

# Application of Grey System Theory to Software Projects Ranking

The aim of this work is the identification of software project quality performance measures that would enable valid comparison and ranking of the completed projects. Software projects can be characterized by a set of influence factors. A subset of the influence factors is relevant for software project quality. In order to predict and determine the ranking of software projects by their success, and thus present a valid software project quality performance measure, we employ Grey system theory. Grey relational analysis is a kind of method which enables determination of the relational degree of every factor in the system. The method can be used for systems that are incompletely described with relatively few data available, and for which standard statistical assumptions are not satisfied. Relational degree between seven relevant software project quality influence factors is calculated for a set of ten software projects. The results demonstrate the usefulness and applicability of the Grey system theory in software project quality assessment.

**Key words:** Grey system theory, Software quality, Influence factors, Relational analysis

**Primjena teorije sivih sustava na kvalitetu softverskih projekata.** Cilj rada je identifikacija mjera performanse kvalitete softverskog projekta koje bi omogućile usporedbu i rangiranje završenih projekata. Softverski projekti mogu se okarakterizirati skupom faktora utjecaja. Podskup tih faktora važan je za kvalitetu softverskih projekata. Da bi se predvidio i odredio poredak softverskih projekata prema uspješnosti i na taj način predstavilo valjanu mjeru performanse kvalitete softverskih projekata, koristi se teorija sivih sustava. Siva analiza odnosa je takva metoda koja omogućava određivanje stupnja odnosa svakog faktora u sustavu. Takva metoda koristi se za sustave koji su nepotpuno opisani s relativno malo podataka i za koje standardne statističke pretpostavke nisu zadovoljene.

Stupanj odnosa između sedam bitnih faktora kvalitete softverskih projekata izračunat je na skupu od deset softverskih projekata. Rezultati pokazuju učinkovitost i primjenjivost teorije sivih sustava u ocjeni kvalitete softverskih projekata.

**Ključne riječi:** teorija sivih sustava, kvaliteta softvera, faktori utjecaja, analiza odnosa

## 1 INTRODUCTION

The inherent uncertainty and incomplete information of the software development process presents challenges for identifying fault-prone modules and providing a preferred model early enough in a development cycle in order to guide software enhancement estimates under small sample and uncertain conditions.

This work examines the potential benefits for providing a software-quality classification based on Grey relational analysis. It attempts to identify software project quality performance measures that would enable valid comparison between the completed projects.

The advantage of the Grey system theory is that it is designed to study uncertainty. It is shown that Grey theory is superior to other methods in theoretical analysis of systems with uncertain information and incomplete data

samples [2]. Especially, it can be used if the large samples are not available or if the user is uncertain whether the data is representative. It can also be used in early effective factor assessment [4]. Therefore, this study adopts Grey system theory approach to propose a feasible and effective software quality analysis method.

Traditional methods require a large number of samples and a data distribution that has to be typical for the process at hand. In contrast, the Grey system theory is designed to work with a system where the available information is insufficient to fully characterize the system [15]. The term "Grey" stands for poor, incomplete and uncertain, and is especially used in relation to the concept of information [8]. The major advantage of Grey theory is that it can handle both incomplete information and unclear problems very precisely [1]. Grey relational analysis (GRA), which is a part of Grey theory, is a kind of method by which the

relational degree of every factor in the system can be analyzed [13]. The main function of GRA is to indicate the relational degree between two measurement sequences by using the discrete measurement method to measure the distances [8]. This means that GRA uses information from the Grey system to dynamically compare influence factors quantitatively. This approach is based on the level of similarity and variability among all factors to establish their relation.

Main contributions to the Grey system theory today come from two parts: GRA and Grey modeling (GM) [13]. GRA can be used for system analysis as an alternative to statistical methods. GM is developed based on requirements for system modeling with limited data, which constitutes a problem for most of the traditional modeling methods.

In this paper we investigate the effectiveness of the Grey system theory to determine the ranking of software projects using relevant influence factors.

The contributions of this paper are:

- a) Identification of a set of software project performance measures and influence factors that are used by software development projects so that a valid comparison of performance can be made between the projects when they are completed.
- b) Demonstration of applicability of GRA to software projects influence factors analysis, which allows for faster, more robust and more effective software projects comparison.

The structure of this paper is as follows: Section 2 discusses some related work on Grey theory. Section 3 elaborates methodology, data collection, and influence factors for software project quality. In Section 4, the Grey theory is presented and explicated. Section 5 elaborates the calculation of the Grey relational degree for relevant software projects influence factors. Section 6 discusses the results from the perspective of the future projects applicability, and finally the conclusion is given in Section 7.

## 2 RELATED WORK

Previously, methods such as “Fuzzy Bayes classifier” [18], which consists of a Bayesian classifier with weighting factors, have been used in analysis of systems with uncertain information and incomplete samples. The proposition of Grey theory that occurred in the period 1982 to 1999 resulted in the use of Grey theory to a number of fields, and the development and application of the theory is still in progress [1,7,13]. Deng [5] proposed Grey system theory in 1982 as a simple and accurate method for multiple attributes decision problems. Grey system as proposed

by [5] included GRA for effective selection of relevant attributes.

Since then, the application of Grey system theory has extended to industry, agriculture, economy, energy, transportation, military, legal, financial and other fields, and has successfully resolved a large number of practical problems in production, life, and scientific research [9,12,15].

Chih-Hung et al. [3] applied GRA to the vendor evaluation model. Hsu and Wang [7] applied GRA to forecast the outputs of integrated circuit industry. Recently this method was also used in the field of sports. For example, Chang et al. [1] applied GRA to the decathlon evaluation model with satisfying results. GRA was also applied to project selection, prediction analysis, performance evaluation, and factor effect evaluation due to the Grey relational analysis based software development [17,19].

## 3 METHODOLOGY

### 3.1 Data collection

The Competence Center for Software Engineering (CCSE) was established in Osijek, Croatia. It is intended for the development of logistics and information technology in wider region, while promoting software quality, reliability and diagnostics, which is an important part of software engineering [16].

An initiative from CCSE is the implementation of a program for software project performance measurement. It is a process of assessing the results of a company, organization, or individual included in the project. The main goals of the assessment are:

- a) to determine the effectiveness of the project operations,
- b) to make changes by addressing observed performance gaps, shortfalls, and other unwanted issues.

Companies and organizations measure their performance in a variety of areas using different methods and criteria for different purposes. In order to be able to compare the performance measures, they need to be commonly defined. We used a list of performance measures according to the Software Engineering Institute’s Technical report [10] in order to:

- (1) define a set of key performance measures that should be used by all software projects,
- (2) define the influence factors for these measures.

The list of software projects influence factors with the corresponding performance measures is shown in Table 1 [10].

Out of the influence factors listed in Table 1, those that are considered by experts to be the most relevant from the software projects quality perspective are listed in Table 2, together with a short definition.

Table 1. Influence factors of software projects.

Number	Factor	Measure
1	Project size	Fp*
2	Artifact reuse	%
3	Project type	Type**
4	Application domain	***
5	Average team size	number
6	Maximum team size	number
7	Team expertise	years
8	Process maturity	CMM****
9	Functional requirement stability	number
10	Project effort	hours
11	Productivity	FP* per hour
12	Project duration	days
13	Schedule predictability	%
14	Requirements completion ratio	%

\* Functional point

\*\* New software, reengineering, or modification

\*\*\* Enterprise, market/industry

\*\*\*\* Capability Maturity Model [14].

Table 2. List of software project quality influence factors.

Number	Influence factor	Definition
1	Artifact reuse	It is the use of existing software or software knowledge to built new software or new documents for the project under consideration.
2	Team expertise	It is a 5-tuple of measures of the proficiency of the project team during each phase of the development cycle.
3	Process maturity	Extent to which project's processes are explicitly defined, managed, measured, and controlled.
4	Functional requirement stability	Measure that quantifies the cumulative degree to which the requirements changed throughout the life cycle of the project from the original requirements baseline.
5	Team productivity	Project size vs. project effort. Expressed as project size per project hour, project size depends on how the size is measured by an organization (e.g. lines of code, functional points)
6	Schedule predictability	Measure of how much the original project duration estimate differs from the actual project duration that was achieved.
7	Requirements completion ratio	Measures the extent to which planned functional requirement were satisfied in the final product implementation.

### 3.2 Explanation of influence factors

In this subsection we provide a thorough explanation of the relevant software projects influence factors listed in Table 2, according to [10].

1. *Artifact reuse* is the use of existing software or software knowledge to build new software or new documents for the project under consideration.

Reusable software knowledge items are referred as reusable artifacts or reusable assets and may include re-

quirements documents, designs, test cases, code, documentation or any other work product that is a part of the project's development process.

An artifact reuse value is determined based on the reuse assessment method that is employed. A proxy measure of artifact reuse is defined by:

$$Artifact\ reuse = \frac{PE_{Saved}}{PE_{Total}} * 100, \quad (1)$$

where  $PE_{Saved}$  is the project effort that was conserved or saved through the reuse of preexisting work products, and  $PE_{Total}$  is the total project effort.

Developing an estimate of artifact reuse relies on judgments made about: (a) the percent of overall project effort required to develop the artifacts, (b) the percent of effort savings realized by artifact reuse. Savings can be estimated from past experience based on the knowledge of team and technology involved. Also, the amount of time that was required for previous implementations, which can be measured objectively in the number of working hours, can be used to assess conserved effort.

2. *Team expertise* is a 5-tuple of measures of the proficiency of the project team during each phase of the development life cycle.

The measure is a subjective one based on the informed expert judgment of those who perform the assessment. The team expertise measure for each phase is an integer in range (1-5) where 1 represents novice proficiency ability and 5 represents expert proficiency:

$$TE = (TE_{req}, TE_{arch}, TE_{edd}, TE_{code}, TE_{est}) \quad (2)$$

where:

$TE_{req}$  is expertise rating for team members, who contribute to the Concept and Requirements Analysis Phase,

$TE_{arch}$  is expertise rating for team members, who contribute to Architectural and/or High-Level Design Phase,

$TE_{edd}$  is expertise rating for team members, who contribute to Detailed Design Phase,

$TE_{code}$  is expertise for Code Construction und Unit Testing Phase, and

$TE_{est}$  is expertise rating for team members, who contribute to System Test Phase.

$TE$  can be represented as a single number by calculating the mean value of the individual phases measurements.

3. *Process maturity* is the extent to which a project's processes are explicitly defined, managed, measured, and controlled. Some of the approaches include ISO 9001 and ISO 15504 (SPICE) standards, and SEI CMM [6]. These approaches use different rating schemes to indicate the degree of process maturity. A maturity level is a defined evolutionary plateau for organizational process improvement.

The maturity levels are based on CMM model [14,21]. It differentiates between five different maturity levels of software processes: 1 - Initial process, 2 - Repeatable process, 3 - Defined process, 4 - Managed process, and 5 - Optimizing process. The maturity levels are measured by the achievement of the goals associated with each predefined set of process areas. Within each of the maturity levels are Key Process Areas (KPAs) which characterize that level, and for each KPA there are five definitions identified: Goals, Commitment, Ability, Measurement, and Verification. The KPAs are not necessarily unique to CMM, representing as they do the stages that organizations must go through on the way to becoming mature. The CMM provides a theoretical continuum along which process maturity can be developed incrementally from one level to the next.

4. *Functional requirements stability* (FRS) is a measure that quantifies the cumulative degree to which the requirements changed throughout the life cycle of the project from the original requirements baseline. FRS is defined as

$$FRS = \frac{R_T - R_C}{R_T}, \quad (3)$$

where  $R_T$  is the total number of requirements that were originally base-lined at the beginning of the project; and  $R_C$  is the total number of changes to the original base-lined requirements.

The maximum value of FRS is 1.0, indicating complete stability of the functional requirements.

5. *Team Productivity* of a software project is calculated as follows:

$$Team\ Productivity = \frac{Project\ size}{Project\ effort} \text{ [units/hour]} \quad (4)$$

where *Project size* depends on how the project size is measured by an organization (e.g. lines of code, functional points including: number of user inputs, number of user outputs, number of user inquiries, number of files, number of external interfaces, MM - men/month), and *Project effort* is usually defined in project hours (see also Table 1).

For example, if a developed project acquired 130 FPs and this was accomplished in 5300 hours, then its *Team productivity* can be calculated as:

$$Team\ Productivity = \frac{130}{5300} = 0.025 \text{ [FP/hour]}$$

6. *Schedule predictability* is a measure of how much the original project duration estimate differs from the actual project duration that was achieved. Schedule predictability is defined as a percentage:

$$SP = \frac{Project\ durat. - Estim.\ project\ durat.}{Estimated\ project\ duration} * 100 [\%], \quad (5)$$

where *Project duration* is measured in hours, *Estimated project duration* is the original estimate of project duration as documented in the base-lined version of the project plan. Note that schedule predictability is a positive value when there is a schedule overrun and a negative value when there is a schedule underrun.

For example: The estimated duration was documented as 316 days in the project plan. The actual duration realized was 325 days. Therefore, Schedule predictability overrun is calculated as

$$SP = \frac{325 - 316}{316} * 100 = 2.85\%$$

7. *Requirements Completion Ratio*. Functional requirements describe what the system, process, product, or service must do in order to fulfill the user requirements.

The Requirements Completion Ratio (RCR) measures the extent to which planned functional requirements were satisfied in the final product implementation.

RCR is expressed as a percentage:

$$RCR = \frac{Satisfied\ requirements}{Planned\ requirements} * 100 [\%], \quad (6)$$

where *Planned requirements* is given as the number of requirements that were originally base-lined at the beginning of the project and that have been added or modified through negotiation with the user, and *Satisfied requirements* is given as the number of functional requirements that the user considers to be satisfied in the delivered software product.

For example: The original base-lined functional requirements specification contained 34 requirements, and 28 of those were satisfied, thus

$$RCR = \frac{28}{34} * 100 \% = 82.35 \%$$

## 4 GREY SYSTEMS THEORY

### 4.1 Grey theory steps

The information that is either incomplete or undetermined is called Grey. The Grey system provides multidisciplinary approaches for analysis and abstract modeling of systems for which the information is limited, incomplete and characterized by random uncertainty [15].

The 1<sup>st</sup> order one variable Grey model denoted as GM (1, 1) is especially applicable for forecasting. GM (1, 1) model uses the variation within the system to find out the relations between sequential data and then establish the prediction model [15].

The three terms that are typical symbols and features of Grey System are [3]:

- a) The Grey number in Grey system is a number with incomplete information.
- b) The Grey element represents an element with incomplete information.
- c) The Grey relation is the relation with incomplete information.

There are several steps of the theory of Grey system [12]:

1. Grey generation: This is data processing to supplement information. It is aimed to process those complicate and tedious data to gain a clear rule, which is called the whitening of a sequence of numbers. The expected goal for each influence factor is determined based on the principle of data processing, as explained in section 4.2
2. Grey modeling: The modeling is performed in order to establish a set of Grey variation equations and Grey differential equations, which is called the whitening of the model. The Grey model is denoted as GM  $(n, h)$ , which is a  $n$ -th order differential equation of  $h$  variables. This Grey differential equation is used for infinite information. Most of the previous researchers have focused on GM  $(1, 1)$  models because of its computational efficiency. GM  $(1, 1)$  model have time – varying coefficients. It means that the model is renewed as the new data become available to the prediction model. A Grey differential equation having  $N$  variables is called GM  $(1, N)$ .
3. Grey prediction: Uses the Grey model to conduct a qualitative prediction, which is called the whitening of development. Grey models predict the future values of a time series based on a set of the most recent data.
4. Grey decision: A decision is made under imperfect countermeasure and unclear situation, which is called the whitening of status. It is primarily concerned with the Grey strategy of situation, Grey group decision making and Grey programming [5]. Grey strategy of situation deals with the strategy making based on multi objects which are contradictory in the ordinary way. It is important to make a satisfactory strategy by means of effect measure maps, which transfer the disconformities samples resulting from different objects into identical scales.
5. Grey relational analysis: Quantifies all influences of various factors and their relation, which is called the whitening of factor relation. It uses information

from the Grey system to dynamically compare factors quantitatively, based on the level of similarity and variability among factors to establish their relation. GRA analyzes the relational grade for discrete sequences.

6. Grey control: Work on the data of system behavior and look for any rules of behavior development to predict future behavior. The predicted value can be fed back into the system in order to enable system control.

This study will adopt the above mentioned research steps to develop an influence factors evaluation model based on GRA and to apply them to influence factors evaluation and selection.

#### 4.2 Grey relational analysis

The generation of Grey relational degree for software projects is shown in Fig. 1. The process is elaborated here.

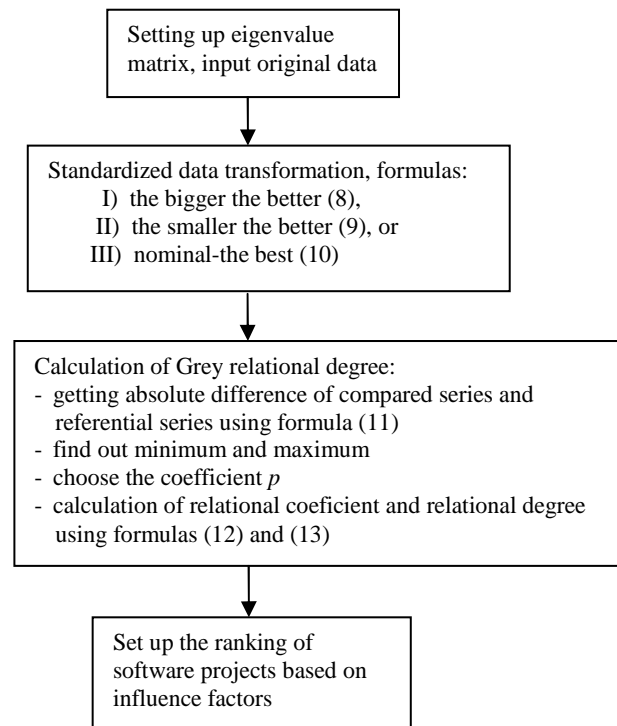


Fig. 1. The generation of Grey relational degree for software projects.

Let the number of the listed software projects be  $m$ , and the number of the influence factors be  $n$ . Then a  $m \times n$  value matrix (called eigenvalue matrix) is set up [19]:

$$X = \begin{bmatrix} x_1(1), x_1(2), \dots, x_1(n) \\ x_2(1), x_2(2), \dots, x_2(n) \\ \dots \\ \dots \\ x_m(1), x_m(2), \dots, x_m(n) \end{bmatrix}, \quad (7)$$

Table 3. The original data of the Influence factors of software projects.

Project number	Artifact reuse A1	Team expertise A2	Process maturity A3	Functional requirement stability A4	Productivity (FP per hour) A5	Schedule predictability A6	Requirements completion ratio A7
X1	0.014	3.0	2.0	0.85	0.026	0.025	0.955
X2	0.012	3.0	2.0	0.80	0.027	0.022	0.960
X3	0.015	2.4	2.5	0.80	0.027	0.024	0.965
X4	0.020	2.4	3.0	0.82	0.026	0.025	0.907
X5	0.014	2.4	2.8	0.78	0.028	0.010	0.924
X6	0.011	2.8	2.5	0.86	0.034	0.010	0.950
X7	0.021	2.8	2.0	0.84	0.028	0.018	0.944
X8	0.015	2.8	2.8	0.80	0.030	0.018	0.960
X9	0.012	3.2	2.6	0.75	0.028	0.020	0.908
X10	0.012	3.2	3.2	0.80	0.028	0.022	0.954

where  $x_i(k)$  is the value of the number  $i$  listed project and the number  $k$  influence factors.

Usually, three kinds of influence factors are included, which are:

1. Benefit – type factor (the bigger the better),
2. Defect – type (the smaller the better)
3. Medium – type, or nominal-the-best (the nearer to a certain standard value the better).

It is difficult to compare between different kinds of factors because they exert a different influence. Therefore, the standardized transformation of these factors must be done. Three formulas can be used for this purpose [3].

$$x_i(k) = \frac{x_i(k) - \min x_i(k)}{\max x_i(k) - \min x_i(k)}. \quad (8)$$

The first standardized formula is suitable for the benefit – type factor.

$$x_i(k) = \frac{\max x_i(k) - x_i(k)}{\max x_i(k) - \min x_i(k)}. \quad (9)$$

The second standardized formula is suitable for defect – type factor.

$$x_i(k) = \frac{|x_i(k) - x_0(k)|}{\max x_i(k) - x_0(k)}, \quad (10)$$

where  $x_0(k)$  is a certain standard value. The third standardized formula is suitable for the medium – type factor.

The Grey relation degree can be calculated by the steps as follows:

1. The absolute difference of the compared series and the referential series should be obtained using the following formula [19]:

$$\Delta x_i(k) = |x_0(k) - x_i(k)|, \quad (11)$$

and the maximum ( $\Delta_{\max}$ ) and the minimum ( $\Delta_{\min}$ ) difference should be found.

1. The distinguishing coefficient  $p$  is chosen between 0 and 1. Generally, the distinguishing coefficient  $p$  is set to 0.5 [5,8,18].
2. Calculation of the relational coefficient and relational degree by (12) as follows.

In Grey relational analysis, Grey relational coefficient  $\xi$  can be expressed as [19]:

$$\xi_i(k) = \frac{\Delta_{\min} + p\Delta_{\max}}{\Delta x_i(k) + p\Delta_{\max}}, \quad (12)$$

and then the Grey relational degree follows as:

$$r_i = \sum [w(k)\xi(k)]. \quad (13)$$

In equation (13),  $\xi$  is the Grey relational coefficient,  $w(k)$  is the proportion of the number  $k$  influence factor to the total influence indicators. The sum of  $w(k)$  is 100%. The result obtained when using (13) can be applied to measure the quality of the listed software projects.

## 5 CALCULATION OF THE GREY RELATIONAL DEGREE FOR INFLUENCE FACTORS

Table 3 shows seven influence factors (A1–A7) measured on a small set of 10 medium and small software projects (X1–X10) performed in the CCSE program.

The influence factors are all benefit – type factors (the bigger the better), except for project predictability, which is a defect-type factor (it needs to be as close to zero as possible). For example, A4=0.86 (FRS) is obtained for project

Table 4. The compared series and the referential series.

Project number	Artifact reuse A1	Team expertise A2	Process maturity A3	Functional requirement stability A4	Productivity (FP per hour) A5	Schedule predictability A6	Requirements completion ratio A7
X0	1.00	1.00	1.00	1.00	1.00	1.00	1.00
X1	0.30	0.75	0	0.09	0	0	0.83
X2	0.10	0.75	0	0.55	0.13	0.2	0.91
X3	0.40	0	0.42	0.55	0.13	0.07	1
X4	0.90	0	0.83	0.36	0	0	0
X5	0.30	0	0.67	0.73	0.25	1	0.29
X6	0.00	0.50	0.43	0	1	1	0.74
X7	1.00	0.50	0	0.18	0.25	0.47	0.64
X8	0.40	0.50	0.67	0.55	0.5	0.47	0.91
X9	0.10	1.00	0.50	1.00	0.25	0.33	0.02
X10	0.10	1.00	1.00	0	0.25	0.2	0.81

Table 5. The relational degree and the ranking.

	Artifact reuse A1	Team expertise A2	Process Maturity A3	Functional requirement stability A4	Productivity (FP per hour) A5	Schedule predictability A6	Requirements completion ratio A7	Relational degree	Rank
w(k)	0.30	0.20	0.20	0.10	0.08	0.08	0.04	-	
$\xi_1$	0.63	2.50	1.00	0.85	1.00	0.33	0.38	1.096	2
$\xi_2$	0.83	2.50	1.00	0.48	0.79	0.38	0.35	1.104	1
$\xi_3$	0.56	1.00	0.54	0.48	0.79	0.35	0.33	0.628	3
$\xi_4$	0.36	1.00	0.38	0.58	0.50	0.33	1.00	0.548	8
$\xi_5$	0.63	1.00	0.43	0.41	0.67	1.00	1.58	0.733	4
$\xi_6$	0	0.50	0.54	1.00	0.33	1.00	0.40	0.420	10
$\xi_7$	0.33	0.50	1.00	0.74	0.67	0.49	0.44	0.659	5
$\xi_8$	0.56	0.50	0.43	0.48	0.50	0.49	0.35	0.495	9
$\xi_9$	0.83	0.33	0.50	0.33	0.67	0.43	0.96	0.568	7
$\xi_{10}$	0.83	0.33	0.33	1.00	0.67	0.38	0.38	0.572	6

X6, which is the largest result among the projects. The reference series is therefore  $X0 = (0.021, 3.2, 3.2, 0.86, 0.034, 0.010, \text{ and } 0.965)$ . The reference series includes the best results for influence factors obtained over all the considered projects. Equations (8) and (9) can be used to do the standardized transformation of this sample, and the result is shown in Table 4.

The next step is to calculate the absolute difference between the compared series and the referential series using (11), and to find the maximum and the minimum. The distinguishing coefficient  $p$  is set to 0.5. The Grey relational coefficient can be calculated using (12).

In formula (13),  $w(k)$  for every influence factor and for every factor report may be different. If we suppose:  $w(1) = 0.30, w(2) = 0.20, w(3) = 0.20, w(4) = 0.10, w(5) = 0.08, w(6) = 0.08, w(7) = 0.04$ , the relational degree can be ob-

tained using (13). The result is listed in Table 5. According to the results obtained for the Grey relational degree in Table 5, the ranking of the software project based on Grey analysis of the influence factors is:

$$r_2 \rangle r_1 \rangle r_3 \rangle r_5 \rangle r_7 \rangle r_{10} \rangle r_9 \rangle r_4 \rangle r_8 \rangle r_6.$$

The ranking is also listed in Table 6. It should be noted that the ranking will change as the weighting value for each evaluating factor is modified. In other words, more relevant factors can be selected by increasing weights, based on the software project requirements.

## 6 DISCUSSION

The enterprises and organizations try to understand their overall performance of software projects and compare it,

Table 6. The ranking of the projects.

Rank	1	2	3	4	5	6	7	8	9	10
Project number	X2	X1	X3	X5	X7	X10	X9	X4	X8	X6

with intention to find a way to improve. The software performance measures are core measures that should be identified as a part of a set of critical measures of success because they address important attributes of any software development project.

Organizations that are more experienced in measurement of software projects usually want to compare performance with their competitors. Before valid measurement comparison can be conducted, common operational definitions for performance measures have to be in place. In this way, organizations are able to efficiently compare software project performance among projects within their organizations as well as with projects outside of their organization.

Detailed monitoring of influence factors is a prerequisite for successful application of the Grey analysis. Monitoring should be performed by an expert quality assurance (QA) team. Having a QA team is common for any major company and it should be encouraged for mid-level and small companies as well.

In order to simplify the calculation of relation between the completed projects using GRA, a sample of only ten software projects have been chosen for this study. Gray system can be used to compare any number of software projects. Hence, the Grey theory imposes no restriction regarding the number of potentially analyzed projects.

The Grey system theory has been successfully applied to various fields and had made a success in analyzing uncertain systems. Traditional methods usually require a large amount of historical data in order to obtain a known statistical data distribution function to be able to make an accurate assessment and prediction of the required indicators [17].

In contrast to the traditional prediction methods, the main properties of the Grey theory are: 1) it does not need to make strict assumptions about the data set, and 2) it is used successfully to analyze uncertain systems that have multi-data inputs, discrete data, and insufficient data. These properties greatly simplify data collection and management, and also allow for timely predictions to be made, sometimes even faster and more accurate than artificial neural networks [11].

Grey system theory is somewhat different from Fuzzy logic approach. The emphasis of the Grey theory is on objects that have clearly defined external boundaries but vague or unknown position between the boundaries. The focus of Fuzzy logic is on objects whose properties have

fuzzy or unclear external boundaries, but have a clear internal membership. Gathering knowledge about Grey systems makes grey objects more white or precise, while gathering knowledge about Fuzzy logic object makes us surer about its membership value, but the value is still fuzzy [20].

The obtained ranking of the projects is a function of both the seven influence factors and the weights that are used to specify which factor is more significant to take into consideration. Since the weights are specified by the involved subject, whoever he is, the final ranking should be regarded as a result of the analysis process by the Grey system theory; it is not the only possible solution. From the perspective of the evaluated projects, those that obtained higher ranking using Grey relational analysis now have an objective reason to be promoted and/or continued, as GRA has been proven to work in practice. Projects with lower ranking will lack such a reason and would be considered as failures from the perspective of QA management, company director, competence centre, or other involved subjects.

## 7 CONCLUSION

Grey system based methods provide various tools to cope with situations of limited data, such as correlation analysis and modeling. Grey system theory aims to deal with the uncertainty of a system by using elements of relational analysis, operational research, system control, system modeling and system forecasting. Through quantitative analysis of Grey relation, it provides more accurate and subjective data. Most distinguished Grey theory methods that are in use are Grey relational analysis and Grey modeling.

The purpose of GRA is not to provide a general method for project evaluation, but a practical and applicable one, especially for solving some specific project evaluation question such as project quality control. GM provides a tool for modeling of discrete series with few data. Focus of GRA and GM is more on the method than on theoretical foundations. Further development of the methods and their foundations is required, particularly regarding comparison with known data and expert systems formalisms.

Based on the study in this paper, GRA can be applied to software projects ranking based on software project influence factors. This study adopted Grey system theory to propose a reasonable method for projects ranking. The final project ranking is dependent on the subject's preference of the influence factors and reflects the goal that needed to be achieved. The results indicate that the Grey system theory is a feasible and effective software project analysis method. This finding may serve as a reference to future studies in this and other research fields.



## REFERENCES

- [1] C. L. Chang, C. H. Tsai, L. Chen, "Applying Grey Relational Analysis to the Decathlon Evaluation Model," *International Journal of the Computer, The Internet and Management*, vol. 11, no. 3, pp. 54-62, 2003.
- [2] Y. M. Chiang, H. H. Hsieh, "The use of the Tagouchi method with grey relational analysis to optimize the thin-film sputtering process with multiple quality characteristics in color filter manufacturing," *Computers & Industrial Engineering*, vol. 56, pp. 648-661, 2009.
- [3] T. Chih-Hung, C. Ching-Liang, C. Lieh, "Applying Grey Relational Analysis to the Vendor Evaluation Model," *International Journal of the Computer, The Internet and Management*, vol. 11, no. 3, pp. 45-53, 2003.
- [4] N. H. Chiu, "An early software – quality classification based on improved grey relational classifier," *Expert Systems with Applications*, vol. 36, issue 7, pp. 10727-10734, 2009.
- [5] J. L. Deng, "The introduction of grey system," *The Journal of Grey System*, vol. 1, no. 1, pp. 1-24, 1982.
- [6] C. Ebert, R. Dumke, *Software Measurement*, CityplaceBerlin, country-regionGermany: Springer, 2007.
- [7] L. C. Hsu, C. H. Wang, "Forecasting integrated circuit output using multivariate grey model and grey relational analysis," *Expert Systems with Applications*, vol. 36, issue 2, part 1, pp. 1403-1409, 2009.
- [8] J. C. Huang, "The key factor of the internet information technology on the quality of life for the elderly: application of grey system theory," *IEEE SMC – eNewsletter*, issue 33, 2010.
- [9] F. Jovic, D. Krmptotic, A. Jovic, M. Jukic, "Evaluation of Grey Prediction Method of Energy Consumption," in: *Proceedings of the 28th International Convention on Information and Communication Technology, Electronics and Microelectronics MIPRO 2007*, (Opatija, Croatia), CTS&CIS, pp. 64-67, 2005.
- [10] M. Kasunic, "A Data Specification for Software Project Performance Measures: Results of a Collaboration on Performance Measurement," tech. rep., CMUS/SEI-2008-TR-012, 2008.
- [11] D. Lim, P. Anthony, H. C. Mun, N. K. Wai, "Assessing the Accuracy of Grey System Theory Against Artificial Neural Network in Predicting Online Auction Closing Price," in: *Proceedings of the International Multi Conference of Engineers and Computer Scientists 2008, Vol. 1 IMECS 2008*, (Hong Kong, China), pp. 19-24, 2008.
- [12] S. Liu, Y. Liu, *Grey information: Theory and Practical Application with 60 Figures*, London, England: Springer-Verlag, 2006.
- [13] L. Meng, W. Kees, "Grey System Theory and Applications: A way Forward," *The Journal of Grey System*, vol. 10, pp. 47-54, 2007.
- [14] R. S. Pressman, *Software engineering a practitioner's approach*, CityplaceNew York, country-regionUSA: McGraw Hill, 2005.
- [15] L. Sifen, J. Forrest, "The Current Developing Status on Grey System Theory," *The Journal of Grey System*, vol. 2, pp. 111-123, 2007.
- [16] T. Stevenson, D. Carrington, P. Strooper, N. Sharon, "An industry/university collaboration to upgrade software engineering knowledge and skills in industry," *The Journal of Systems and Software*, vol. 75, issues 1-2, pp. 29-39, 2005.
- [17] J. H. Stock, "Time Series: Economic Forecasting," in: N. J. Smelser, P. B. Baltes, eds., *International Encyclopedia of the Social & Behavioral Sciences*, Elsevier Science Ltd., pp. 15721-15724, 2001.
- [18] S. Suhk-Hook, "Fuzzy-Bayes Fault Isolator Design for BLDC Motor Fault Diagnosis," *International Journal of Control, Automation, and Systems*, vol. 2, no. 3, pp. 354-361, 2004.
- [19] Y. Xiong, "Grey Relational Evaluation of Financial Situation of Listed Company," *Journal of Modern Accounting and Auditing*, vol. 3, no. 2, pp. 41-44, 2007.
- [20] Y. Yang, R. John, "Grey Systems and Interval Valued Fuzzy Sets," in: *Proceedings of the 3rd Conference of the European Society for Fuzzy Logic and Technology*, (Zittau, Germany), pp. 193-197, 2003.
- [21] H. Zuse, *A Framework of Software Measurement*, Berlin, Germany: Walter De Gruyter Inc., 1997.



**Ninoslav Slavek** received B.Sc. degree in 1975, M.Sc. degree in 1987 in Electrical Engineering from the Faculty of Electrical Engineering in Zagreb, University of Zagreb, Croatia. Ph.D. degree he received from the University of Graz, Austria in 1998 with the thesis: Concept of Quality Management Information System for Software Production. He had been employed as an R&D engineer at the company Elektroprojekt Zagreb and was engaged in software projects and quality assurance aspects of nuclear power plant Krško,

Slovenia. From 1998 he worked in the company IBM as the PM. Since 2000 he has been with Faculty of Electrical Engineering Osijek where he is currently an assistant professor at the Department of Software Engineering. His research interests are: software quality assurance, development of embedded real-time systems and control software systems.



**Alan Jović** received B.Sc. degree in 2006 in Computer Science from the Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb. He is currently working towards his Ph.D. degree in Computer Science at FER Zagreb. From 2006 to 2007 he was employed at Rudjer Boskovic Institute on the FP6 European project Heartfaid, where he developed computer ontology to categorize terms related to congestive heart failure syndrome. He is currently working as research assistant with the Department of Electronics, Microelectronics, Computer and Intelligent Systems, at FER Zagreb. He is responsible for creating a Java-based computer framework for feature extraction from cardiac rhythm. His main research interests include: data mining and knowledge discovery, knowledge representation, and time-series modelling. He is particularly interested in application of artificial intelligence methods in biomedical research. He is a member of IEEE.

#### **AUTHORS' ADDRESSES**

**Asst. Prof. Ninoslav Slavek, Ph.D.**  
**University of Osijek**  
**Faculty of Electrical Engineering**  
**Department of Software Engineering**  
**Osijek, Croatia**  
**email: ninoslav.slavek@etfos.hr**

**Alan Jović**  
**University of Zagreb**  
**Faculty of Electrical Engineering and Computing**  
**Department of Electronics, Microelectronics,**  
**Computer and Intelligent Systems**  
**Unska 3**  
**Zagreb, Croatia**  
**email: alan.jovic@fer.hr**