

DOI: <https://doi.org/10.18454/RULB.2018.14.2.3>**АЛГОРИТМ ОБРАБОТКИ РЕЗУЛЬТАТОВ ПСИХОЛИНГВИСТИЧЕСКИХ ТЕСТОВ ДЛЯ
ФОРМИРОВАНИЯ ПРОЕКТНОЙ ГРУППЫ**

Научная статья

Панфилова О.А.¹, Полянская А.Д.^{2,*}, Валитова Ю.О.³^{1, 2, 3} Университет ИТМО, Санкт-Петербург, Россия

* Корреспондирующий автор (polyanskaya.anastasia[at]yandex.ru)

Аннотация

Формирование проектной группы является важной и актуальной задачей. Это связано с тем, что именно от того, насколько слаженно и эффективно будут работать участники проектной группы, зависят результаты ее работы. В статье предлагается еще одна методика формирования проектной группы. Она основана на учете метапрограмм, свойственных личности. Исследования показывают, что метапрограммы, проявляемые в речи, являются по сути установками, определяющими особенности поведения личности и ее реакции на события. Так как определение метапрограмм является сложным процессом, требующим привлечения экспертов и временных затрат, то разработка алгоритма, позволяющего автоматизировать формирование проектной группы и определять роль каждого из ее участников, является важной задачей, поскольку позволит значительно снизить временные затраты и зависимость от эксперта. В алгоритме можно выделить следующие этапы: предобработка; поиск заданных слов в тексте; формирование карточки метапрограмм; сравнение с шаблонами и присвоение роли; формирование проектной группы; визуализация. В статье подробно описаны все этапы работы алгоритма, а также приводятся результаты его тестирования. Результаты показали, что процент совпадения итоговых значений, полученных в ходе автоматизированной обработки, с мнением эксперта составил 89%. При этом были сформированы полные проектные группы. Проверка работы полученных проектных групп на практике была проведена для тестовой выборки. Результат оправдал ожидания – группы работали слаженно, оперативно и показали хорошие результаты.

Ключевые слова: психолингвистика, алгоритм, проектная группа, тестирование, автоматизация.

**ALGORITHM FOR PROCESSING OF THE RESULTS OF PSYCHOLINGUISTIC TESTS FOR THE
FORMATION OF A PROJECT TEAM**

Research article

Panfilova O.A.¹, Polyanskaya A.D.^{2,*}, Valitova Y.O.³^{1, 2, 3} ITMO University, Saint-Petersburg, Russia

* Corresponding author (polyanskaya.anastasia[at]yandex.ru)

Abstract

The formation of a project team is an important and urgent task. This is due to the fact that the results depend on how well and efficiently the participants of the project team work. The article suggests one more method of forming a project team. It is based on considering the meta-programs peculiar to the individual. Studies show that meta-programs are manifested in a speech of a person and can be defined as basic settings that determine the characteristics of his behavior and reaction to events. The definition of meta-programs is a complex process, requiring the involvement of experts and time-consuming, so the development of an algorithm that allows to form a project team and determine the role of each of its participants is an important task. It will allow people to work with meta-programs even if they are not experts. The algorithm can be divided into the following stages: preprocessing; search for given words in the text; formation of a meta-program card; comparison with templates and role assignment; formation of a project team; visualization. The title describes in detail all the steps of the algorithm. Also the results of testing the algorithm are given. The test results showed that percentage of match with the expert's opinion was 89%. In this case, complete project teams were formed. The verification of the work was carried out for the test sample. The result met expectations – the teams worked smoothly, quickly, and showed good results.

Keywords: psycholinguistics, algorithm, project team, testing, automation.

Introduction

Development of modern information systems and software applications is not an easy task. This requires not only skills of using of various software implementation tools, but also a large number of professional competence and only a small part of them can be categorized as basic. Others are expert. They are interdisciplinary and reflect complex aspects of the development of information systems. More than that some of the competence such as design development, content creation, that are necessary during the development of modern information systems can be hardly attributed to the competence of a programmer.

Therefore, development of most information systems is carried out by project teams. The project team is a temporary organizational formation of people with a leader at the head. It includes all the necessary specialists and allows activities usually aimed at achieving one common goal.

It should be mentioned that members of project teams should not only carry out assigned tasks, but also make decisions on design issues and other occurring problems.

That's why accuracy of selection of project team members is one of the most important features that determines efficiency of design and implementation of information systems and quality of the work of all project

team. This statement is confirmed by the results of the analysis of studies on this problem and by questioning of employers and members of project team.

Despite the fact that there are methods for the formation of project teams, there is no method or a group of methods that are currently acknowledged. Also, the analysis shows that this problem is still relevant.

One of the possible ways to solve the problem is to define a role in a project team based on psycholinguistic tests.

The most effective way to do this is to process the results of the test automatically, because manual processing requires a lot of efforts as well as special knowledge.

General description of the methodology.

The meta-program method was chosen as the basis for determining the personal qualities corresponding to a particular role in the project team. Meta-programs can be identified as unconsciously used ways of thinking of the person underlying his behavior. Being the attitude, they determine the specific of perception of surrounding world and the ways of response to internal and external stimuli [5], [9], [11].

One of the features of meta-programs is that they are almost impossible to simulate [6], [7]. A set of identified metaprograms defines a meta-program portrait of a person.

Meta-programs can be defined by several ways, but the most simple and fast is the study of human speech. It is possible to use both oral and written speech. Taking into account further automated processing to define meta-programs, we chose written form of a test. The questions were formulated as open-ended in order to get detailed answers. For example:

- You were charged to execute a new project. What will you start with?
- Do you successfully enter a new team? Why do you think so?
- Your colleague slightly breaks the internal regulations. What would you do?

Some of the questions were related to the behavior in the professional field as the results of defined meta-programs will be used to determine a role in a project team [4].

After holding the test, the process of interpretation of the results can be started.

The processing of the results of a test consists in finding specific words or phrases in responses of respondents. It should be taken into account that a set of tokens, we are looking for, is different for each meta-program. Therefore, each question in a test is related to a certain meta-program [8].

Most of meta-programs are represented in pairs as two opposite meanings (characteristics) that reflects diametrically opposed ways of human thinking. Also one of the intermediate values can be obtained. This means that characteristics of a meta-program are expressed equally, or one of them prevails. The degree of expression of each characteristic depends on the number of words or phrases found in a text and related to a particular way of human thinking.

Counting of found words allows to define the meaning of meta-program.

Knowing the meaning of each meta-program, it is possible to create meta-program portrait of a person [1], [3].

As each role in a project team has a certain degree of expression of characteristics in a meta-program portrait we can define a role which corresponds to a person, just by comparing the meta-program portrait of the person with the

templates of meta-program portraits of roles in the project team.

Implementation of this actions is rather difficult process, so it is necessary to develop an algorithm that allows to create a project team using the methodology described above.

Description of the algorithm

To implement the algorithm, it is necessary to prepare a file with the answers of respondents, which the algorithm will work with.

The designed algorithm can be conditionally divided into six stages:

1. Preprocessing.
2. Search for the key words in the text.
3. Forming a meta-program card.
4. Comparison with patterns and defining the role.
5. Forming of a project team.
6. Visualization.

Let's consider each stage in detail.

1. Preprocessing.

The purpose of this stage is to bring the text to a particular state that is suitable for processing. It is necessary to create a dictionary with stop words that should be removed from the text at this stage and with the key words to search in the text. Also, a file with patterns of meta-programs personal cards should be created.

After the data is loaded the preprocessing starts. It is carried out through regular expressions and includes removal of punctuation marks and stop words: conjunctions, prepositions, particles and partly lemmatization that is the reduction of a word to the infinitive. Lemmatization is performed only for nouns, adjectives, participles and gerunds. Verbs do not lead to an infinitive, because according to the method of meta-programs the verb form has a value and affects the result.

2. Search for the key words in the text.

In the preprocessed text, it is necessary to find certain words and / or phrases that determine the meaning of the meta-program. The set of tokens is different for each meta-program, so the answers to each of the questions need to be connected with words or phrases related to them. After analyzing the ready-made dictionaries available in the public domain, we stated that they do not approach the task at hand. Consequently, such a dictionary needs to be developed and that was done [10]. The dictionary is uploaded at the first stage. The dictionary is grouped by its belonging to a specific meta-program and most of words are verbs. The pronouns "I", "we", "they" are also analyzed.

The search for words occurs sequentially on a dimensionless array of strings arrays «Arraylist<Arraylist<String>>». The data is processing separately for each group of answers corresponding to one meta-program. In groups, questions are also analyzed in turn.

3. Forming a meta-program card.

After performing a dictionary search, one of the two characteristic values is assigned for each question, depending on the number of words that were found and their attributions to a particular characteristic.

Next, the final value for the group of questions is determined, which is stored as an integer and written as the first element of the string. Definition of the final value are carried out according to the following formula:

Let us denote the first characteristic as x_1 , and the second characteristic as x_2 (e.g., x_1 – activity and x_2 – reflectivity), then, if:

$x_1 = 0$, then we have the boundary value of the second characteristic.

$x_2 = 0$, then a boundary value x_1 is obtained.

$x_1 = x_2$, then we write the intermediate value.

$x_1 > x_2$ и $x_1 \neq 0$ и $x_2 \neq 0$, so x_1 prevails, otherwise – x_2 prevails.

Similarly, the remaining groups of questions are processed.

As a result, a sequence of values of characteristics is obtained, which together form the card of the subject or a card of meta-programs.

4. Comparison with templates and role assignment.

In order to determine the role in the project team based on the answers, the subject's card obtained in the previous step is compared with the templates available in the database, each of which corresponds to a certain role in the project team.

The templates of meta-program cards with all possible variants of distribution of characteristic values for each role

Characteristic	1	2	3	4	5	Characteristic
Activity						Reflectivity
External reference						Internal Reference
Motivation to						Motivation from
Out of time						In time

Fig. 1 – A pattern of card of meta-programs

5. Forming the project team.

Formation of the project team, that is, directly grouping employees by five to six people.

The project team is formed as follows: first, the array of people is consistently reviewed for a single match with each role, without a group and the preservation of the position of the person with this role. As soon as we reach six matches, the group number is written in the last field of the array of these people, which is then incriminated. The same thing happens as to groups of five people, without an expert and an architect. After that, the array is reviewed again.

In the case of a situation where the test results lack one or more roles, or there are too few experts in the given specialty, there is a "lowering the role" algorithm. It consists in the following. After determining the roles by the results of testing while forming the groups, if the necessary roles are

in the project team must be developed in advance. The basis for this is an analysis of the intensity of the characteristics for each project role. An example of a meta-program map is shown in Fig. 1.

Next, for convenience of processing, the cards were encoded as a four-digit numerical sequence. Each element of the sequence is encoded as an integer from one to five. For the example shown in Figure 1, this sequence will look like 1423.

Comparison of the subject's card with the template goes by character. At the same time, a partial match of three characters and a full match of four characters are searched. If a full match is not found, then the first match of the three characters is retained.

If the meta-program card matches the template by more than 75 percent, then the role assigned to the template is considered the most suitable and assigned to the subject.

absent/not present in sufficient quantity (in the case when it is necessary to form more than one group), then the roles are transformed according to the following schemes:

- leader → analyst → expert → software developer / tester / system architect;
- analyst → leader.

After that, an array of people is reviewed again.

6. Visualization

At this stage, the user is presented with the results of the algorithm. The results can be presented in the form: metaprogram cards for each subject, the name of the role obtained from the test results, the final role in the project and the number of the project team; composition of project teams.

The schematically described algorithm is shown in Fig. 2.

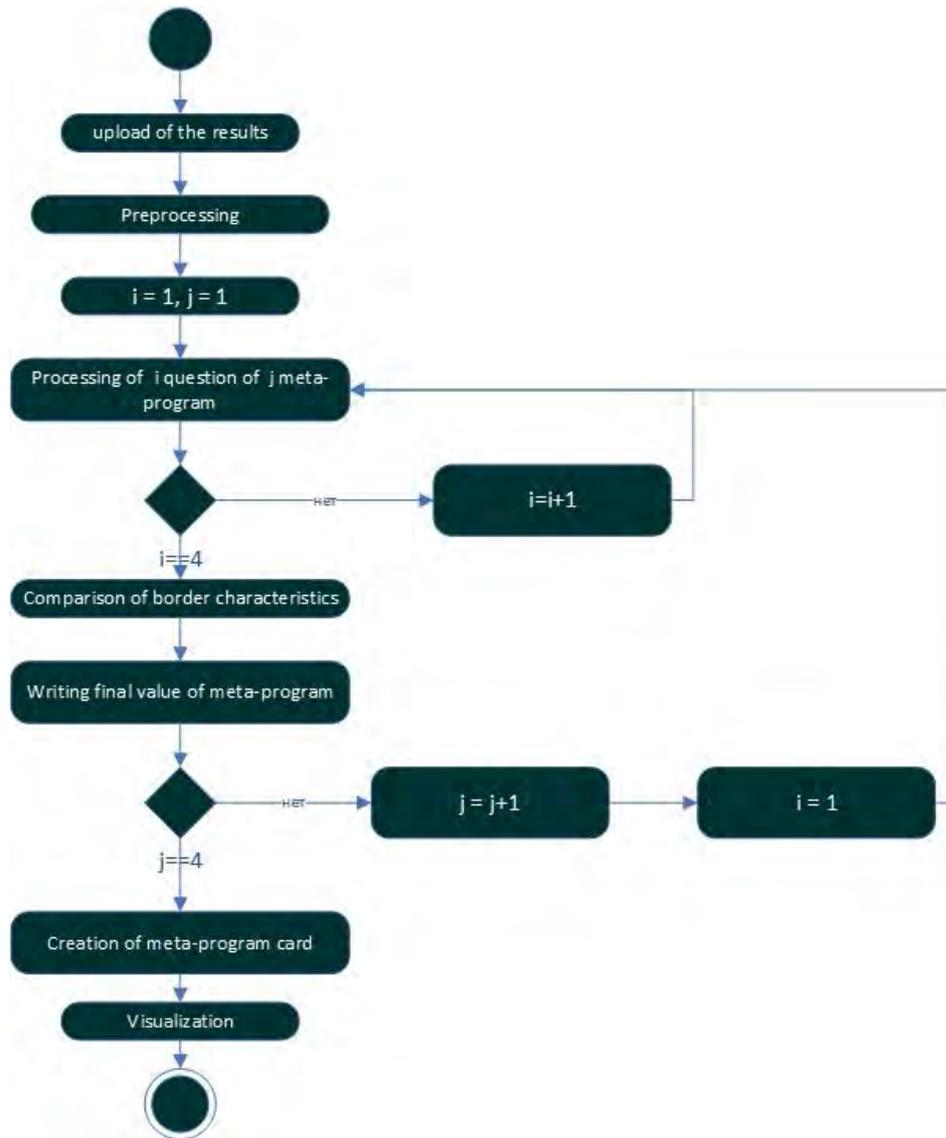


Fig. 2 – Algorithm of processing meta-programs

3. Experimental verification of the algorithm

To test the algorithm, a console application was developed in the Java language. As the compilation environment, IntelliJ IDEA's integrated development environment is used.

The main third-party library used for development is Apache POI. It allows you to work with MS formats, in particular *.xls and *.xlsx. Built-in libraries are also used [2].

The main tool for forming internal arrays is a dimensionless array of lines of the array of lines: `ArrayList<ArrayList>`. All comparisons occur using standard cycles (foreach, for, while).

For the operation of the algorithm, the following were developed:

- list of questions that allow us to define four meta-programs of personality;
- dictionary of lexemes, grouped by belonging to a specific meta-program;
- meta-program card;
- templates of metaprogram cards with all possible variants of the distribution of characteristics corresponding to the roles in the project team encoded in the form of a four-digit numerical sequence;
- templates for output the results of the algorithm operation.

At the first stage, a survey was conducted on the developed questionnaire. The survey was conducted remotely. You could take part in the survey only once, and you cannot go back and change the answer. The results were saved in a table.

The survey was conducted in two stages: first, data was collected for the training sample, then for the test sample. People of different ages and sex took part in it. After that, the algorithm was tested.

The results of the algorithm work were compared with the results achieved by the expert.

Based on the results of the first testing of the algorithm, the percentage of matches with the expert distribution was 48%. The analysis showed that the main reason for the incorrect definition of characteristics in some cases was the insufficient number of words in the dictionary. In this regard, it was decided to expand the dictionary, supplementing it with a large number of words. Also, a set of role templates was enhanced, and code optimization was performed.

These actions enabled that at a repeated testing of the algorithm the percentage of match with the expert's opinion on the training sample amounted to 91%.

In the training sample, people were identified belonging to all the determinable roles, which allowed the formation of the project team. The project team based on the training sample was formed from six people with the most intensive

characteristics and with partially similar characteristics: all members of the group have a high degree of motivation and organization; three members of the group are active, three are more reflective. Of the remaining five people, no group was formed, because not all the necessary roles were present.

On the test sample the percentage of match turned out to be slightly less and amounted to 88%, at the same time, there was a complete mismatch of three results with the expert distribution. After the distribution of the roles, project groups were formed.

At the end, the entire set of data was also processed. The percentage of match with the expert's opinion was 89%, and full project teams were formed.

The obtained result of the algorithm operation is good, the error does not exceed 15%. When processing text data, it is extremely difficult to achieve one hundred percent result, because it is quite difficult to formulate a dictionary that takes into account all possible variants of words used in answering each of the questions.

The composition of the received project groups matched with the expert distribution by 87%, which can be considered a fairly good result. Verification of the work of the received project teams in practice was carried out for the test sample. The result met expectations – the groups worked smoothly, quickly, and showed good results.

Conclusion

In the framework of the research the algorithm for the automated formation of the project team based on meta-program method was designed and tested. The results of the research make it possible to state that the use of the algorithm and the developed materials allows effectively to form a project team and / or to determine the role of a particular person in it.

In this case, the use of an automated algorithm has a number of advantages over manual processing:

- the speed of data processing (for an expert it takes about 5 minutes to process one answer while the algorithm processes all results in 3,6 seconds);

- the possibility to process data and interpret the results without an expert as the use of special knowledge is no longer required;

- the ability to verify accordance to different roles in the project team without additional processing of the results.

At present, work to increase the number of meta-programs used to form project teams is held. Moreover, the dictionary of key words is being modified to improve the accuracy of the algorithm and, accordingly, the accuracy of defining of metaprogram values.

The results of the research can be used not only for the formation of project teams, but also for figuring out personal qualities.

Конфликт интересов

Не указан.

Conflict of Interest

None declared.

Список литературы / References

1. Глухов В. П. Психоллингвистика. Теория речевой деятельности / В. Глухов, В. Ковшиков – М.: Астрель, 2007. – 589 с.
2. Лафоре Р. Структуры данных и алгоритмы в Java / Р. Лафоре – СПб.: Питер, 2016. – 704 с.
3. Любимов А. Метапрограммы в схемах / А. Любимов – М.: Sentertrenings, 2015. – 30 с.
4. Adler H. Handbook of NLP: A Manual for Professional Communicators / H. Adler – UK: Gower Publishing, 2002 – 320 p.
5. Dmitrieva N. V. Scientific and Theoretical Aspects of the Staff Recruitment Organization within the Concept of "Talent Management" / N. V. Dmitrieva, N. A. Zaitseva, O. S. Kulyamina, A. A. Larionova, S. A. Surova // Asian Social Science journal – Canada: Canadian Center of Science, 2015. – P. 358-365.
6. Hall M. L. The User's Manual for the Brain / M. L. Hall, B. C. Bodenhamer. – Wales, UK: Crown House Publishing Limited, 2014. – 640 p.
7. Hall M. L. Figuring Out People: Design Engineering with Meta Programs / M. L. Hall, B. C. Bodenhamer – Wales, UK: Crown House Publishing Limited, 2007. – 320 p.
8. Kolarzowski J. J. Preferred thinking patterns — Metaprograms / J. Kolarzowski – Semiotyczne, 2001. – 228 p.
9. Рогозина И.В. Контекстное использование методов нейро-лингвистического программирования (НЛП) в оценке персонала организации // Информационные технологии в профессиональном образовании, научной и библиотечно-информационной деятельности. - СПб.: СПбГУКИ, 2008. - С. 224-227.
10. Valitova Y. O. Development of a dictionary for information system of automated figuring out person's meta-programs profile / Y. O. Valitova, O. A. Panfilova // Russian Linguistic Bulletin. – 2017. – No. 2(10). – Pp. 65-68.
11. Vickers A. Essential NLP: Teach Yourself / A. Vickers, S. Bavister – UK: Hachette, 2010 – 336 p.

Список литературы на английском / Список литературы латинскими символами / References in English

1. Glukhov V.P. Psiholingvistika. Teoriya rechevoj deyatelnosti [Psycholinguistics. Theory of speech activity] / Vadim Glukhov, Valery Kovshikov - Moscow: Astrel, 2007. [in Russian]
2. Lafore R. Struktury dannykh i algoritmy v Java [Data structures and algorithms in Java] / Robert Lafore – SPb.: Piter, 2016. – 704 p. [in Russian]
3. Lafore R. Struktury dannykh i algoritmy v Java [Data structures and algorithms in Java] / Robert Lafore – SPb.: Piter, 2016. – 704 p. [in Russian]
4. Adler H. Handbook of NLP: A Manual for Professional Communicators / H. Adler – UK: Gower Publishing, 2002 – 320 p.
5. Dmitrieva N. V. Scientific and Theoretical Aspects of the Staff Recruitment Organization within the Concept of "Talent Management" / N. V. Dmitrieva, N. A. Zaitseva, O. S. Kulyamina, A. A. Larionova, S. A. Surova // Asian Social Science journal – Canada: Canadian Center of Science, 2015. – P. 358-365.

6. Hall M. L. *The User's Manual for the Brain* / M. L. Hall, B. C. Bodenhamer. – Wales, UK: Crown House Publishing Limited, 2014. – 640 p.
 7. Hall M. L. *Figuring Out People: Design Engineering with Meta Programs* / M. L. Hall, B. C. Bodenhamer – Wales, UK: Crown House Publishing Limited, 2007. – 320 p.
 8. Kolarzowski J. J. *Preferred thinking patterns — Metaprograms* / J. Kolarzowski – Semiotyczne, 2001. – 228 p.
 9. Rogozina I.V. *Kontekstnoe ispol'zovanie metodov nejro-lingvisticheskogo programmirovaniya (NLP) v ocnke personala organizacii* [Contextual use of methods of neurolinguistic programming (NLP) in the assessment of the organization's personnel] / I.V. Rogozina // *Informacionnye tekhnologii v professional'nom obrazovanii, nauchnoj i bibliotechno-informacionnoj deyatel'nosti* [Information technologies in professional education, scientific and library-information activities]. – Saint-Petersburg: Publishing house of the St. Petersburg State University of Culture and Arts, 2008. – P. 224–227. [in Russian]
 10. Valitova Y. O. *Development of a dictionary for information system of automated figuring out person's meta-programs profile* / Y. O. Valitova, O. A. Panfilova // *Russian Linguistic Bulletin*. – 2017. – No. 2(10). – Pp. 65-68.
 11. Vickers A. *Essential NLP: Teach Yourself* / A. Vickers, S. Bavister – UK: Hachette, 2010 – 336 p.
-
-