UDC 007.5:004.02
Original Scientific Paper

Method for Estimating the Complexity of Designing Business Information Systems

Patrizia Poščić *patrizia@inf.uniri.hr*
University of Rijeka
Department of Informatics

Mile Pavlić *mile.pavlic@ris.hr*
University of Rijeka
Department of Informatics

Neven Vrček *neven.vrcek@foi.hr*
University of Zagreb
Faculty of Organization and Informatics Varaždin

Abstract

The most used method for determining the complexity of information systems is the function point method. This paper illustrates the new method, Data on Document (DOD) for estimating the complexity of designing business information systems. Similar methods are shown in brief. The objective is to explain the reasons for using the newly defined DOD method instead of the function point method. The new method is easier to use while the usage of function point method takes a lot of time and effort, and is more complex. The DOD method is compared to the function point method. A high degree of correlation was determined in observed information systems. The DOD method helps in early stages of the development of information systems, when it is not possible to apply the function point method.

Keywords: Information system, complexity estimation, function point, documents, data, correlation

1. Introduction

Information systems (IS) development projects are more and more complex. Project leader has to deliver a finished project in agreed timeframe and expenses. Often, this is not possible due to bad judgment about the needed time and other resources for the project. Even though organizations put a lot of money into developing IS, hoping this will better their operationability and strategic position, many fail in their intent. To return the investment put into information technology (IT), organizations first need to improve their ability to deliver IS development projects on time. The complexity of IS design stems from the fact that, besides a technological basis, it implies and covers business processes, and reflects on the final project goal, time, costs and quality. Furthermore, the design complexity reflects on managing human resources, such as a team, users and top management support.

Still too little money and time is invested into measuring the complexity of IS development, and what is invested mainly applies to measuring the complexity of the software. Even the function point method [7] has a primary focus in measuring the complexity of software itself. The question is whether the complexity of a project can be estimated before the very „end" of software product? Here, the emphasis is on estimation, because we cannot measure something that doesn't exist. Many authors have dwelled upon the problem of estimating the complexity [1], [3], [6], [11], [14], and this paper is a small contribution in the area.

A question emerges; why is the estimation important? A possible answer is that estimation can help in making a good business decision, i.e. determining the project feasibility

in a given timeframe, expenses and functional limitations, either coming from the user or the resources. Furthermore, the estimation enables an organization to avoid work which results in no financial gain, i.e. to choose work which it wants to perform and which overlaps with its strategic goals. Good estimations help to avoid highly risky work which can damage the reputation and subsistence on the world market. [3]

Estimation in the development and offer of software product is a tool that helps in the decision making process and ensures information needed for defining the offer and negotiation with the users. It is assumed that a larger quantity of different measuring elements (like in the function point method) results in a better estimation. However, processing a large quantity of input data is time-consuming and demanding, and sometimes a quick estimation of software development is required. Since the function point method is the most used method for measuring the complexity of software development, but its usage demands a lot of knowledge of the method itself, as well as its application and defining basic concepts, there have been attempts to simplify it and use it in a quicker manner. [11]

The application of the function point method demands a full, detailed level of documentation, like a functional specification of the software system being measured. There are at least two situations in which, instead of using the FP method, it would be better to apply a method for the estimation of functional size [9], [11]. The first happens when the development project is in an early stage where the application is simply impossible. The FP method requires a set of data which can be obtained only after a detailed analysis. The second situation arises when there is a need for evaluating the existing software product, but there is no corresponding documentation or time and resources for executing correct measurement of function points.

2. Existing Methods for Estimating the Complexity of Software System Development

Methods for estimating software system development can be generally put into two groups. These are Direct Estimation Methods and Derived Estimation Methods [11].

Direct Estimation Methods are also known as Expert Opinion Methods. These imply the cooperation of one or more experts which directly estimate required elements of the estimation of function points, basing their estimation on experience and intuition. Derived Estimation Methods or Algorithmic Model Methods provide with the estimation of complexity as a function of more variables which relate to certain attributes of software development project. Generally, these methods are in correlation with the Decomposition Method [2]. By decomposing the project into smaller parts – subprojects, it is possible to conduct the evaluation part by part, in detail.

The basic difference between direct and derived estimation methods is that in the latter, estimation isn't performed directly on function point values, but rather on different project parameters which are somehow related to function point values.

According to the aforementioned categorization of direct and derived estimation methods, one of the most known direct estimation method is the Delphi or Shang method [8] according to which the predictions of a number of experts are combined.

Estimations based on analogy also fall into the category of direct estimation methods [10]. There is the simple analogy method and structural analogy method. Simple analogy implies finding similar projects in the past; based on these projects a fast estimation of a new project can be performed. Structural analogy is a “finer” method and it is actually based on the comparison between the new project and one or more earlier projects.

One of the most known derived estimation methods is the early function point method [15]. It combines different estimation approaches with the aim of ensuring a better estimation of software system size. It uses analogical and analytic functionality and data classification. Analogy enables spotting similarities between the new software “part” and similar “parts” in other application software, which are already classified according to the method. The analysis ensures a certain stability in the estimation, because the weight of certain software objects is

not attributed through empirical examination of the data gathered on the current project, but is based on conceptual grounds, i.e. connected to the way certain software objects in the classification structure are built. The aim of the method is early prediction on the functional size, which can be used for preliminary technical and managing decisions.

Of course, a precise, standard measuring always has to be conducted in a later stage in order to confirm the validity of a reached decision. Key elements of the method are: macro functions, functions, micro functions, primitive functionalities and logical sets of data [10], [15]. Notionally speaking, primitive functionalities respond to elementary processes in the standard analysis of function points, i.e. external inputs, external outputs and external inquires. Macro functions and micro functions are different aggregations of more primitive functionalities on different level of detail. Logical sets of data respond to standard logical files, without emphasizing the difference between "external" and "internal" data. A set of function point values (minimal, average and maximum number) is joined to each method object and these values are summed up in order to obtain a number of unadjusted function points. Value adjusted factor (VAF) is defined like in the standard function point method. Estimations reached by this method can be "detailed", "medium" or "concise", depending on the chosen level at an early stage of functionality classification. The reliability of function point method is directly proportional to the ability to recognize application components as a part of one of the described classes. The method has shown reliable and the results were within ± 10% of real values of FP in most cases, while in the terms of saving time (and expenses) it is within 50 and 90% in relation to a corresponding measuring method [11].

There are many others estimation methods and it is impossible to describe them all in one place, however, some of these should be mentioned: Use case points [6], COCOMO method (COnstructive COst Model) [5] and Noun-Verb Count - NVC method [1].

All mentioned methods are originally defined for the estimation of software application complexity; however, these can also be applied for estimating the complexity of information system design.

The majority of mentioned methods demand a complex calculation including numerous factors, conditions, specifics and assumptions. For their correct application a lot of time and effort is required. It is essential to define five functional elements: Internal Logical File - ILF, External Interface File - EIF, External Input - EI, External Output – EO and External Inquiry - EQ (FP method and similar methods), cost and scale drivers (COCOMO), who are the participants and which are the ways of use (use-case) etc. Only the NVC method differs, mostly by the time invested in its application. It demands no further criteria definitions; by reading out user demands the number of nouns and verbs is recorded. Authors Ahm and Baker claim it is difficult to prove that the results of other methods are better and more acceptable.

3. Estimation of Design Complexity Based on Quantity of Data on Documents

The method proposed in this paper is primarily assigned to the estimation of IS design complexity. The method concentrates on measuring the quantity of data on documents, and is accordingly called the Data on Documents method (DOD method). During the method definition process the main aim was to simplify the estimation and, by doing that, speed up the process of estimation. The point was to obtain certain indicators, as simply as possible, which could be compared to function points, since the FP method is the most widespread and the most accepted method.

3.1. Estimation Elements in DOD Method

In the process of defining estimation criteria, a lot of thought was given to the function of an information system, i.e. answering the question what all information systems have in common. The purpose of every system, including information system, is to result in needed information based on input data and its processing. Years of experience in developing information systems resulted in understanding that the quantity of documents and the

complexity of a given document inside a business system somehow determine the complexity, not only of designing, but also developing an information system. In most cases, to fully understand them, more time and effort is needed for systems with numerous documents, than for those with fewer documents. Proportionally, the project development is more complex and time-consuming.

When documents are concerned, we can divide these into two groups: basic (original) documents, and classic reports documents (derived, calculable, summary, preview documents).

DOD method is based solely on measuring one element, and that is the number of data on basic documents. Basic documents are those which connect the environment to the system or the sole processes in the system.

Classic report documents mostly contain data which belong to some other document or are calculated from previously familiar data.

Some basic documents in business systems can be put into this category, based on the preceding report description. These are mostly documents which serve as an output to the environment (like an insurance policy created out of the insurance offer, note of receipt created out of supplier's note of delivery, a receipt created out of the order, etc.). The data on mentioned documents is copied without any changes or with little change. For example, a decreased quantity on the note of receipt, document date, added data from coding table etc. according to the proposed method, such documents are not considered classic report documents and they enter the analysis for the complexity estimation by DOD method. Unlike such outputs from the IS, there are so called classic reports which are just a view to the condition of data in the data base, and their function is to inform or manage a business system (various statistic reports…) These do not belong to the group of basic documents of a business organization. It is difficult to say how many reports like these are in a business system, since there are continuous inquires for new reports with the same or different data, according to the same or different criteria. Therefore, classic reports are not considered in the proposed method and are not counted.

Except classic reports, the DOD method does not count or analyze documents like various regulations and laws which determine system operations. The basis of the method is data, and processes over this data do not interest us in this method. All processes represent a "black box". What are important are process inputs and outputs, and the algorithm itself is ignored.

The sole number of documents in a system cannot be sufficient criteria for the estimation of design complexity because these cannot significantly differ. Each document needs to be analyzed, but not in detail like in the process of its modeling, but it is necessary to count different kind of data on document. Each data on document needs to be counted, no matter whether this data is repeated on more documents. If we would track the repetition of data on documents, we would draw closer to data modeling methods eliminating redundancy and increasing estimation time, which is not our aim here.

Therefore, according to DOD method, there are two criteria for the estimation of IS development complexity: document and the number of data on document.

Since the aim of the proposed DOD method is to estimate the complexity of developing business information systems in a quicker and simpler manner, weight values are not assigned to criteria, but every observed data on document has the same weight factor. It is hard to determine the weight of a given data at the very start of the project, i.e. its influence on further complexity of the project. We could possibly consider the importance of a certain document for the system, and based on that, assign a weight to a certain document. The same applies to data on documents. Assigning weight factors to documents and data on them would have a few drawbacks like: an increased estimation time, the need for estimation experts and possible erroneous estimation.

3.2. Measuring Criteria Comparison for External Inputs for FP and DOD Methods

Unlike the proposed method, the function point method has exact determined values for every measured element. These values are joint to the elements, depending on the number of

counted, i.e. primary elements (DET, RET or FTR elements[1]). Tables 1 and 2 illustrate the way the complexity of an identified element is determined, in this case external input [7].

File Type Referenced (FTR)	Data Element Type (DET)		
	1-4	5-15	≥16
<2	Low	Low	Medium
2	Low	Medium	High
>2	Medium	High	High

Table 1. The degree of influence of DET and FTR elements on external input (EI)

	Degree of influence		
Elements	Low	Medium	High
Internal logical files - **ILF**	7	10	15
External interface files - **EIF**	5	7	10
External inputs - **EI**	3	4	6
External outputs - **EO**	4	5	7
External queries - **EQ**	3	4	6

Table 2. IFPUG[2] table of unadjusted function points

Let us further consider the problem of assigning weight factors. The input screen of a document represents an external output, according to the function point method. According to Table 1, if the screen has 16 fields and at least 2 FTPs, its complexity is high and the corresponding factor of complexity is 6 (Table 2). If the screen shows a document input that has a larger number of attributes, e.g. more than 50, the same factor of complexity emerges. But, is this really the same level of complexity? Is there a difference between entering 16 or 76 data items? It is obvious that a limit must be set, but where? Figures 1 and 2 graphically illustrate the complexity of an external input according to the function point method and DOD method, depending on the number of fields on a document (screen) with a constant number of FTRs.

[1] DET – Data Element Type; RET – Record Element Type; FTR – File Type Referenced

[2] International Function Points Users Group

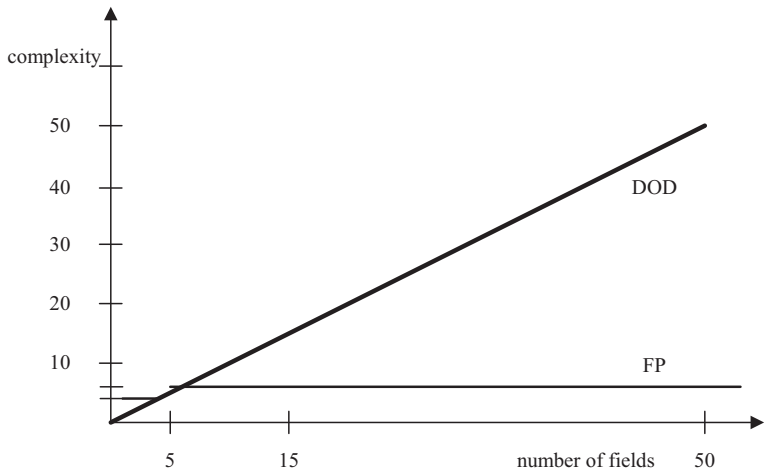

Figure 1. The complexity of an external input with 3 FTRs and a larger number of fields according to FP method and DOD method

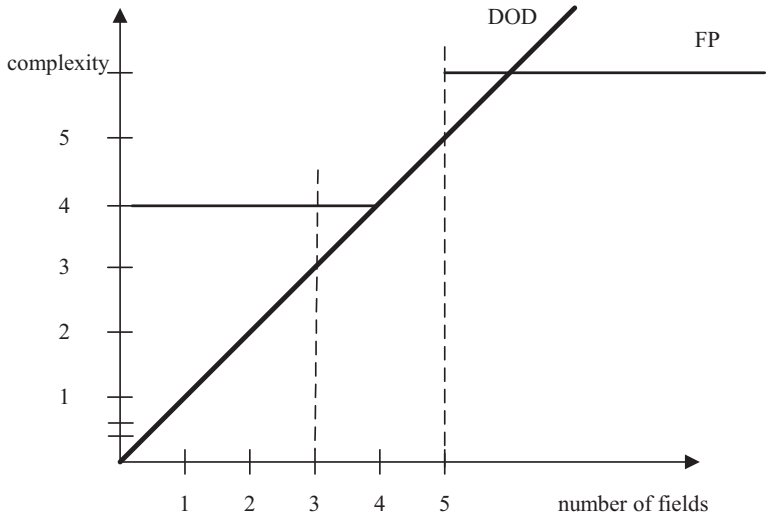

Figure 2. The complexity of an external input with 3 FTRs and a lesser number of fields according to FP method and DOD method

One can observe that in the case of documents (screens) with a large number of fields, the FP method significantly lowers complexity. That is, with 5 fields and 50 fields, the complexity is the same. On the other hand, DOD method increases document complexity proportionally to the number of fields. In such case, the difference in complexity according to FP method and DOD method is larger that with a lesser number of fields (Figure 2).

To validly define weight values of a given criteria, a lot of measuring has to be performed and the obtained results have to be analyzed. Since the DOD method has just been defined and not enough measuring has been performed yet, we did not dare to determine weight factors, i.e. introduce complexity intervals.

3.3. A Model for Estimating the Complexity of Designing Information Systems Using DOD Method

The DOD method is illustrated by the model in Figure 3. The model is built using the entity-relationship method [4] and added processes on data in entity types.

In the center of the DOD method are document and data, and in the model they are shown by entity types Document and Data. For each document, its code and name are recorded. Document is shown as a weak entity type in relation to System, which means that a certain document "interests" us as a part of a given system, i.e. that one document belongs to the given system. A document can hold more data items, which is shown by relationship cardinality contains (0, M). Data is a weak entity type in relation to Document. Each data has its code and name. One data item can belong to exactly one document – cardinality (1, 1).

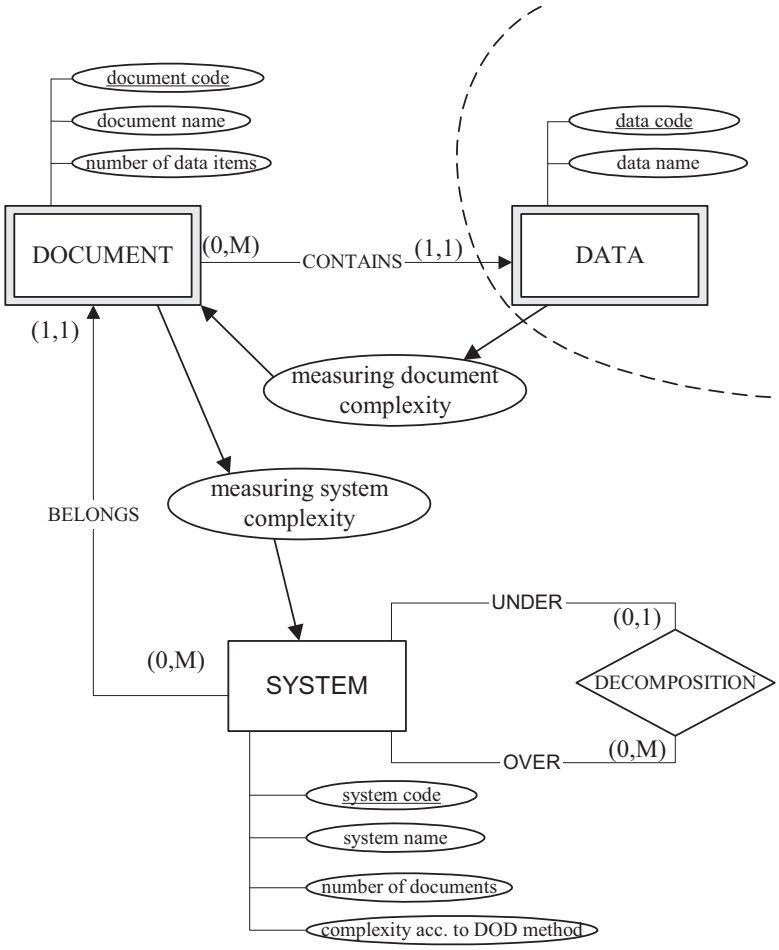

Figure 3. Model of Data on Documents method (DOD)

The cardinality should be explained in more detail: when we observe a data, e.g. person's name and surname, it appears on more documents in a given system. Therefore, we could have cardinality (1, M) at entity type data, so the relationship contains would become an aggregation (Figure 4).

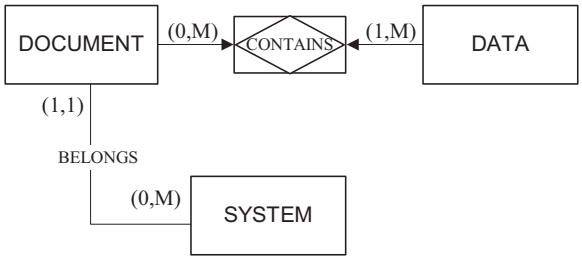

Figure 4. The relationship of entity types Document, Data and System in a business organization

However, as our goal is to make the proposed method as simple and quick as possible, we do not pay much attention whether certain data appears more than once, but we count it separately on each document. If we would account for each data occurrence, then the document analysis would be on the level of entity-relationship model concept, i.e. we would not be concerned about data on documents, instead – we would analyze entity types and the responding attributes.

For each data on document we would be checking whether it is an entity type or an attribute of an entity type. Such an analysis would demand much more time and would make the method more complex, and that is not our aim. Estimation of system complexity using the DOD method is conducted before the system modeling stage, according to the life-cycle of the information system development. Therefore, each data on document is separately counted in the DOD method and belongs only to the given document. The accepted DOD model is shown in Figure 3. Besides documents and data, the model shows another entity type: System. The estimation of complexity is conducted for a system that has its code and name. Systems are, if they are complex, decomposed into smaller subsystems; this is shown by a feedback relationship Decomposition in the figure.

Each system can be a subsystem of a given system, and, on the other hand, can be a suprasystem of a larger number of subsystems. For each system, a total number of documents flowing through that system is also recorded. After applying the DOD method, the system complexity is determined. The relationship Belongs between entity types Document and System illustrates that the document belongs to exactly one system, whether it is a subsystem or a suprasystem. On the other hand, a system without documents may exist. According to the DOD method model, it is possible to acquire data on all levels of a business system. In the first version of DOD method, illustrated in this paper, data appearing on documents is not listed, but only counted and that number is recorded. This is why entity type Data in Figure 3 is separated by a dashed line from the rest of the model. However, listing data can be performed in a way to make the verification of the method easier, because we have the information how the counting took place, i.e. which data is counted on which document. Of course, this would increase the estimation time.

It is visible that the method model is very simple, which attributes to the simplicity of the suggested DOD method. We hold that its application is not limited to experts for estimation of design complexity, and that it should be tested whether “non-experts” could use the method.

3.4. Stages of DOD Method Application

The application of DOD method can be described through the following stages:

1. Definition of system name
2. Decomposition of system into subsystems
3. Data collecting
4. Creating list of documents
5. Determining the number of documents
6. Linking documents and the system
7. Counting data on each document
8. Creating data list
9. Data processing

The first step in the application of DOD method is to define the system for which complexity is being estimated. After that, if needed, the system is decomposed into subsystems. For each defined system/subsystem, all documents used in the system are collected, and the number of these is determined. If a document is being used in more than one subsystem, it is counted for each system separately. A list of collected documents is created, and the documents are linked to the corresponding system. Data is counted on each document, and this number is written down next to the document name on the document list. The number of data on the document shows us the level of document complexity. Optionally, for each document, the counted data can also be noted down, and this enables an easier testing

process of the method application. At the end, the acquired data is processed, i.e. system complexity is estimated by summing up all documents belonging to the system.

The basic aim of the DOD method is to collect all documents of the observed business system, and count data on them. As previously mentioned, the DOD method encompasses basic documents; classic reports and documents describing a process (laws, regulations, etc) are not analyzed and data on them are not being counted. We will illustrate the application of the DOD method.

The chosen document is Prescription from ABCmed system [13] shown in Figure 5. By analyzing this document, one can see it consists of two parts. One part relates to the data items which are filled out by the physician and are a part of AMCmed system, and the other part is filled out by a pharmacist and that part does not belong to the observed IS, and data from this area is not counted. Prescription would, due to the two parts, be interesting during monitoring the document life-cycle as well; however, this is not the subject of this paper. By applying the DOD method on Prescription we come to the number of 17 data items on the document. These are: area office, insurance number, personal identification number, name and surname, address, etc. The counted data are marked with an X in Figure 5.

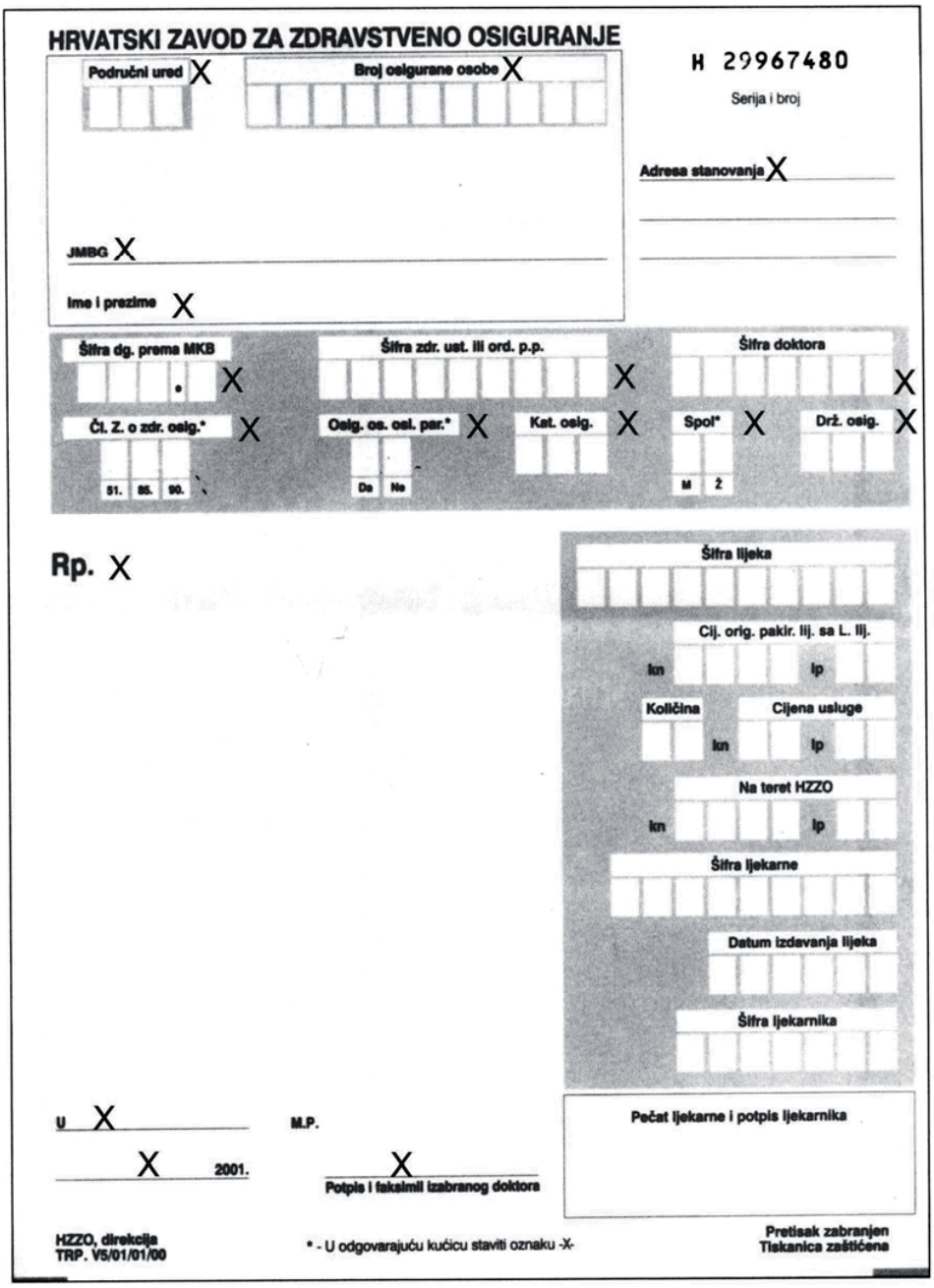

Figure 5. Document example - Prescription

4. The Results of the DOD and FP Method Application on Business Information Systems

An estimation of design complexity was conducted on 11 business information systems [12]. These are:

- ABCmed – general medical practice,
- Basic resources
- Glass insurance,
- Earthquake insurance,
- Stock fire insurance,

- Fire insurance – fixed sum insurance,
- Machinery insurance – fixed sum,
- Burglary/robbery insurance,
- Householders insurance,
- Car responsibility insurance and
- All-risk car insurance.

Two methods were applied on these systems: the function point method and the DOD method.

The results of the application are shown in Table 3.

Business IS	FP	DOD
Basic resources	301	97
Glass insurance	930	281
Earthquake insurance	981	293
Householders insurance	1180	317
Stock fire insurance	1018	318
Machinery insurance – fixed sum	1025	328
Burglary/robbery insurance	1065	333
ABCmed – general medical practice	1619	1211
Fire insurance – fixed sum insurance	1244	514
Car responsibility insurance	1555	994
All-risk car insurance	1393	753

Table 3. The results of the DOD and FP method application on business IS

Figure 6 is a graphic illustration of the acquired data. It can be seen that the complexity estimated by FP method is bigger than the one estimated by DOD method. This is understandable if we consider the way of application and the number of counted, i.e. estimated data. The FP method was applied to finished projects of business IS, and it has 5 measuring parameters where each has its own "weight", in intervals from 3 to 15, with which it is multiplied. The DOD method has only two parameters, and only one of them – the number of data on documents – is used as a result of complexity estimation. On the other hand, the DOD method is applied before the design, so it makes sense that the reached "DOD complexity" would be smaller than the measured complexity of a finished project. A question emerges; is there a connection between two reached complexities? By observing Table 3 and Figure 6, one can see that as "FP complexity" grows, also the "DOD complexity" grows. There is just one exception at IS Householders in relation to IS Stock fire, Burglary and Robbery and Machinery fixed-sum.

The reason for this lies in the fact that the insurance system for household has a lot more identified functional elements, so the system model is far more complex than the rest. On the other hand, the number of documents in the system of household insurance is lesser than the number of documents in other systems, which means that the total number of data items is smaller.

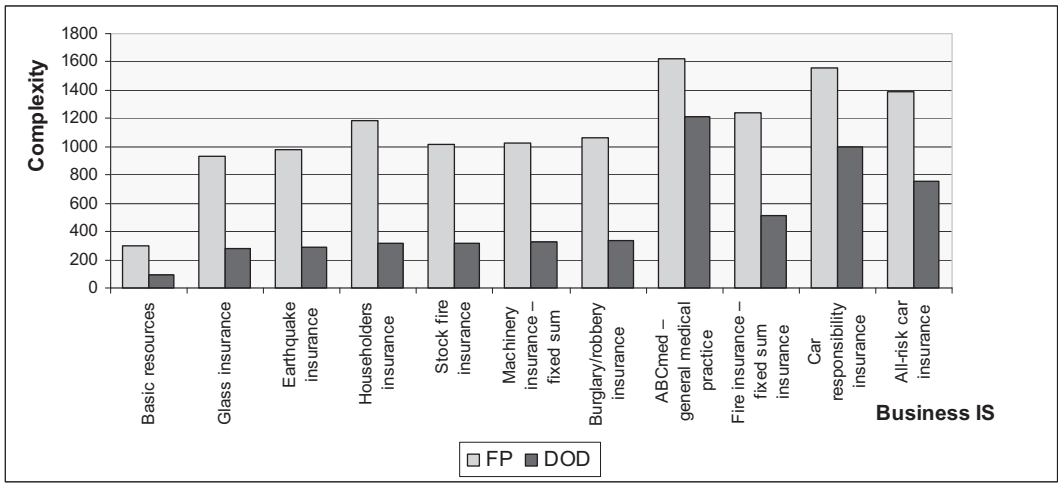

Figure 6. A graphic illustration of the results of FP and DOD method application on business IS

5. The Analysis of the Connection between FP and DOD Method

The results from the conducted measuring and shown in Table 3, were statistically processed by SPSS for Windows v.14. software. The aim of our statistic analysis was to check the degree of data connection reached by FP and DOD method on all units of business IS, and to determine the most appropriate model of the relationship between the mentioned variables (a model by which using DOD results, one can predict FP results with the least possible error).

It is possible to question a linear model of connection and a nonlinear model of connection between results acquired by FP and DOD method.

A nonlinear (logarithmic) connection model was tested. It can be described by the following equation: y=b0 + b1*ln(x). The results of the testing are shown in Table 4.

	Non-standardized coefficients		Standardized coefficients			
	B	Stand. err. B	Beta	$Beta^2$	t	Sig.
ln(DOD) (b_1 and β_1)	499.488	33.783	0.980	0.960	14.785	0.000
(constant b_0)	-1.872.451	203.603			-9.197	0.000

Table 4. Nonlinear model (logarithmic) for predicting the FP results (criteria) based on DOD results (predictors)

Standardized coefficient in nonlinear model is 0.980 (Beta in Table 4), which means that by DOD method results 96.0% (Beta2 in Table 4) of result variance, acquired by FP method, can be explained.

Figure 7 illustrates a linear and nonlinear model of FP and DOD method connection through the scatter diagram.

The nonlinear connection is best described by the logarithmic model. The full line on the scatter diagram represents the linear model (line of regression), and the broken line represents the logarithmic model.

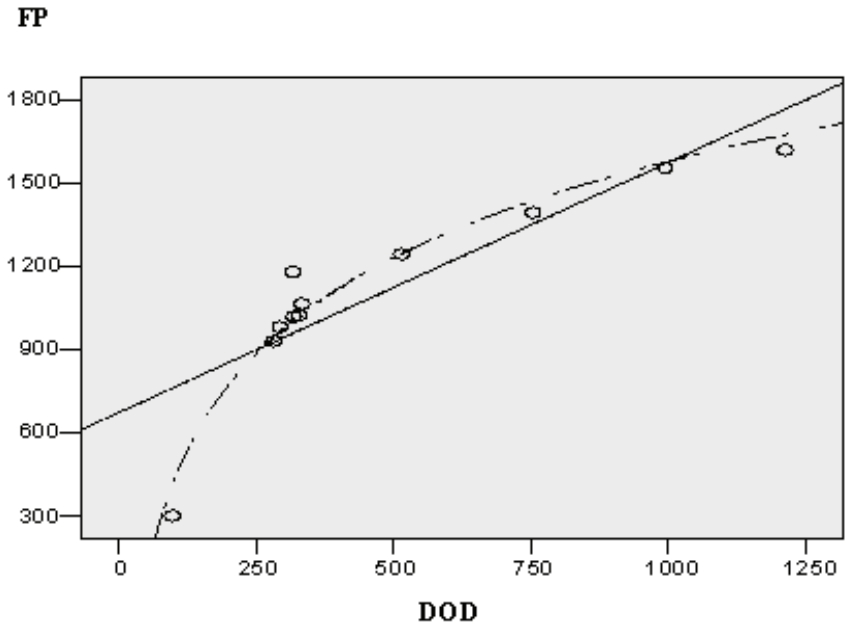

Figure 7. A graphic illustration of linear and logarithmic model of FP and DOD method connection using the scatter diagram

Since the reached standardized parameters are close to their maximum values, it can be concluded that a logarithmic nonlinear model describes the reached data very well, and that it can be used to predict the result of FP method using the known DOD method results. The equation of logarithmic nonlinear model is:

y = -1.872.451 + 499.488*ln(x)

where y represents FP values, and x DOD values, i.e.

FP = -1.872.451 + 499.488*ln(DOD).

The results of comparison acquired by measuring complexity of design on real business IS have shown that there is a correlation between DOD and FP method. It has been shown that by knowing the results of measuring by one method, the value of measuring by the other method can be predicted. The analysis of acquired data has determined that the correlation between the application of the two methods is best described by the logarithmic model of connection, which resulted in the connection coefficient 0.98.

The reached connection coefficient is high; however, due to a relatively small sample (11 business information systems); one should be careful when making general conclusions about the connection coefficient between DOD and FP methods.

6. Conclusion

The complexity of IS development needs to be observed through the complexity of all its stages. However, the analysis of existing methods for measuring complexity shows that their application is the greatest at measuring the complexity of the software itself.

The FP (function point) method, as the most wide-spread method, is a good method for measuring complexity; however, its usage demands a lot of time and effort, not only for the application, but also for familiarizing oneself with the method. It can be used at all stages of IS development, in the stage of design as well, but it is primarily appropriate for measuring complexity after the conducted design and programming.

The DOD method is assigned to the estimation of business IS design complexity; especially in the stage of business IS analysis. The basic assumption of the DOD method is that complexity of a business system process is built into the quantity of documents and data items on them. It has been presumed and shown that the quantity of documents and data on these, which flow through a business system, determines the complexity of IS design. The DOD method is independent from methodologies used at IS development. It only uses business system documents and is applicable in various methodologies.

The greatest advantage of this method is that in a quick and, statistically proven, exact way the complexity of business IS design can be estimated. By applying the DOD method, and depending on the data number, the design complexity expressed in the FP method can be calculated and estimated. This way, time is saved, because the application of the function point method is time-consuming.

References

[1] Ahm, S.I.; Baker, S. The NVC Method of Software Project Estimation. Jackson Open Kilobyte Enterprises, 1998. http://www.tribalsmile.com/nvc/, downloaded: February, 20th 2007.

[2] Barrier, T. System Analysis. *Encyclopedia of Information Systems*, Vol.4, 2003.

[3] Beyers, C.P. Estimating Software Development Projects. *IT Measurement: Practical Advice from te Experts*. IFPUG, Addison-Wesley, Boston, pages 337-362, 2002.

[4] Chen, P.P. The Entity - Relationship Model - Towards a Unified View of Data. *ACM TODS*, Vol.1, No.1, 1976.

[5] COCOMO II model definition manual, 1999. ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf, downloaded: January, 27th 2007.

[6] Damodoran, M; Washington, A.N.E. Estimation using Use Case Points, 2002. http://www.bfpug.com.br/Artigos, downloaded: January, 13th 2007.

[7] Garmus, D; Herron, D. *Function Point Analysis: Measurement Practices for Successful Software Projects*, Addison-Wesley, Boston, 2004.

[8] Gordon, T. J. *The Delphi Method*, 1994. www.futurovenezuela.org/_curso/5-delphi.pdf, downloaded: May, 13th 2007.

[9] Meli, R. Early and Quick Function Point Analysis from Summary User Requirements to Project Management. *IT Measurement: Practical Advice from the Experts*, IFPUG, Addison-Wesley, Boston, pages 417-441, 2002.

[10] Meli, R. Human factors and analytical models in software estimation: an integration perspective. *ESCOM-SCOPE 2000*, Munich, 2000.

[11] Meli, R; Santillo, L. Function point estimation methods: A comparative overview, *FESMA 99*, Amsterdam, 1999.

[12] Poščić, P. *Metoda procjene složenosti projektiranja poslovnih informacijskih sustava*. PhD thesis. Fakultet organizacije i informatike, Varaždin, 2007.

[13] RIS. Dokumentacija projekta ABCMed, Tvrtka RIS, Kastav, 2003.

[14] Santillo, L. ESE: Enhanced Software Estimation. *IT Measurement: Practical Advice from the Experts*, IFPUG, Addison-Wesley, Boston, pages 391-406, 2002.

[15] Santillo, L; Meli, R. Early Function Points: some practical experiences of use. *ESCOM-ENCRESS 98*, Rome, 1998.