula for images: $\text{Hours}_{\text{TIFF}} = 5 + Y \cdot 0.031$, 5 being the training time in hours, and Y the number of images.

Finally, they add the cost of hardware and software and consider an hourly wage for the operator, to calculate the final cost: $\text{Cost}_{\text{textfile}} = (\text{costofhardware}) + (\text{costofsoftware}) + (\text{hourlywage}) \cdot \text{Hours}_{\text{textfile}}$. A similar formula is used for the cost of scanning TIFF images.

Bauer also recognizes that the cost also depends on several other factors, like the age, condition, and font styles of the documents to be digitized, and proposes as a solution, to perform the digitization process on a small sample of one or two articles, and then adjust the formulas.

There are several interesting studies that provide useful data, good practice and recommendations, such as an early article by Simon Tanner, giving an insight to the problem [8], another early article by Steven Puglia, providing tables with data that can still be used as a reference [9], an article by Stuart Lee pondering the pros and cons of digitization and its associated costs [10], a handbook for *Assessing the Costs of Conversion* that was written on the basis of the experience gained in the large-scale digitization project performed by the Michigan University Library (US) [11], a comprehensive list of literature related to digitization costs within the recommendations of the European Union's MINERVA project (*Good Practices in Cost Reduction for Digitization*) [12], and a relatively recent report on digitization costs by Hammond and Davies (*Understanding the Costs of Digitization*) [13]. There are some on-line tools for digitization cost estimation like the RLG worksheet for estimating digital reformatting costs [14] or the Presto-Space preservation project cost calculator [15].

Finally, there are several reference studies on time/cost estimates for software engineering projects which, in spite of being from a different field, may serve as a source of inspiration and comparison for digitization projects [1,16–26].

1.3 Methodology used

We first started to collect digitization metrics (time spent and special features of the task) by means of forms that the operators/encoders filled for every task assigned to them. Later, we implemented these data collection forms into the workflow system for better accuracy and automatic handling of the data, and added new fields for checking special features that appeared to be related to the time required to complete the task. In this sense, good expert knowledge of the digitization processes helps us to detect the factors that affect digitization times (see Sect. 3 below), and to adjust the estimate equations.

[2] Bauer states that "the time necessary for training is fixed: that is, it does not vary with the number of articles to be produced", but neither says that training time should not be considered for subsequent digitization projects, nor takes into account the level of expertise of the operator/encoder, which is expected to improve with time.

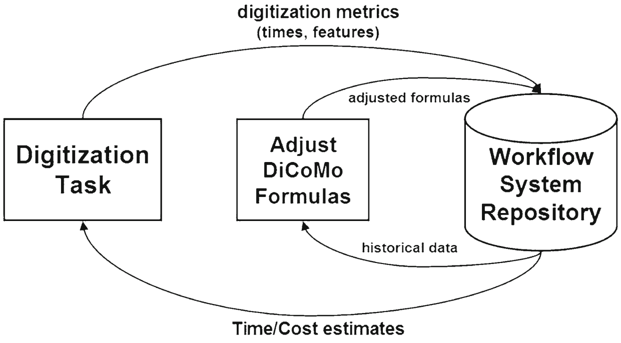


Fig. 2 Iterative adaptive methodology to continuously adjust the DiCoMo formulas

With the first 100 values, we obtained the first DiCoMo curves, and with subsequent metrics, we detected and then adjusted the modifiers.

Iteratively, therefore, we adjust the curves and the modifier factors based on past metrics. We use the new adapted model to predict the next bunch of books, which in turn will add new metrics to the collection, that will help us to re-adjust the estimation model. This iterative process is repeated until the model settles (the model is predicting the new tasks reasonably well). Whenever there are changes in the processes, technology, etc., or we feel the estimates are not adequate, the model can be adjusted again to the new conditions. In this way, the cost equations are dynamically improved by re-adjusting the parameters with the new data fed back from recently finished projects (see Fig. 2).

We have tested the model in several occasions with samples of about 100 books each time, and the estimated time required to process the whole lot was often between 90 and 105% of the real time required, but remember that encoders know the target estimate beforehand, and always try to meet their deadlines. To collect new data for statistical verification, we should have had to perform new tasks blindly, i.e. without any estimate given in advance to the encoders, because if the encoders have a target estimate, they will try to fulfil it, and then the statistical verification would be somehow biased. This is why we made "some previous remarks on the nature of time-and-cost estimates" at the beginning of the article (Sect. 1.1). In our opinion, the best way to know if an estimate is adequate, is to measure the stress applied to the team. If a "task can be done under moderate pressure (within the comfort zone)", then the estimate is adequate. In this sense, the estimates done with DiCoMo were adequate.

The model was developed originally to solve a practical need of the library, and the iterative adaptive method was very effective in this sense, as the library was urging us to provide estimates as soon as possible.

2 The basic digitization cost model

In the digitization cost model we proposed [27], we use an equation similar to Intermediate COCOMO [1], but with some differences:

- **Size-independent overhead**. We added a new term called *Size-independent overhead (SIO)* that represents the preparatory work for the task, which is independent from its size. Examples of this SIO are the time needed to adjust the parameters of an image scanner and OCR before starting a scanning session, and the disassembling/unbinding of a document to separate the pages before scanning. This is a fixed time which does not depend on the number of pages to be processed.
  In the case of digitization of fragile materials, preparatory processes are very important. They might involve different preparatory tasks like conservation and even restoration. We did some digitization of delicate books of this kind, like the Sacred Bible of Cocentaina[3] which dates back to thirteen and fourteen centuries, but the number of these cases was not sufficient to draw conclusions or obtain data to adjust the formulas. To make matters worse, each of these digitization projects presented particular problems of its own, which made them inadequate for generalization. However, if the need arises, special preparatory tasks can be estimated based on "expert judgement", and the time be added as another independent term to the formula. Note that the SIO is meant for preparatory tasks which are *independent* from the volume of the material to process (page restoration, for instance, does not fall into this category).
  We must warn at this point that DiCoMo is meant for massive production of digital contents, where tasks are repeated under similar conditions. Tasks that are not routinely performed, and special tasks like page restoration, are not candidates for this kind of estimation model which is based on collected historical data of similar cases.
- **The size is known beforehand**: One of the reasons why COCOMO often fails in estimating software costs is because its calculations are based on an estimated size of the code to be built, measured in Kilo Lines Of Code (KLOC, i.e. thousands of lines of program code), which is highly uncertain at the initial stages of the project. When applying a similar method to estimate digitization costs, the first thing we realize is that we do not have to guess the size of the work because we can easily know it, or can accurately estimate it. The size of the documents to digitize is measured as the number of pages P and can

[3] http://www.cervantesvirtual.com/s3/BVMC_OBRAS/ff5/e81/ec8/2b1/11d/fac/c70/021/85c/e60/64/mimes/ff5e81ec-82b1-11df-acc7-002185ce6064_810.htm.

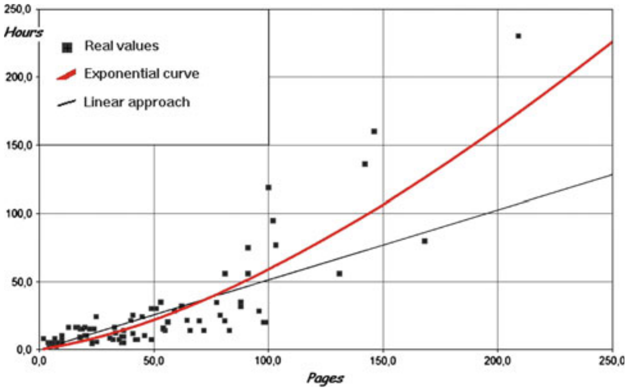


Fig. 3 Document digitization times (hours vs. pages) using Basic DiCoMo (exponential curve)

be measured (or calculated with reasonable accuracy) beforehand.

- **Time is cost**: There is one similarity with software development projects: since most of the cost in digitization is human labour, which in the long run overweighs the cost of the hardware and software used, time estimates of a digitization task can be directly converted to cost estimates, using some cost factor (amount of money paid per hour of work).

Given the number of pages P, we can directly calculate the time in hours T, with the Basic-DiCoMo formula:

$$T = a \cdot P^{b} + \mathrm{SIO} \qquad (1)$$

For a graphical example of this DiCoMo approach, see Fig. 3, where an estimation curve (thick line) approaches real data spots (black squares) that represent time measures of real digitized documents. The thin straight line represents a linear approach to the spots ($a \cdot P$), which is not the best approach, while the curve ($a \cdot P^{b}$) fits more accurately, although not perfectly. The values for a and b are obtained by adjusting the curve to best approach the cloud of points (historical data), using the least-squares method. The value of the fixed term SIO is the point at which the curve crosses the Y axis, or the time needed for the impossible case of a task of size 0, and, for our purposes, represents the preparation time mentioned before. For certain tasks (e.g. big tasks), this time may be negligible and hence ignored.

For example, the following equation, based on early experience at the MCDL project, gives us the estimated number of hours to process a text given the number of pages:

$$T = 0.069 \cdot P^{1.465} + 0.6 \qquad (2)$$

Using this formula, a standard-complexity book of 100 pages will take about 59 h of scanning, correction and XML markup altogether.

2.1 The importance of historical cost-data:

Inside most organizations, the estimation of production costs is usually based on past experiences. Historical data are used to identify the cost factors and to determine their relative importance within the organization [28]. Historical data will be used first to adjust the basic estimation algorithm (the exponential curve), and later to adjust the detected impact factors to be used as modifiers to obtain more accurate results. This is the reason why it is so important to systematically collect and store time and feature data from projects, and to take note of the perceived factors that affect the times as well as the amount of this impact.

3 Adjusted DiCoMo (the advanced model)

The simple approach used in equation 1 does not take into account the fact that different literary works have different degrees of difficulty owing to several factors (discussed later), which will affect production times. We have detected the most important of these factors, and assigned weights to them to enable us use them as feature-modifiers. We added an effort adjusting factor (EAF) to the adjusted DiCoMo equation, equivalent to the one used in intermediate-COCOMO, but based in this case, on specific digitization features. The EAF is calculated as the multiplication of relevant feature-modifiers chosen from a table (see, for instance, Table 1). The modifiers shown in the table were obtained from historical data collected at the MCDL project. The value of these modifiers is 1.00 in the normal case, then having no impact on the overall EAF factor, or values slightly above or below 1 in the other cases, contributing to raise or lower the unadjusted estimate, producing the desired "adjust" effect (see vertical lines from the exponential curve to the white-filled triangles in Fig. 4).

Table 1 DiCoMo: complexity modifiers used to calculate the EAF

Modifier	Low	Normal	High
Encoder experience and skills	1.30	1.00	0.70
Familiarity with task	1.20	1.00	0.80
Familiarity with computer tools	1.20	1.00	0.80
Foreign/ancient languages present	–	1.00	1.25
Stained or old paper	–	1.00	1.15
Old font faces	–	1.00	1.15
Special care required (old books)	0.80	1.00	1.20
High quality demands	0.80	1.00	1.20
Inadequate technology used	0.80	1.00	1.20

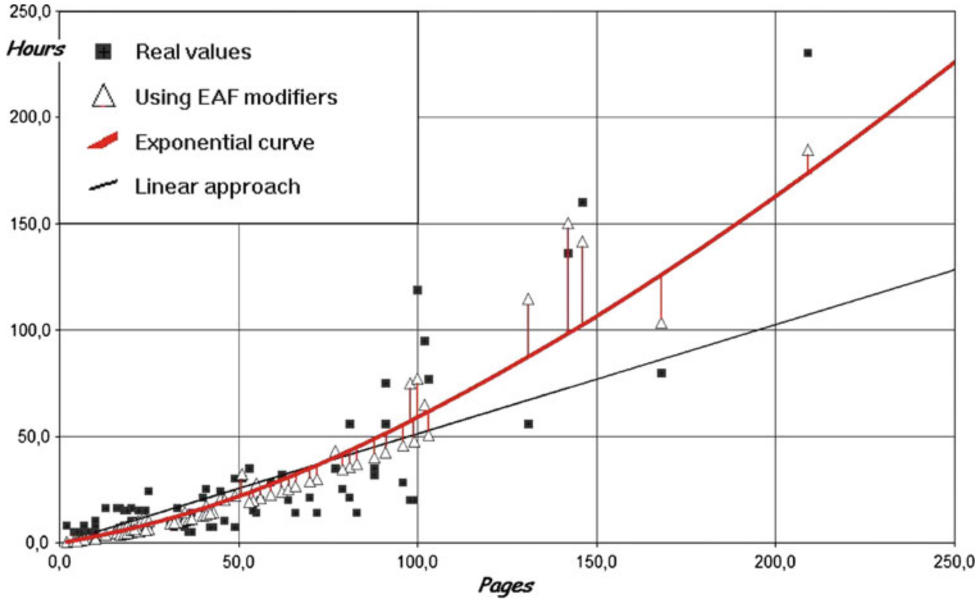


Fig. 4 Document digitization times (hours vs. pages) using Adjusted DiCoMo (triangles represent adjusted values)

$$T = a \cdot P^b \cdot \mathrm{EAF} + \mathrm{SIO} \qquad (3)$$
$$\text{where: } \mathrm{EAF} = \prod \mathrm{modifier}_i$$

3.1 Factors that affect digitization costs

There are several factors that affect the cost of production of digital objects. Both these factors and their effects on costs are difficult to determine and have to be carefully studied. They are detected by experience, as features which are found to affect the time required to complete a task either positively or negatively. Once a factor of this type is detected, we have to measure its impact, as a percentage relative to the “normal-case” time. The best way to do this is by gathering time records of digitization tasks, and record also their particular features and their weights (e.g. low, normal, and high). With enough records of this type, algorithm optimization techniques can be applied to infer the range of impact of a given feature as a +/- percent. For instance, we detected that the literary style of a text affected its digitization time, because of harder or simpler markup requirements. We started recording this feature, indicating whether a text was mainly (from the best to the worst case): prose, verse, drama written in prose or drama written in verse. We stored records of these together with the times required to complete the digitization task. After gathering 300 records, we used optimization techniques to get the optimum value range for this new modifier, which turned out to be +/- 7,6%. Therefore, in the case of drama written in verse (hardest markup case), we will have to add 7,6% more time to the estimate. Among the factors detected, we can highlight the individual skills and experience of the persons assigned to the project, as well as their familiarity with the specific characteristics of the work to be digitized, the familiarity with the computer tools to be used, the complexity of the task, size, quality requirements, technology used, etc. Also important are some features of the document that affect digitization times, such as: the presence of foreign or ancient languages, stained/yellowish paper, old/irregular font faces, high quality demands, inadequate technology used, special care required for old books, etc. See figs. 5 and 6 for examples of difficult texts for OCR that would require extra proofreading and correction time[4].

The main factors detected, which influence digitization costs are

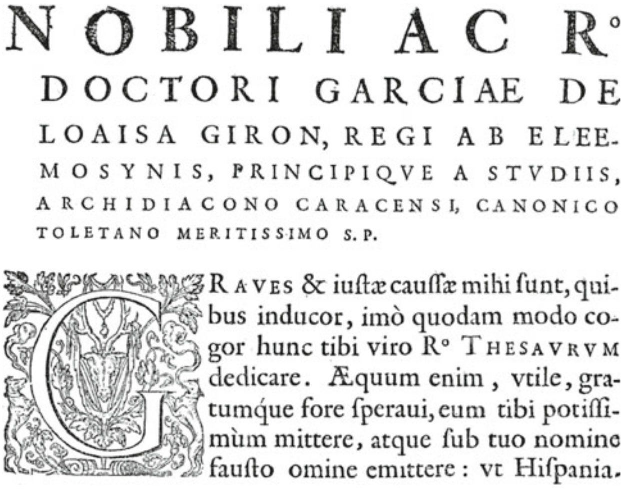
- Volume of the material to publish (already included as the variable of the formula)
- Individual skills of scanner-operators, correctors and encoders
- Complexity of the task
- Special quality requests
- Technological infrastructure of the working environment

A list of the complexity modifiers used for calculating the EAF is shown in Table 1.

The Adjusted DiCoMo equation (3), customized with historical data from previous projects (4), and using the EAF factor, now gives us better estimates of the time needed to digitize a text given the number of pages:

$$T = 0.081 \cdot P^{1.462} \cdot \mathrm{EAF} + 0.1 \qquad (4)$$

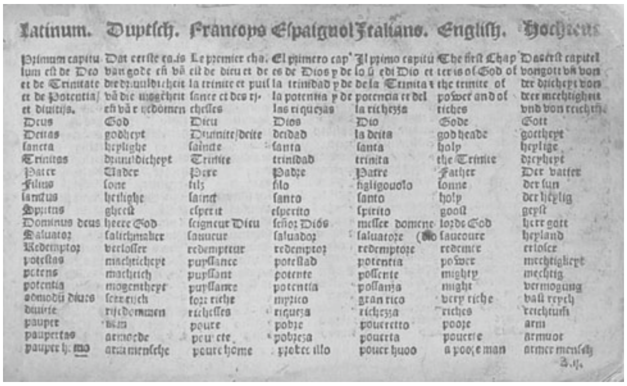
[4] See online at: http://adrastea.ugr.es/tmp/_webpac2_1097709.22421 and http://www.lluisvives.com/FichaObra.html?portal=10\&Ref=9552.



NOBILI AC R°
DOCTORI GARCIAE DE
LOAISA GIRON, REGI AB ELEEMOSYNIS, PRINCIPIQVE A STVDIIS,
ARCHIDIACONO CARACENSI, CANONICO
TOLETANO MERITISSIMO S.P.

GRAVES & iuſtæ cauſſæ mihi ſunt, quibus inducor, imò quodam modo cogor hunc tibi viro R° THESAVRVM dedicare. Æquum enim, vtile, gratumq́ue fore ſperaui, eum tibi potiſſimùm mittere, atque ſub tuo nomine fauſto omine emittere: vt Hiſpania.

Fig. 5 An example of problematic page for OCR featuring irregular font sizes, alignment and spacing, special characters and ligatures, Latin language and a graphic embedded. Extracted from *Abrahami Ortelii Antuerpiani, Thesaurus geographicus, recognitus et auctus* (1596). Bib. Universidad de Granada



Latinum.	Duytſch.	Francoys	Eſpaignol	Italiano.	Engliſh.	Hochteuts
Primum capitu	Dat eerſte ca.is	Le premier cha.	El primero cap'	Jl primo capitu	The firſt Chap	Das erſt capitel
lum eſt de Deo	van gode eñ vã	eſt de dieu et de	es de Dios y de	lo ſi edi Dio et	ter is of God of	von gott vñ von
et de Trinitate	die driuuldicheit	la trinite et puiſ	la trinidad y de	de la Trinita &	the trinite of	der driheyt von
et de Potentia/	vã die mogẽtheit	ſance et des ri·	la potentia y de	potencia et del	power and of	der mechtigheit
et diuitijs.	eñ vã r ycdõmen	cheſſes	las riquezas	la richezza	riches	vnd von reichtũ
Deus	God	Dieu	Dios	Dio	Gode	Gott
Deitas	godheyt	Diuinite/deite	deidad	la deita	god heade	gottheyt
ſancta	heylighe	ſaincte	ſanta	ſanta	holy	heylige
Trinitas	driuuldicheyt	Trinite	trinidad	trinita	the Trinite	dreyheyt
Pater	Vader	Pere	Padre	Patre	Father	Der vatter
Filius	ſone	filz	filo	figligouolo	ſonne	der ſun
ſanctus	heylighe	ſainct	ſanto	ſanto	holy	der heylig
Spiritus	gheeſt	eſperit	eſperito	ſpirito	gooſt	geyſt
Dominus deus	heere God	ſeigneur Dieu	ſeñor Diós	meſſer domene	lorde God	herr gott
Saluator	ſalichmaker	ſauueur	ſaluador	ſaluatore	ſauioure	heyland
Redemptor	verloſſer	redempteur	redemptor	redemptore	redeemer	erloſer
poteſtas	machtichheyt	puyſſance	poteſtad	potentia	power	mechtigkeyt
potens	machtich	puyſſant	potente	poſſente	mighty	mechtig
potentia	mogentheyt	puyſſance	potentia	poſſanza	might	vermogung
admodũ diues	ſeer rijck	fort riche	muy rico	gran rico	very riche	vaſt reych
diuitie	rijcdommen	richeſſes	riqueza	richezza	riches	reichtum
pauper	arm	poure	pobre	poueretto	poore	arm
paupertas	armoede	pourete	pobreza	pouerta	pouertie	armuot
pauper hõmo	arm menſche	poure homme	pobre hõbre	pouer huõo	a poore man	armer menſch

A.ij.

Fig. 6 Another example of problematic page for OCR and also for facsimile production featuring irregular styled fonts, yellowish paper with stains and several languages. Extracted from *Septem linguarum Latinae, Teutonicae, Gallicae, Hispanicae, Italicae, Anglicae, Almanicae, dilucidissimus dictionarius mirum quam utilis nec dicam necessarius omnibus linguarum studiosis*... (ca. 1535). MCDL (Lluis Vives portal)

For instance, a book of 100 pages with stained/old paper (+15%) and foreign or ancient languages present (+25%), will take approximately 98 h to complete, compared to the 59 h estimated using the basic equation without modifiers: $T = 0.081 \cdot 100^{1.462} \cdot 1.15 \cdot 1.25 + 0.1 = 97.85\,\text{h}$

Figure 4 shows the Basic DiCoMo exponential curve (thick line) that approaches the black square data spots that represent measures of real digitized documents.The EAF-adjusted results are shown as white-filled triangles which in most cases approach more closely the real values. In a very few cases, however, the EAF results are worse than the basic curve.

The time assigned affects mainly the quality of the product obtained which is notably reduced when the times assigned are unreasonably short, forcing the technicians to work under excessive pressure. This is particularly true for the correction-and-editing process, where text output from OCR has to be carefully proofread and corrected. This is a delicate craft that takes time and cannot be done under excessive pressure. When not properly done, further revisions and corrections are needed, with a very negative impact on costs. Next, each one of these factors is described in detail.

3.2 Volume of the material to publish

Digitization projects, compared to software development projects, have the advantage that we can know quite precisely beforehand the size of the work to be done (namely the number of pages or words to digitize). On the contrary, in software development projects, the number of lines of code is not known at the beginning of a project. This is the main drawback of the original COCOMO method, which was modified and renamed as COCOMO-II [17–19] to overcome this problem. Other methods, like Function Points (FP) [20,21], Use Case Points (UCP) [22,23] and Predictive Object Points (POP) [24], which are based on functionality aspects instead of lines-of-code, do not have this problem.

There are various ways to measure the volume of the material to digitize. The first and the easiest way to determine the raw size of a text to be digitized is to count the pages. This is the most common method, and is generally sufficient for accurate estimation purposes. A disadvantage is that pages are not equally dense for all books. We can have an approach to the density by counting the words that fit in a standard page, or the words that fit in a fixed size window, and then assuming that the rest of the pages are similar in this respect. To count individual words would be more accurate (we verified this by experience), but it is not a practical approach: the improvement in accuracy does not justify the effort. However, after the OCR process takes place, we will obtain a text file, with errors, but nevertheless a text file where we can automatically count the number of words or get the size in bytes. This is a good measure of the size of the proofread and correction work that follows, and may serve to adjust the initial estimates for higher accuracy.

3.3 Effort adjusting modifiers

There are many complexity factors that affect every stage of the digitization process (scanning, proofreading or correction, and markup). In the case of the correction stage, which we consider the most critical one, there are various factors to be taken into consideration:

- the type of text: prose, verse, drama (both written in prose, and in verse), dictionary;
- footnotes (if they are used very frequently);
- quotations in foreign or classical languages (when they are very frequent);
- the complexity of the author style and vocabulary;
- the quality of the OCR output (few or lots of errors);
- the legibility of the original (paper copy from which the digital version id produced).

Concerning markup, complexity varies according to the number and difficulty of the tags to be added. Drama, for instance, with the need of a *castlist*, *speaker* and *speeches*, require an additional amount of tagging compared to prose.

Verse with split lines is another good example of extra complexity, since special care needs to be taken to assign attribute values which indicate which part of the split line of verse is which (initial, middle, or final).

In the case of the production of digital facsimiles from manuscripts, a case of particular complexity occurs when we have to work on rare and valuable originals that have to be handled with special care (wearing rubber gloves for instance) and using a digital photographic camera instead of a flat-bed scanner. On the contrary, digitizing unbounded pages using a flat-bed scanner with automatic page feeder would be the easiest case.

For each of the critical features mentioned, three possible modifier values were set, to be used when the features appear as high, normal, or low (e.g. the values could be 1.10, 1.00 and 0.90 in each case). For a given task, all the modifier values that apply to the case and that are different than normal (1.00), are multiplied to obtain the EAF factor.

3.3.1 Individual skills of the technicians

In the programmer's world, individual productivity has been measured extensively. Harold Sackman et al. carried out an experiment in 1968 [29]. They found evidence that performance differences registered in individual programmers were much bigger than those attributed to the effect of the working environment. The difference between the best and the worst performances was very high, the experience being a decisive factor. In a later experiment, Sackman observed a variation in the productivity of as much as 16:1. DeMarco and Lister also discussed the effects of a well-integrated group to enhance productivity in their book Peopleware [30] that deals with the human component in software projects.

In digitization, the results that we have measured comparing correctors' performances show remarkable differences in productivity, depending on their individual skills and experience (sometimes a 3:1 ratio). Variations in productivity of this magnitude are significant for cost estimates, making it necessary to express this in the calculations by means of a modifier.

The skill factor is not static: when performing the same task repeatedly people usually learn to do it more efficiently, following a learning curve. To face this fact, the *encoder experience and skills* factor chosen for a particular team of inexperienced operators/encoders may change for future projects, to reflect the newly acquired skills.

3.3.2 Special quality requirements

In the case of digital text production, producing a modernized digital edition from an ancient text consumes additional time and effort compared to processing a modern text, since modernization is a complex task that involves difficult decisions.

Using Madison markup for the transcription of a manuscript is another example of additional requested complexity. So is the case of making highly legible digital facsimiles from ancient manuscripts, where special care and fine-tuning of the scanning equipment may be required, as well as graphical post-processing.

3.3.3 Technological level of the environment:

This is a relevant issue when using different technologies or migrating from old to new production tools. When the environment is stable and well known, and the estimation equations are well adjusted for it, there is no need to bother about this issue. Changes in technology, however, will surely require modifications to the equations, and may make historical time and cost data obsolete for future estimate adjusting purposes.

3.4 Procedure to estimate costs using DiCoMo

1. Establish the production process to follow (production workflow) (see figure 7 for an example). There may be different production workflows for different purposes (e.g. facsimile images are only scanned, while text undergoes scanning, OCR, proofreading and markup).
2. Identify all the objects (books, images, etc.) to be digitized and their associated tasks (Work Breakdown Structure).
3. Measure or estimate the size of each object to be digitized.
4. Establish the production stages to be followed by assigning the right workflow.
5. Specify the effort adjusting factors for each object.
6. Calculate the time each unit will take (use the adequate equation with the corresponding complexity factors).
7. Calculate the total development time for the project as the sum of individual times.

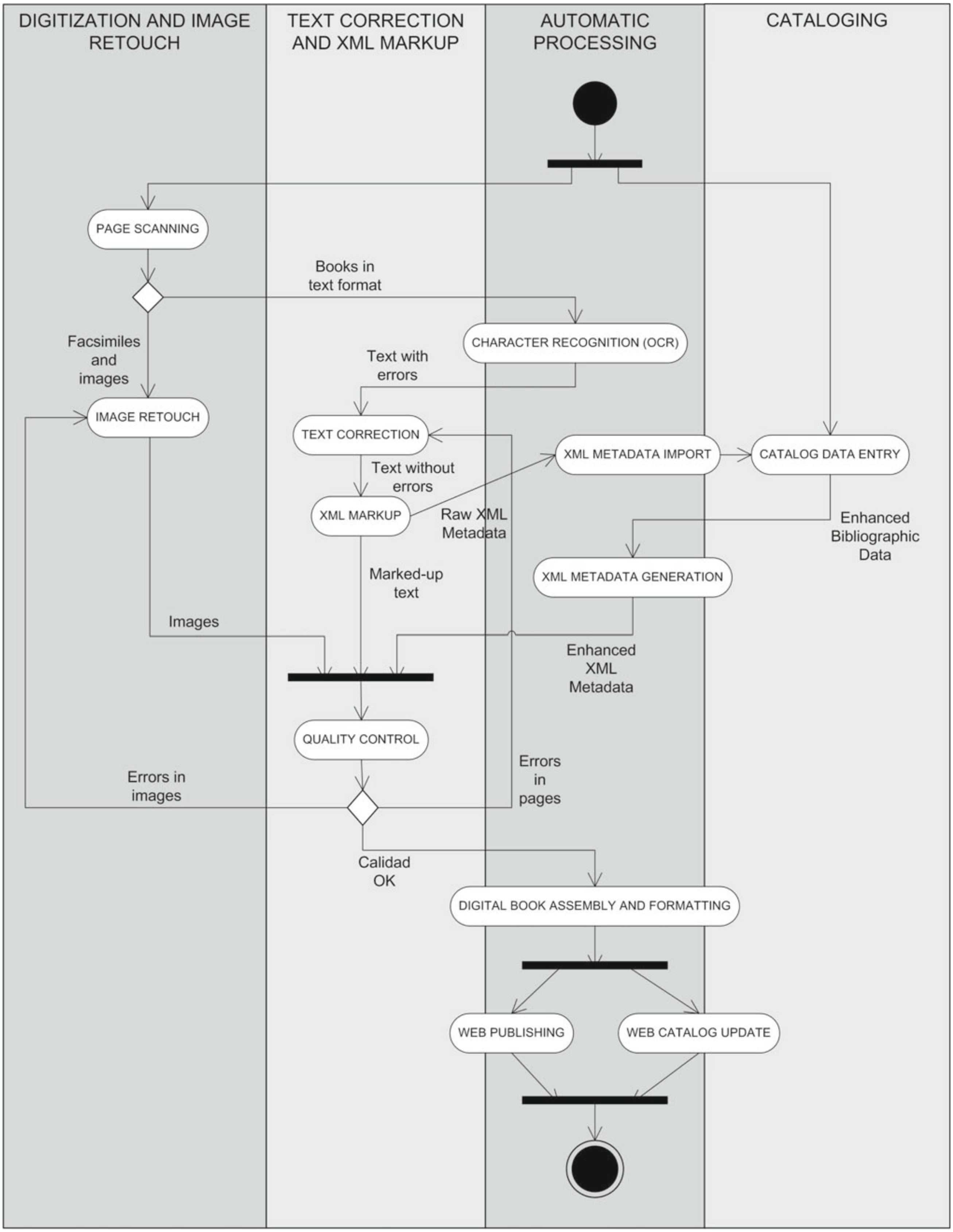


Fig. 7 Content production workflow shown as a UML activity diagram

8. Optionally, compare the estimate with another, perhaps a top-down one like the *DELFI* technique or *expert-judgement*, identifying and correcting the differences in the estimate, if necessary.

3.5 Estimating by digitization stages

In previous examples, we have used a single formula to estimate the whole digitization pipeline (several digitization tasks altogether), which is simpler to do, but better results can be obtained by using specific formulas with their respective adjustments for each stage in the digitization process (e.g.: cataloguing, scanning, proofreading and correction, markup).

Many studies have attempted to relate size-oriented methods like COCOMO and function-oriented methods like function-points [21]. We take from the *function-points* model the idea of modularization according to functions, and from the COCOMO II model [17] the estimates by project stage.

Therefore, in this case we consider each production stage of the digitization pipeline as a functional unit to be performed independently, to which a specific estimation equation is applied. The global estimate T turns out to be the sum of all the specific estimates for each of the stages:

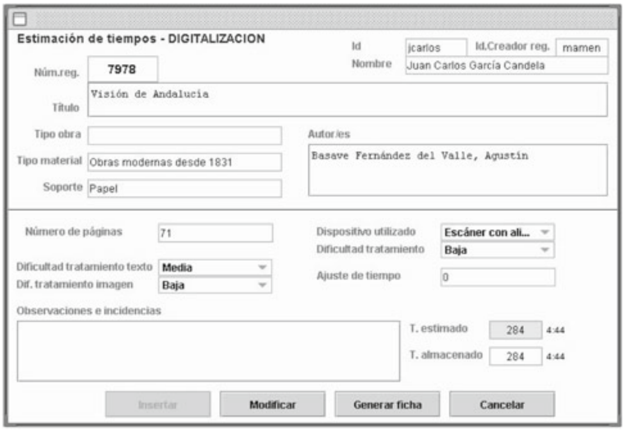
$$T = \sum_{\text{stages}} \left(a \cdot P^b \cdot \prod \mathrm{eaf}_i + \mathrm{SIO} \right) \qquad (5)$$

We sum the times of each stage under the assumption that they are performed sequentially. In all cases, this is correct if we want to calculate the total work-time or *time cost* (TC), but may not be so, if we want to calculate the *time span* (TS) of the project, when some tasks are performed in parallel (see Sect. 4).

3.6 Implementation

The DiCoMo method was implemented into the digital library's workflow and document-handling system, a software application that controls the whole production process of all the types of digital resources produced by the digital library (see Figs. 7, 8, 9, 10 and 11). It provides useful management information for estimating costs and times of digitization projects. It estimates times of cataloguing, scanning, correction and markup in the case of text production, and cataloguing, scanning, and graphical processing in the case of facsimiles.

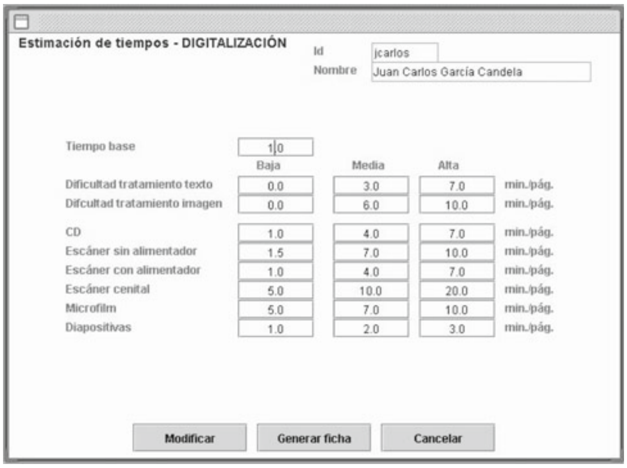
A few screenshots captured from this system are shown later. Figure 8 shows a scanning-only estimate for a 71 page book. Figure 9 shows average historical values for different types of complexities and types of scanning device. Figure 10 shows a correction-only estimate, and Fig. 11 shows the final summary of costs for the production of a digitized text book.


Estimación de tiempos - DIGITALIZACION

Id: jcarlos
Id.Creador reg.: mamen
Nombre: Juan Carlos García Candela
Núm.reg.: 7978
Título: Visión de Andalucia
Tipo obra:
Tipo material: Obras modernas desde 1831
Soporte: Papel
Autor/es: Basave Fernández del Valle, Agustín

Número de páginas: 71
Dificultad tratamiento texto: Media
Dif. tratamiento imagen: Baja
Dispositivo utilizado: Escáner con ali...
Dificultad tratamiento: Baja
Ajuste de tiempo: 0
Observaciones e incidencias:
T. estimado: 284 4:44
T. almacenado: 284 4:44

Fig. 8 Estimate of scanning costs

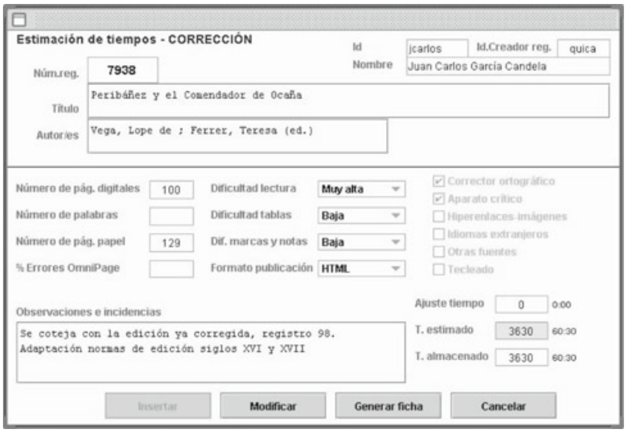

Estimación de tiempos - DIGITALIZACIÓN

Id: jcarlos
Nombre: Juan Carlos García Candela

Tiempo base: 1.0

	Baja	Media	Alta	
Dificultad tratamiento texto	0.0	3.0	7.0	min./pág.
Difcultad tratamiento imagen	0.0	6.0	10.0	min./pág.
CD	1.0	4.0	7.0	min./pág.
Escáner sin alimentador	1.5	7.0	10.0	min./pág.
Escáner con alimentador	1.0	4.0	7.0	min./pág.
Escáner cenital	5.0	10.0	20.0	min./pág.
Microfilm	5.0	7.0	10.0	min./pág.
Diapositivas	1.0	2.0	3.0	min./pág.

Fig. 9 Parameters used to estimate digitization costs


Estimación de tiempos - CORRECCIÓN

Id: jcarlos
Id.Creador reg.: quica
Nombre: Juan Carlos García Candela
Núm.reg.: 7938
Título: Peribáñez y el Comendador de Ocaña
Autor/es: Vega, Lope de ; Ferrer, Teresa (ed.)

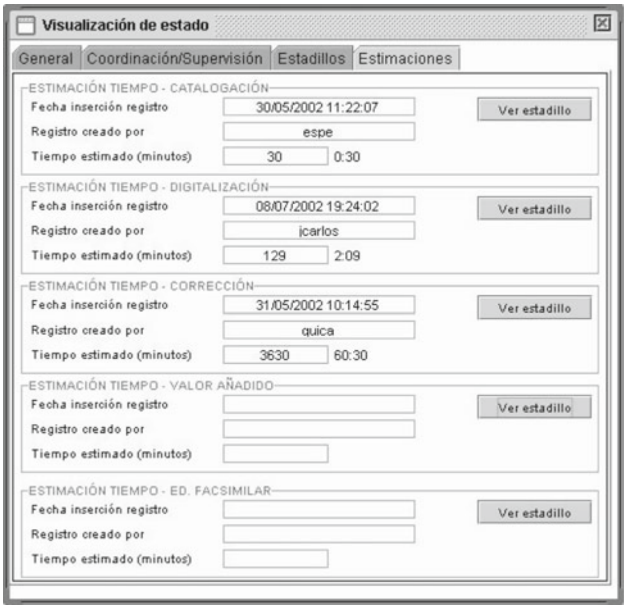
Número de pág. digitales: 100
Número de palabras:
Número de pág. papel: 129
% Errores OmniPage:
Dificultad lectura: Muy alta
Dificultad tablas: Baja
Dif. marcas y notas: Baja
Formato publicación: HTML

☑ Corrector ortográfico
☑ Aparato crítico
☐ Hiperenlaces-imágenes
☐ Idiomas extranjeros
☐ Otras fuentes
☐ Tecleado

Observaciones e incidencias:
Se coteja con la edición ya corregida, registro 98.
Adaptación normas de edición siglos XVI y XVII

Ajuste tiempo: 0 0:00
T. estimado: 3630 60:30
T. almacenado: 3630 60:30

Fig. 10 Estimate of correction costs



Visualización de estado

ESTIMACIÓN TIEMPO - CATALOGACIÓN

Fecha inserción registro	30/05/2002 11:22:07
Registro creado por	espe
Tiempo estimado (minutos)	30 0:30

ESTIMACIÓN TIEMPO - DIGITALIZACIÓN

Fecha inserción registro	08/07/2002 19:24:02
Registro creado por	jcarlos
Tiempo estimado (minutos)	129 2:09

ESTIMACIÓN TIEMPO - CORRECCIÓN

Fecha inserción registro	31/05/2002 10:14:55
Registro creado por	quica
Tiempo estimado (minutos)	3630 60:30

ESTIMACIÓN TIEMPO - VALOR AÑADIDO

Fecha inserción registro	
Registro creado por	
Tiempo estimado (minutos)	

ESTIMACIÓN TIEMPO - ED. FACSIMILAR

Fecha inserción registro	
Registro creado por	
Tiempo estimado (minutos)	

Fig. 11 Final report of digitization costs for a book (shows cataloguing, scanning and correction of the text)

4 Parallelizing tasks

Now we consider the problem of splitting some digitization task (e.g. correction and markup) amongst N encoders that will work in parallel, assigning to each of them P/N pages, from a total of P pages. Based on equation 3, we made the following modifications:

The TS is the time from start to finish, and for simplification, it will be the time of any of the parallel subtasks (assuming they are all equally long, i.e. P/N pages):

$$TS = a\left(\frac{P}{N}\right)^b \cdot \prod \mathrm{eaf}_i + \mathrm{SIO} + (N-1)\cdot \mathrm{SJ} \tag{6}$$

where SJ is the *split–join* time, which is the time needed to split and later rejoin the material of the N workers, a new task derived from working in parallel. We think it is sound to assign a small time to each split and join, and multiply it by the number of split–joins, which is $N-1$.

For an example, we will use the following values: $a = 0.081$, $b = 1.462$, $\mathrm{EAF} = \prod \mathrm{eaf}_i = 1.00$ (for simplification), $\mathrm{SIO} = 0.1$ and $\mathrm{SJ} = 0.25$.

Then, for a correction and markup task of a book of 100 pages, the time required by one person is 68 h, and the time span of two persons working in parallel is 25 h (less than a half), and if we used five encoders, it would take 7.5 h to complete the task (see Fig. 12).

This time span does not reflect the actual cost of the task, but only the time needed to complete it. The TC, which is the sum of the times spent by all the N encoders working in parallel, will tell us how many person-hours will have to be paid, and is equivalent to approximately N times the *time span*[5] (see Figs. 13, 14):

$$TC = N\left[a\left(\frac{P}{N}\right)^b \cdot \prod \mathrm{eaf}_i + \mathrm{SIO}\right] + (N-1)\cdot \mathrm{SJ} \tag{7}$$

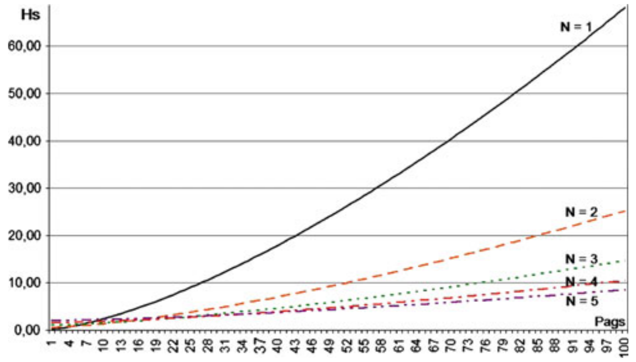


Fig. 12 Parallelizing a task: this graphic compares the time it takes to complete a task (time span) of correction and markup of documents of different sizes, when the task is done by N persons working in parallel (*curves* shown are for N varying from 1 to 5)

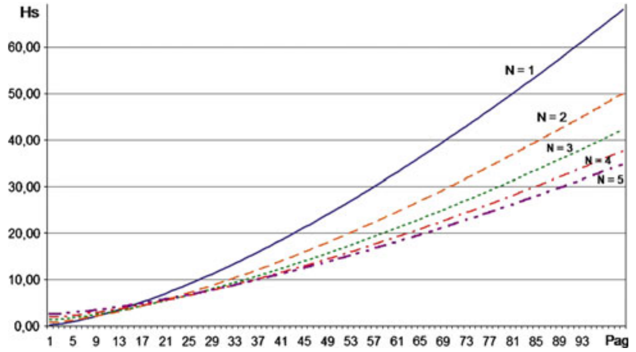


Fig. 13 Parallelizing a task: this graphic compares the total time spent (time cost) of correction and markup of documents of different sizes, when the task is done by N persons working in parallel (*curves* shown are for N varying from 1 to 5)

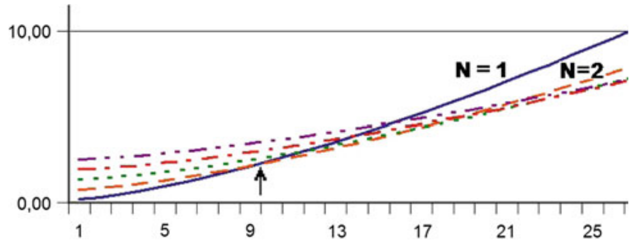


Fig. 14 Parallelizing a task: detailed view of Fig. 13, that shows the crossing points of the curves (see *arrow*). This example shows that for a book of more than nine pages, it is more convenient to split the task in two. The *arrow* marks the intersection of the curves for $N = 1$ and $N = 2$

[5] It is not exactly N times, because of the S–J time necessary to split and rejoin the task parts, this being a task that cannot be parallelized.

For instance, for a correction and markup task of a book of 100 pages, the time required by one person is 68 h, the total time required by two persons working in parallel is 50 h, and the total time needed by five encoders, would be 34 h (see Fig. 13).

Finally, we can calculate the actual cost (C) in a given currency by multiplying the TC by a cost factor (CF) which is the amount of money paid per hour of work:

$$C = \mathrm{TC} \cdot \mathrm{CF} \tag{8}$$

We conclude that although we have one *preparation task* time (SIO) for each encoder, and a new *split–join task* (SJ) required for parallelization, all of which increase the final cost, for a number of pages big enough, the total work-time or TC and hence the actual cost C, are smaller when we have encoders working in parallel. Moreover, the time to have things done, or *time span* is remarkably smaller when we parallelize, since the time curve is exponential (e.g. dividing the work in two, will take less than half the time required by one person). Figure 14 shows the intersection points of the time curves for different number of parallel workers. These intersections indicate the number of pages from which it is more convenient to split, than not to do so.

4.1 Applying Amdahl's law to parallel tasks

According to Amdahl's-law [31], the speed-up of a program using multiple processors in parallel computing is limited by the time needed for the *sequential fraction* (the part that cannot be parallelized) of the program. Although this law is meant for parallel processors, we are going to apply it to the problem of parallel tasks performed by humans. Note that one of the differences between computer processor work and human work, is that the performance (amount of work done per time) is linear in the case of a processor, while we have already observed that human work roughly follows an exponential curve: humans get tired[6], while computer processors do not.

Amdahl's law can be expressed as

$$S = \frac{1}{(1-F) + F \cdot I} \tag{9}$$

where
S is the overall speed-up factor of the whole process obtained after parallelization of one task;
F is the original fraction of time of the task to be parallelized relative to the whole process;
I is the fraction of time of the parallelized task compared to the same task done sequentially, e.g. 0,5 means half the time after parallelization, i.e. twice as fast.

For example, consider a typical digitization sequence of the following three tasks:

1. cataloguing (0.5 h);
2. scanning and OCR (2.16 h);
3. text correction and markup (68 h)

F in this case is $\frac{68}{0.50+2.16+68} = 0.962 \quad \text{(i.e. } 96\%\text{)}$;
I is the new time for the third task using two encoders divided by the old time using one encoder: the new time is 25 h (as seen above in section 4), so $I = 25 \div 68 = 0.368$;

Therefore, the expected speed-up factor will be

$$S = \frac{1}{(1-0.962) + 0.962 \cdot 0.368} = 2.55 \tag{10}$$

If we split the correction-and-markup task in two parts, the whole three stage process will be 2.55 times faster (or will take 39% of the original time).

Another way to get to this result is to calculate the original total time ($0.50 + 2.16 + 68 = 70,66$ h) and the new total time ($0.50 + 2.16 + 25 = 27,66$ h), and divide the two.

The reason why the improvement is so high, is because we have chosen to parallelize the longest task (text correction and markup).

5 Conclusions

We have developed a cost estimation model for digitization projects based on known software-engineering cost models. This method allowed us to obtain reasonable estimates of the time required to complete digitization tasks[7]. Digitization projects, compared to software development projects, have the advantage that the size of the work to be done can be known beforehand (namely, the number of pages or words to digitize). In software design, we can only have a "guess" of the total number of lines of code or modules that a project will require, and the accuracy of the calculated time estimate will depend largely on this preliminary "expert judgement estimate".

[6] Several factors are believed to be responsible for the slowdown of human performance on big tasks, like the sheer complexity of big tasks, boredom, reduction of vigilance (the state of readiness to respond) [32], fatigue, procrastination (giving priority to smaller, simpler tasks), discouragement, or the additional coordination, communication and planning required. To the question of "why don't operators work as fast as possible?", Kieras and Meyer conclude that 'some operators had optimized their performance with respect to energetic, "ergonomic", or fatigue-based criteria in their task strategy, rather than maximizing their performance speed' [33].

[7] By *reasonable estimate*, we refer to a target time-span that allows us to complete the task under reasonable pressure.

We verified that the algorithmic model we proposed works well in practice, and can be easily applied to different digital-production processes, or be adapted to other kinds of project tasks, provided that the cost equation is fine-tuned for each type of task using historical data. This requires two things to be done in advance:

– To collect sufficient historical data to fine tune the parameters of the cost equation.
– To identify the critical factors that affect the time required to do the task, and calculate and assign adequate effort adjusting modifiers to each of them.

With this information, a cost-equation for a specific production process can be easily obtained.

We have also studied the problem of splitting the work amongst several workers working in parallel, obtaining equations to calculate the time span, the total *person-hours* to be paid, and the optimal number of workers for a given digitization task size (see Figs. 13, 14).

Concerning task parallelization, we realized that for the biggest tasks, it is convenient to have various encoders working in parallel, and we provide some clues on the size of the task and the number of parallel workers recommended. We also realized in practice that operators/encoders were more productive when working for shorter periods of time, like 4 h a day, instead of 8.

As future work on digitization-cost estimates, we expect to be able to apply DiCoMo to other digitization projects to improve or confirm the modifiers of the formula. We are also willing to cooperate with interested partners in digitization-cost research, and to apply DiCoMo to new and different tasks, apart from the ones shown here.

References

1. Boehm, B.W.: Software engineering economics. Prentice Hall, Englewood Cliffs (1981)
2. Magazinovic, A.: Exploring cost estimation inaccuracy: why do practitioners still fail to predict the actuals? Technical report, Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden (2008)
3. Galorath, D.: Software project failure costs billions... Better estimation and planning can help. http://tinyurl.com/Galorath (2008)
4. Bia, A., Pedreño, A.: The Miguel de Cervantes Digital Library: the Hispanic Voice on the Web. LLC (Literary and Linguistic Computing) J (Oxford University Press) **16**(2), 161–177 (2001)
5. Bia, A: The use of multimedia to enhance the accessibility of digital library resources: The multicultural-scope of the services offered by the Miguel de Cervantes digital library project. In: Anderson, J., Dunning, A., Fraser, M. (eds.) Digital resources for the humanities 2001 and 2002: an edited selection of papers, Office for Humanities Communication, vol. 16, pp. 1–11. King's College, London (2003)
6. Nixon, P.G.: The human function curve. Practitioner pp. 765–769; 935–944 (1976)
7. Bauer, K.: Cost analysis of a project to digitize classic articles in neurosurgery. J. Med. Libr. Assoc. (JMLA) **90**(2), 230–234. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC100769/ (2002)
8. Tanner, S., Smith, J.L.: Digitisation: how much does it really cost? In: Digital resources for the humanities, King's College, London (1999)
9. Puglia, S.: The costs of digital imaging projects. RLG DigiNews **3**(5). http://chnm.gmu.edu/digitalhistory/links/cached/chapter3/link3.10b.digitalimagingcosts.html (1999)
10. Lee, S.D.: Digitization: is it worth it?. Computer Libraries **21**(5), 28–31. http://www.infotoday.com/cilmag/may01/lee.htm (2001)
11. UMich-MoA: Assessing the costs of conversion: Making of America IV: the American Voice 1850–1876. http://www.lib.umich.edu/files/services/dlps/moa4costs.pdf (2001)
12. Winer, D.: Good practices in cost reduction for digitisation: resources for minerva and minerva plus WG on good practices. http://www.minervaeurope.org/structure/workinggroups/goodpract/costreduction/documents/wp6costreduction0904.pdf (2004)
13. Hammond, M., Davies, C.: Understanding the costs of digitisation: detail report. http://www.jisc.ac.uk/media/documents/programmes/digitisation/digitisation-costs-full.pdf (2009)
14. Research Library Group: RLG worksheet for estimating digital reformatting costs. http://www.oclc.org/research/activities/past/rlg/digimgtools/rlgworksheet.pdf (1998)
15. Presto-Space: Preservation project cost calculator. http://digitalpreservation.ssl.co.uk/hosted/d13.2/newcalc.php (2007)
16. Putnam, L.H.: A general empirical solution to the macro software sizing and estimating problem. IEEE Trans. Software Eng. **SE-4**(4), 345–361, This article introduces the SLIM method (1978)
17. Boehm, B.W., Clark, B.K., Horowitz, E., Westland, C., Madachy, R., Selby, R.: Cost models for future software life-cycle processes: COCOMO 2.0. In: Arthur, J., Henry, S. (eds.) Annals of software engineering special volume on software process and product measurement, vol 1, pp. 45–60. J.C. Baltzer AG, Science Publishers, Amsterdam, The Netherlands (1995)
18. Clark, B.K., Devnani-Chulani, S., Boehm, B.W.: Calibrating the COCOMO II post-architecture model. In: 20th international conference on software engineering. Center for Software Engineering, Computer Science Department, University of Southern California, Los Angeles (1998)
19. CSE COCOMO II model definition manual: Center for software Engineering, Computer Science Department, University of Southern California, Los Angeles (1997).
20. Albrecht, A.J.: Measuring application development productivity. In: Proceedings of the Joint Share/Guide/IBM Applications Development Symposium pp.83–92 (1979)
21. Albrecht, A.J., Gaffney, J.E.: Software function, source lines of code, and development effort prediction: a software science validation. IEEE Trans. Software Eng. **SE-9**(6), 639–648 (1983)
22. Banerjee, G.: Use case points, an estimation approach (2001)
23. LCI: *Use cases and function points*. (Longstreet Consulting Inc., Blue Springs 2004)
24. Minkiewicz, A.F.: Measuring object oriented software with predictive object points. PRICE Systems, LLC (1997)
25. Valerdi, R.: The constructive systems engineering cost model (COSYSMO). Phd thesis, University of Southern California. http://csse.usc.edu/csse/TECHRPTS/PhDDissertations/files/ValerdiDissertation.pdf (2005)
26. Salvetto-de-León, P.F.: Modelos automatizables de estimación muy temprana del tiempo y esfuerzo de desarrollo de sistemas de información. Phd thesis, Departamento de Lenguajes y Sistemas Informáticos e Ingeniería de Software, Universidad Politécnica de Madrid. Supervisors: Francisco Javier Segovia-Pérez, Juan Carlos Nogueira-de-León. http://oa.upm.es/367/1/PEDROSALVETTOLEON.pdf (2006)

27. Bia, A., Muñoz, R., Gómez, J.: Estimating digitization costs in digital libraries using DiCoMo. Lectur Notes Comput. Sci. **6273**, 136–147 (2010)
28. Fairley, R.E.: Software engineering concepts. McGraw Hill, New York (1985)
29. Sackman, H., et al.: Exploratory experimental studies comparing online and offline programming performance. Communications of the ACM **11**(1) (1968)
30. DeMarco, T., Lister, T.: Peopleware, productive projects and teams. Dorset House Publishing, New York (1987)
31. Amdahl, G.: Validity of the single processor approach to achieving large-scale computing capabilities. In: AFIPS conference proceedings pp. 483–485 (1967)
32. Ballard, J.C.: Computerized assessment of sustained attention: a review of factors affecting vigilance performance. J. Clin. Exp. Neuropsychol. **18**(6), 843–863 (1996)
33. Kieras, D.E., Meyer, D.E.: The role of cognitive task analysis in the application of predictive models of human performance. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.39.2570&rep=rep1&type=pdf (1998)