COCOMO METHOD FOR PLANNING HUMAN RESOURCES IN SOFTWARE DEVELOPMENT

Katalin Nátz – Szergej Vinogradov – Zsigmond Gábor Szalay

Abstract

Over the past decades, software development has been a priority project for business leaders. Compared to the production or purchase of a productive asset, IT development could be carried out at a lower cost, and controlling always found a hidden reserve that could be used for this purpose.

But this process is in transition at the beginning of the 21st century. IT projects should be examined early in the development phase to determine the functional size of a software application and to determine the business value and expenditure that the application represents for a particular segment of the business. However, process analysis and productivity and cost analysis of these developments are more challenging than average for business analysts.

Enterprise information systems are evolving very rapidly, and their user base is growing. These applications handle so much data and are so complex that it is difficult for the average user to understand how they work. New coding tools are now enabling software developers to meet growing and increasingly complex customer needs more efficiently and quickly. A comprehensible, comparable, and clear method for interpreting, estimating, and verifying projects is needed.

Keywords: *function point, COCOMO, effort estimation,*
JEL: *C88, M11, P42*

Introduction

FSM methodologies aim to define the size of software by quantifying the set of functions provided to users. Function Point Analysis (FPA) was the first FSM method (Albrecht, 1979); it was originally introduced in the mid-1970s. Today, it is used by business organizations worldwide. Allan Albrecht, working at IBM, was the first to develop a method to estimate the boundary and scope of measurement. His name is forever linked to the issue of software functionality. However, one of his main achievements was to use the method to measure the productivity of software projects (Pönsgen, 2012).

Since its founding in 1986, the International Function Point Users Group (IFPUG) has continuously developed and updated the original Albrecht Methodology for Software (IFPUG CPM) (Krasner, 2022). In 2009, a new version of the IFPUG Function Point (FP) standard, CPM 4.3.1, was published and has been in use since early 2010 (Pönsgen, 2012).

As a measure, the function point defines the functional size of applications or software that are created, modified, or deleted during the pre-development phase, including the functions supported by the IT system beyond the boundaries of the measurement. However, only those functions ordered by the sponsor shall be measured from the developed functions.

The FPA method is suitable for measuring complete applications, projects, or so-called releases. The method is independent of the type of application or development. It can also be described as a structured problem-solving algorithm that breaks down systems into smaller components with the aim of making the technical description and implementation of a project more understandable and analyzable.

The paper starts with model-based estimation theories and continues with a presentation of function point analysis. In the second part of the paper, the project under study is presented, with the difference between planned and actual efforts and an analysis of a specific case study. Finally, a summary follows with a critique of the function point analysis with COCOMO, and the conclusions drawn.

Literature review

Theoretical background

Selection of the measurement procedure

At the start of the project, it was not easy to decide which method would be the most appropriate for both traditional and agile software and mixed development. It was not clear that the FPA method was the easiest to implement for this company.

Discussions with the consultants revealed that several methods are used in the group to measure the volume of output, as shown in Figure 1. At the same time, the following was formulated: the measurement method used in the future should be independent of the type of software or the specific technology.

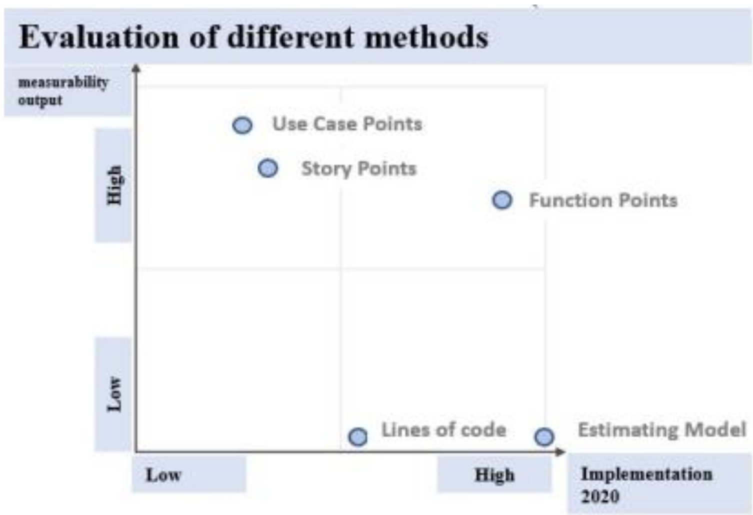

Figure 1: Evaluation of methodologies

Source: own editing for model-based metrics

The functional size of the developed software can be defined, for example, by the amount of source code. Nowadays, however, thanks to modern programming languages and development environments, more and more functions can be developed with less and less source code, so the Lines of Code methodology is not up to the mark. For many years, several methods have been used in different large companies, which can generally be grouped into three broad categories: expert, model-based and hybrid approaches.

Expert methods rely on the explicit and implicit knowledge of experts to make estimates and are the most popular and seem to be the simplest. However, when the resource and cost requirements of estimation are considered in this method, the picture is not so clear. On the other hand, an expert opinion is considered less useful for agile projects (than for traditional projects), as estimates are made for user stories or questions that require different skills from several people (rather than from an expert on a specific task). This is precisely why, for example, Planning Pokers are used in such cases within Scrum teams.

Model-based approaches use data from previous projects and follow the construction of customized models or the use of fixed models; therefore, they are crucial in determining the functional size of software (Choetkiertikul et al, 2016). Hybrid approaches combine expert knowledge with available data, which is perhaps the most effective way to test functionality. Today's dynamic and constant change-driven projects require a different approach to time and resource planning (Alkoffash et al., 2008). Some recent approaches use machine learning techniques to help estimate the business value of agile projects. However, none of these approaches or methods can be said to be exclusively applicable. It is always the business, the situation, and the available data that determine which method is applicable in a given environment. The most typical methods are Domain-specific models (Delphi), Hybrid models (Neural networks, Fuzzy, Rule-based, Regression trees, Hybrid neural fuzzy system), Model-based or technology-specific models (Story Point, Use Case Points, Lines of code, Estimating Models – COCOMO and Function Point) (Balaji et al., 2013).

An overview of relevant literature

COCOMO and COMPLIMO

The COCOMO, also known as the Estimation model, provides a direct estimation tool for calculating costs. It is a simple model based on the analysis of inputs on the functional size of systems and on cost drivers, i.e., cost-increasing factors that affect productivity (Alkoffash et al., 2008).

COCCOMO comprises several models: a base model that is applied to the requirements and an extended model that is used after the design is completed. The COMPLIMO (initial product life cycle cost estimation model is based on the COCOMO II model. Based on experience in aerospace product lines, it presents a model for the use of product lines with characteristic parameters that affect two sources of cost investment and savings: the relative cost of software development with reuse in mind and the relative cost of reuse (Boehm et al., 2004).

Function Point

According to the International Function Point Users Group (IFPUG) standard, Function Point Analysis (FPA) is the most widely used method in the world for measuring the functional size of software, regardless of technical constraints such as programming language. By "software size" we do not mean the file size or code length of the executable software, but the range of performance that the software provides to the user. Functional size, as a unit of measure, defines the functional size of applications and software created, modified, or deleted during development, including the functionality supported by the IT system beyond the measurement boundary. It can also be referred to as a structured problem-solving algorithm that breaks down systems into smaller parts to make the technical description and implementation of a project more understandable and analyzable. Among the functions and data sets developed, only those ordered by the client are measurable (Nátz et al., 2020)

Basics of FPA

When defining the functional scope of an FPA, the user's point of view is crucial. The standard contains specific rules for assigning point values to transactions and databases. Given the number of elementary functions and the normalized values, the allocated frames are cumulative. In the analysis, the elementary processes are the smallest unit that makes sense to the user and represents a complete transaction. A file is considered identical to another if it requires the same data element types, references the same file types, and requires the same processing logic to complete the elementary process (Nátz et al., 2020).

Transaction functions

Table 1: Complexity of EI, EO, EQ

EI		DET 1-4	DET 5-15	DET >=16
FTR	0-1	3	3	4
FTR	2	3	4	6
FTR	>=3	4	6	6

EO		DET 1-5	DET 6-19	DET >=20
FTR	0-1	4	4	5
FTR	2-3	4	5	7
FTR	>=4	5	7	7

EQ		DET 1-5	DET 6-19	DET >=20
FTR	0-1	3	3	4
FTR	2-3	3	4	6
FTR	>=4	4	6	6

Source: Elaborated by the authors

Transactions are the user-provided functions for processing data in the application. There are three types of transactions: external input (EI), external output (EO) and external query (EQ) (Table 1). External Inputs: The main purpose of an input is to maintain one of several internal databases or to change system behavior. To process technical data or control information, an input contains processing logic that goes into the application beyond the limit value (Pönsgen, 2012).

External Outputs: The output provides information to the user. It contains at least one of the following forms of processing logic: mathematic computations, maintenance of datasets, generation of derived data, and change of system behavior (Pönsgen, 2012).

External Inquires: An inquiry is the presentation of information to the user, and it references a dataset to read technical data or control information, and it does not meet the requirements for an EO. It can be also a list box. It is important to mention that the sorting and arrangement of data cannot be evaluated at all (Pönsgen, 2012).

In practice, it is often difficult or impossible to distinguish between EO and EQ. A query should only be evaluated if the output requirements are not met, e.g., in the case of a "simple" list box or multiple queries (Nátz et al., 2020).

Data functions

Table 2: Complexity of ILF, EIF

ILF		DET		
		1-19	20-50	>=51
RET	1	7	7	10
	2-5	7	10	15
	>=6	10	15	15

EIF		DET		
		1-19	20-50	>=51
RET	1	5	5	7
	2-5	5	7	10
	>=6	7	10	10

Source: Elaborated by the authors

Data functions (Table 2) show logical data groupings and databases that the user needs for his work. There are two types of data functions: internal logical files (internal data) and external interface files (external data).

Internal Logical Files (ILF): A dataset that is maintained within the considered application is classified as an internal dataset: user-recognizable, logical groups of functional data, maintained by elementary processes of the application (Pönsgen, 2012).

External Interface Files EIF): A dataset that is read-only but not maintained within the considered application and that is classified as ILF in at least one other application is classified as an External Interface File: a user-identifiable, logical group of functional data that is not changed in the application / to be cared for. It is maintained by elementary processes of another application (Pönsgen, 2012; Longstreet, 2005; Nátz et al., 2020).

The rapid and the complex method

The function point rapid method assigns a point value to a given function or data set after identifying the elementary processes. This is no different for the complex method, but the important difference is that the FP value assigned to a function or data field is weighted according to the number of fields, field groups or data fields the user sees, as shown in Tables 1 and 2. The rapid method is also referred to as the complex approximation method since it generally uses the mean values of the complex method for transactions and the low values for data fields. However, it is important to note that the approximation procedure must be applied consistently. This means that the measurement cannot deviate from the rules laid down and evaluate one or two transactions or data sets differently. In Pönsgen's experience, the measurement error with this approximation is around 5% for baseline measurement and for larger projects. This does not mean that there is that much difference between the two methods, but that experience has shown that this is the number of cases where measurement has led to false results. In smaller projects, however, the error rate can be as high as 20%, in which case the measurement may have to be repeated (Pönsgen, 2012). Some of the companies found during the measurements that the granularity of the existing requirements documentation made it impossible to define data elements, field groups and data fields accurately. This is how the rapid approach was developed, which makes it much easier to define complexity. However, it is not yet part of the standard or uniformly used. No extensive research has been carried out on the real relationship between the rapid and complex methods, whether they differ from company to company or whether a general formula can be used. Not forgetting that if the aim is not just to determine business value, then the complex method may need to be more detailed.

The research hypothesis

The aim of this thesis is to answer the following question by the author when examining the internal systems of a telecommunication company:

H1: *The value (plan) calculated using the COCOMO method for planning human resource requirements in software development is measurably and significantly less than the actual amount (booked, fact) used*

The objective is completed with the following set of conditions: the selection of projects is arbitrary and random and both integrated and non-integrated systems can be compared.

The research processes

1. creation and analysis of a suitable database using the SPSS program:
   a. Preparation of research design (sampling, procedures)
   b. Sample creation (n=121),
   c. Preparation of descriptive statistics,
   d. Homogeneity analysis,
   e. Normality test,
   f. Selection and application of appropriate statistical methods.
2. Evaluation and comparison of the results obtained.

Methodology

Description of the circumstances of the investigation

The analysis of the productivity of software projects has a long history in the company under study. Since 2017, our team has been working on function point analysis within the group. The aim of this workflow is to examine and evaluate applications under development while they are still in progress, using the function point analysis method. In other words, to determine their functional size - like the square meters of an apartment - and then, based on the results, to decide on their further fate: to develop them further or not (Nátz & Orosz, 2018). In addition, create a data and knowledge base that can be used to easily determine the business value of a piece of software, which can be predicted using the function point analysis method. It can facilitate the development of related applications, or even stop an agile development if it does not deliver the expected results based on measurements.

Effort estimation and advantages for the projects

Every software project must have an effort estimate. This effort estimate should be carried out and checked by an expert, and the dual control principle must also be observed. For projects larger than 100 k€ (in some cases 250k€), a Function Point (FP) estimate is also required, determined using the CPSQ estimation method (Krasner, 2022). To facilitate the project and contribute to its success, this function point estimate is offered as a shared service. The Function Point effort estimation is based on two pillars. The functional specification of the application to be developed is determined using the standardized FPA method, which provides several so-called Function Points (FPs). In the second step, the COCOMO II effort estimation procedure determines the project effort in full-time employees (FTEs) or hours) from the function points.

Project procedures

The use of function point cost estimation brings the greatest benefits in the following project phases: the deal phase is a solid basis for initial planning and tender preparation.
- At the start of the project: after the tender has been won, a re-estimation of the project effort is made according to contractually defined requirements
- Post-estimation at the end of the project or release: ensures continuous improvement of estimation by adjusting the estimation parameters.

FPA can be based on an appropriate level of software documentation, from concept descriptions or requirements specifications to user manuals. The level and quality of the documentation determine the level of estimation, but the documentation should be appropriate to the stage of the project. Thus, no additional or special documentation is required from the project.

Statistical findings and discussions

Descriptive statistics

Table 3: Descriptive statistics: estimated and booked effort for IT company applications using FPA rapid and COCOMO in 2022

FP data	N Statistic	Minimum Statistic	Maximum Statistic	Mean Statistic	St. Deviation Statistic	Skewness Std. Error	Kurtosis Std. Error
FP rapid 2022 calc	121	697.00	560000.00	27009.00	73314.70	0.220	0.437
FP complex 2022 booked	121	618.00	657277.00	27948.00	78784.75	0.220	0.437

Source: Computed by the authors

Using the function point analysis and COCOMO method, 121 applications were measured by the rapid method in 2022. To evaluate the database, a correlated sample analysis was performed, which is "1tailed", i.e. the difference is one-way. The data are independent of variables and are of scale type.

Table 3, which shows the estimated and booked effort for the year 2022 for the applications under study, shows that in all cases there is a significant difference between the minimum and maximum values and, accordingly, the associated variance is high. The size of the applications examined varies considerably. The applications range from a navigation system to a ticket manager and SAP. The latter represents much more functionality than an event manager. However, as the method of function point analysis is system independent, this factor is not relevant for the analysis.

Table 4: Estimated and booked efforts for IT company applications using FPA rapid and COCOMO in 2022

Schedule / method	Rapid calculated			Rapid booked			%	
FP rapid 2022 calc	Average	Median	Std. Deviation	Average	Median	Std. Deviation	Average	Median
2022	27009.06	12187.00	73314.70	27948.14	12260.85	0.220	3.48%	0.61%

Source: Computed by the authors

Table 4 shows the estimated and accounted efforts of the function point analysis and COCOMO for the IT company's applications using the rapid method for 2022, i.e., the means, median and standard deviations of the estimated and accounted efforts for the measured applications (n=121). The deviation from the estimated effort is shown by the values in the delta column, alongside which the percentage of these deviations is also shown.

Normality test– estimated and accounted effort in 2022

Table 5: One-sample Kolmogorov-Smirnov test - Estimated and booked effort for IT company applications using FPA rapid and COCOMO in 2022 (N=121)

FP data	N Statistic	FP rapid 2022 calc	FP complex 2022 booked
N		121	121
Normal Parameters	Mean	27009.057	27948.165
	Std. Deviation	73314.704	78784.751
Most Extreme Differences	Absolute	0.360	0.364
	Positive	0.346	0.363
	Negative	-0.360	-0.364
Test Statistic		0.360	0.364
Asymp. Sig. (2-tailed)		<0.001	<0.001
Monte Carlo Sig. (2-tailed)	Sig	0.000	
	99% Confidence Interval Lower Bound	0.000	0.000
	99% Confidence Interval Upper Bound	0.000	0.000

Source: Computed by the authors

The hypothesis test for the group's 2022 rapid estimated and booked effort data requires a normality check before a statistical test. Therefore, a one-sample Kolmogorov-Smirnov was run, the results of which are shown in Table 5.

If the Kolmogorov-Smirnov test is not significant ($p > 0.05$), the variable is normally distributed. A t-test can then be performed. If the variable is significant, a Wilcoxon test should be performed. It can be concluded that the mean function point values of the FPA rapid and COCOMO measurements do not meet the criteria for a normal distribution at 5% significance level in the given time periods. In other words, there are no indicators and periods for which a parametric analysis can be performed. This characteristic determines which tests can be used in hypothesis testing.

For Kolmogorov-Smirnov D (121), the statistical power is as shown in Table 6: 0.360; 0.364. The corresponding 'p' value is $p<0.001$ in all cases; thus, we can assume a non-normal distribution for the yearly measurement results.

Table 6: Kolmogorov-Smirnov test for N=121

	Kolmogorov-Smirnov (a)			**Shapiro-Wilk**		
FP data	**Statistic**	**df**	**Sig.**	**Statistic**	**df**	**Sig.**
FP rapid 2022 calc	0.360	121	<0.001	0.288	121	<0.001
FP complex 2022 booked	0.364	121	<0.001	0.281	121	<0.001

Source: Computed by the authors

***Note**: **a**: Lilliefors Significance Correction*

Wilcoxon test

Table 7: Wilcoxon test results; N=121

Wilcoxon Signed Ranks Test				
		N	**Mean Rank**	**Sum of Ranks**
FP complex calc - FP rapid booked	Negative Ranks – complex < rapid	64	59.36	3799.00
	Positive Ranks – complex > rapid	57	62.84	3582.00
	Ties – complex = rapid	0		
	Total	121		

Source: Computed by the authors

The normality test has shown that it is not possible to perform a parametric test on the measured and logged data produced by FPA rapid and COCOMO. Thus, a Wilcoxon test should be used instead of a t-test. This test ranks the differences between measurements by temporarily discarding their sign during the ranking process. It replaces the differences by their ranks, or ranks, and then assigns the sign of the original differences to the ranks.

Table 7 shows the evolution of the ranks in the rank table for the measurement data for the year 2022. The footnote to the table helps to interpret the data. 121 measurements are shown in the test. We can see that there are more of negative ranks, so the effort data booked is higher.

According to the results of the Wilcoxon signed rank test in Figure 6: (Z= -0.281, p =0.779) there is no significant difference between the estimated (Mdn= 382.71) and the recorded (Mdn=495) effort data. The value is 0.61%% for the median and 3.48% for the mean.

Table 8: Test statistics, N=121

FP data 2022	FP complex booked – FP rapid calc
Z	-0.281
Asymp. Sig. (2-tailed)	0.779

Source: Computed by the authors

***Note*: *a*:** *Wilcoxon Signed Ranks Test* ***b*:** *Based on positive ranks*

Table 8 shows the results of statistical tests for the FPA rapid and COCOMO methods for the period 2022. By examining the values of the paired non-parametric test (Wilcoxon test), it can be seen that similar results are obtained for both the rapid estimated and rapid booked effort data. At a five percent significance level, the difference between the two-effort data can be seen to be statistically unproven when comparing them.

Regression and correlation

Table 9: Spearman correlation

		Value	Asymptotic Standard error (a)	Approximate T(b)	Approximate Significance
Interval by Interval	Pearson's R	0.993	0.002	89.220	<0.001 ©
Ordinal by Ordinal	Spearman Correlation	0.993	0.002	92.679	<0.001 ©
N of Valid Cases		121			

Source: Computed by the authors

***Note*: *a*:** *Not assuming the null hypothesis* ***b*:** *Using the asymptotic standard error assuming the null hypothesis* ***c*:** *Based on normal approximation*

Spearman correlation in Table 9 is a type of Rank Correlation, which belongs to the field of non-parametric statistical procedures. The larger the sample size, the more accurate the correlation value. To measure the closeness of the relationship, the difference between the ranks of the two variables is calculated.

When examining the sample by year and the total sample, the correlation between the estimated and the recorded measurement data is very high, i.e., there is a strong dependent relationship, since the correlation coefficient (r) has a positive sign, is in the same direction and approaches 1. The linear model explains 93% of the complex measurement values with the rapid measurement data.

Summary and conclusion

Extending traditional expert estimation increases the reliability of predictions, improves the quality of project planning, and reduces business risk. By comparing the survey with the results of the function point estimate, we can then get indications as to whether the effort has been over- or underestimated. In addition, the results obtained using FPA can be used for benchmarking with other bidders as well as serve as a solid basis for further negotiations with the client. FPA is even required by some clients as a binding condition in the tendering process.

We, who deal with FPA analyses, play an advisory role in this case. The function points can help to establish effective communication between developers and users and provide a better understanding for the non-technical user. It is applicable throughout the software development life cycle. It is independent of the programming language, development technology, platform or knowledge of the project members and connects directly to user requirements and functions. The team can identify gaps in the functional requirements and thus avoid the early introduction of errors in the application.

Based on the sample examined, which includes data on the size of integrated and non-integrated enterprise information systems, it can be stated that the value (plan) calculated using the COCOMO method for planning human resource requirements is measurably and significantly not less than the actual quantity (booked, actual) used.

The management of the area in question did not expect this result at all, so it asks for an application-by-application comparison for the last 10 years. If the analysis shows no significant discrepancy, the COCOMO methodology is introduced for the pre-calculation of software in other areas.

References

[1.] Albrecht, A. J. (1979): Measuring application development productivity. *Proceedings of IBM Applications Development Symposium,* 83–91.

[2.] Alkoffash, M. – Alrabea, A. (2004): Board characteristics, accounting report integrity, and the cost of debt. *Journal of Computer Science,* 4(7), 606–612. https://doi.org/10.3844/jcssp.2008.606.612

[3.] Balaji, N. – Shivakumar, N. – Vignaraj, V.A. (2013): Software Cost Estimation using Function Point with Non-Algorithmic Approach. *Global Journal of Computer Science and Technology Software & Data Engineering,* 13(8), Version 1.0. *https://globaljournals.org/item/2038-software-cost-estimation-using-function-point-with-non-algorithmic-approach*

[4.] Boehm, B. – Madachy, R. – Yang, Y. (2004): A software product line life cycle cost estimation model. Proceedings. *2004 International Symposium on Empirical Software Engineering.* https://doi.org/10.1109/ISESE.2004.1334903

[5.] Choetkiertikul, M. – Dam, H.H. – Tran, T. – Menzies T. (2016): A deep learning model for estimating story points. *IEEE Transactions on Software Engineering,* 99. https://doi.org/10.1109/TSE.2018.2792473
[6.] Krasner, H. (2022): The Cost of Poor Software Quality in the US: A 2022 Report. *Consortium for information &Software Quality (CISQ)*, 59–61. https://www.it-cisq.org/wp-content/uploads/sites/6/2022/11/CPSQ-Report-Nov-22-2.pdf
[7.] Longstreet, D. (2005): Fundamentals of Function Point Analysis. *Total Metrics, Software Development Magazine.* http://jodypaul.net/SWE/FunctionPointAnalysisFundamentals.pdf
[8.] Nátz, K. – Orosz, G.T. (2018): Function point analysis by an SAP application. i*n AIS 2018: 13th International Symposium on Applied Informatics and Related Areas,* 81–85. http://ais.amk.uni-obuda.hu/proceedings/2018/AIS2018_Abstract_Book.pdf
[9.] Nátz, K. – Orosz G.T. – Szalay, Zs.G. (2020): Methods of functional measurement of software. *in AIS 2020: 15th International Symposium on Applied Informatics and Related Areas,* 112–116. http://ais.amk.uni-obuda.hu/proceedings/2020/AIS2020_Proceedings.pdf
[10.] Pönsgen, B. (2012): *Function-Point Analyse. Ein Praxishandbuch.* 2nd edn. Dpunkt.verlag, Heidelberg.

Authors

Katalin Nátz
ORCID: 0000-0002-5116-6639
PhD student
Doctoral School of Economics and Regional Sciences,
Hungarian University of Agriculture and Life Sciences, H-2100 Gödöllő, Hungary
natz.katalin@phd.uni-mate.hu

Dr. Szergej Vinovgradov
ORCID: 0000-0002-6242-3063
Associate professor, PhD
Institute of Agricultural and Food Economics,
Hungarian University of Agriculture and Life Sciences, H-2100 Gödöllő, Hungary
vinogradov.szergej@uni-mate.hu

Dr. Zsigmond Gábor Szalay
ORCID: 0000-0001-6301-3237
Associate professor, PhD
Institute of Rural Development and Sustainable Economy,
Hungarian University of Agriculture and Life Sciences, H-2100 Gödöllő, Hungary
szalay.zsigmond.gabor@uni-mate.hu