THE MULTIFACTOR MATHEMATICAL MODEL FOR CONSTRUCTING A MULTITUDE OF CONSISTENT EDUCATIONAL PATHS FOR TRAINING FULL STACK SPECIALISTS

Vyacheslav Kozlov
Samara State Technical University, Russian Federation

Alexander Guryanov
Samara State Technical University, Russian Federation

Elena Alontseva
Samara State Technical University, Russian Federation

Abstract. *The modern labor market, especially in the field of information technology startups, requires the training of a sufficiently large number of specialists with competencies in creating the minimum viable product (MVP). The educational organization should be able to quickly form the trajectory of specialist training based on the challenges of the labor market, taking into account the dynamics of its change, while maintaining the integrity and consistency of the educational program. The mathematical model of the formation of an educational program is proposed taking into account a large number of variable parameters, which allows one to construct many possible training paths for specialists and select the optimal ones according to the criteria of cost, efficiency or laboriousness.*
Keywords: *educational program, full stack specialist, information technology, minimum viable product, multifactor mathematical model, startup.*

Introduction

In the present work, the term "full-stack" of a specialist will mean a person who is able, alone, without involving other specialists, to solve the problem of a complex production task. For example, in the context of web development (hereinafter, examples of specific competencies will be given from the field of web development), this is a person who is able to develop a small website application completely on his own - from a technical task to a result worthy of an act of work (Gurcan & Köse, 2017).

The modern labor market, especially in the field of information technology startups, requires the training of a sufficiently large number of specialists with competencies in creating the minimum viable product (MVP) (Münch et al., 2013; Bosch, Olsson, Björk, & Ljungblad, 2013). The demand for such specialists has

been continuously growing since 2013, as evidenced by the growth of search queries (https://trends.google.com, 100 points - maximum popularity) in Fig. 1, while the need for highly specialized specialists is not only not growing, and has a downtrend.

This is due to the fact that full development is rapidly gaining momentum, and full-stack developers are becoming very popular in some companies (Guryanov, Kozlov, & Zhuravliova, 2019; Savoskina, Domnina, & Kozlov, 2017). A study by Stack Overflow 2017 showed that this type of developer is not only the most popular, but also the most popular. The demand, especially in small companies, for full-stack developers is related to the specifics of these companies, for example, the "start-up" movement (Dennehy et al., 2019), and the fact that having one person with several skills rather than several people with specific skills is of real value to many organizations. In addition, there is a big time savings, as a developer who can switch between levels and understand the whole process.
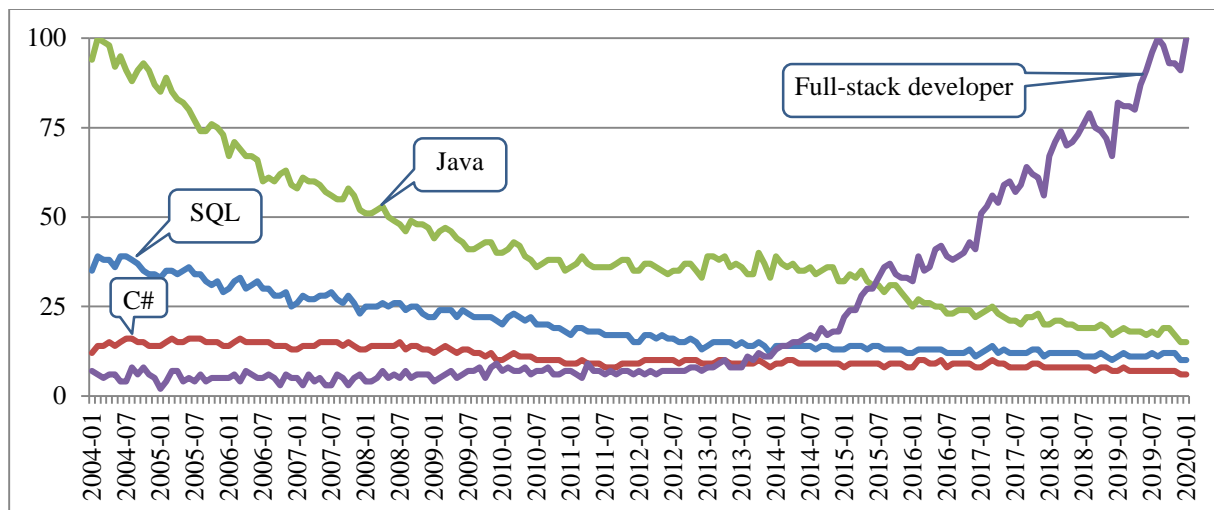

Figure 1 **Search query dynamics**

General developers work in all layers of software. They understand the principles and can work on both sides, although they do not always master all the subtleties as their highly specialized colleagues (Kozlov & Nasyrov, 2014).

Model

The development of a full-stack specialist training plan requires the creation of a specialized mathematical model that should take into account a lot of factors and form a coordinated stack of the technologies studied, covering all the necessary competencies for the chosen training area and should have developed

monitoring of the quality of training (Kozlov, Alontseva, & Guryanov, 2019; Yusupova, Alontseva, Kozlov, & Kulakova, 2017).

Let a set of knowledge, skills and competencies related to the direction of training. Obviously, some competencies require preparatory steps in mastering the previous competencies, for example, it is obvious that the "access to databases" competency requires a preliminary study of the SQL data access language and some programming language for which the database is bound. So for a professional in the field of web development, the set will be as follows ("System Administrator" Magazine, 2016 / Issue No. 01-02 (p. 158-159), "Vacancy: Full Stack-Web Developer"), for understanding, the full minimum list of competencies of the developer of WEB applications (Park & Wiedenbeck, 2011; Northwood, 2018):

- Understanding the principles of networks and cloud technologies;
- Ability to design and develop APIs and integrate;
- Basic experience in administering operating systems;
- Knowledge and ability to work with relational DBMS;
- Knowledge of the working principles of the HTTP protocol and the network stack as a whole;
- Ability to work with WEB servers (understanding of their differences, advantages, features, etc.);
- Knowledge of server programming language (one or more);
- Knowledge of server frameworks (one or more);
- Experience working with modern IDEs;
- Knowledge of the client layer at an average level;
- Ability to work with version control system;
- Ability to work with caching systems;
- Ability to work with NoSQL database;
- Skills of development of highly loaded services. Understanding scalability;
- Experience in writing Unit tests.
- Knowledge of algorithms;
- Understanding User Experience;
- Knowledge of flexible development methods and ability to work in a team;
- Ability to work with customers and to formulate technical requirements from business requirements;
- The ability to design and develop data layers and business logic, arguing for architectural decisions.

Let B_j - a set of technologies, and the value C_{ij} determines the relationship *i-th* of that competency from the set A_i and *j-th* of that technology from the set.

$$C_{ij} = \begin{cases} 1, & if\ technology\ j\ provides\ competency\ i \\ 0, & if\ technology\ j\ is\ not\ related\ to\ competency\ i \end{cases} \tag{1}$$

Due to the large size of the matrix C_{ij}, we present in table (Table 1) only a fragment of it (for clarity, zero elements are not indicated).

The components of the vector B_j are binary variables, defined as:

$$B_j = \begin{cases} 1, & if\ technology\ j\ is\ being\ studied \\ 0, & if\ technology\ j\ is\ not\ studied \end{cases} \tag{2}$$

Obviously, each competency must be supported by at least one technology, therefore, the inequality that determines the formation of all competencies will be of the form:

$$\sum_j C_{ij}B_j \geq P_i\ for\ \forall\ i \tag{3}$$

Table 1 **Technology and competency interconnection C_{ij}**

Competencies A$_i$	Technologies B$_j$																
	Visual Studio	Zend Studio	Net Beans	Eclipse	ODBC	Linq	Hibernate	HTML	CSS	JavaScript	PHP	C#	Java	DOT NET	Spring	YII	Laravel
Knowledge of server frameworks														1	1	1	1
Knowledge of server programming language											1	1	1				
Mid-level knowledge of the client layer								1	1	1							
Knowledge and ability to work with relational DBMS					1	1	1										
Experience with modern IDE	1	1	1	1													

where P_i sets the minimum number of technologies that ensure the formation of the *i-th* competency. Note that an additional matrix V_{ij} can be introduced whose elements will determine the value (weight) of the technology for competency *i*. Then the inequality that determines the formation of all competencies can be written as:

$$\sum_j V_{ij}B_j \geq P_i \ for \ \forall \ i \tag{4}$$

In this case P_i sets the minimum level of formation of the *i-th* competence.

Obviously, the technologies are interdependent, that is, individual technologies cannot be studied without a preliminary study of the supporting technologies. For example, learning the YII or Laravel framework involves learning the PHP language, but the reverse is not true. Thus, the interconnection of technologies can be represented as a directed graph or as a square asymmetric adjacency matrix. Let D_{jk} (the index k varies within the same limits as the index j) characterizes the dependence of j that technology on k and is given by the rule:

$$D_{jk} = \begin{cases} 1, & if\ technology\ j\ requires\ knowledge\ of\ technology\ k \\ 0, & if\ technology\ j\ is\ independent\ of\ technology\ k \end{cases} \tag{5}$$

Table 2 shows an example of a matrix D_{jk} as applied to the technologies described in table 1.

*Table 2 **Technology adjacency Matrix D_{jk}***

Competencies A_i	Technology B_j																
	Visual Studio	Zend Studio	Net	Eclipse	ODBC	Linq	Hibernat	HTML	CSS	JavaScri	PHP	C#	Java	DOT NET	Spring	YII	Laravel
Visual Studio																	
Zend Studio																	
Net Beans																	
Eclipse																	
ODBC																	
Linq														1			
Hibernate													1				
HTML																	
CSS								1									
JavaScript								1	1								
PHP																	
C#	1																
Java																	
DOT NET												1					
Spring											1		1				
YII											1						
Laravel											1						

The compatibility condition of the studied technologies can be written as a system of inequalities:

$$D_{jk}B_j \geq B_j \; for \; \forall j, k \qquad (6)$$

However, not so unambiguous connections of technologies are possible, but the requirements of studying one or more technologies from a certain family, for example, Java technology requires knowledge of one of the modern IDEs - Net Beans or Eclipse. We introduce the concept of a family of technologies and number them. Each technology can be assigned to one or more families, through matrix M_{nj} (Table 3).

$$M_{nj} = \begin{cases} 1, & if \; technology \; j \; belongs \; to \; the \; family \; n \\ 0, & if \; technology \; j \; is \; not \; part \; of \; the \; family \; n \end{cases} \qquad (7)$$

*Table 3 **Technology joining families M_{nj}***

Technology family number	Technology B_j																
	Visual Studio	Zend Studio	Net Beans	Eclipse	ODBC	Linq	Hibernate	HTML	CSS	JavaScript	PHP	C#	Java	DOT NET	Spring	YII	Laravel
1 (IDE for Java)			1	1													
2 (IDE for PHP)		1		1													
3 (Programming languages)											1	1	1				

Then we introduce R_{jn} as a matrix whose elements determine the dependence of the *j-th* technology on the *n-th* family of technologies (Table 4).

$$R_{nj} = \begin{cases} 1, if \; technology \; j \; requires \; at \; least \; one \; technology \; of \; the \; family \; n \\ 0, & if \; technology \; j \; does \; not \; depend \; on \; technology \; n \end{cases} \qquad (8)$$

*Table 4 **Technology dependence on technology families R_{nj}***

Technology family number	Technology B_j																
	Visual Studio	Zend Studio	Net Beans	Eclipse	ODBC	Linq	Hibernate	HTML	CSS	JavaScript	PHP	C#	Java	DOT NET	Spring	YII	Laravel
1 (IDE for Java)													1				
2 (IDE for PHP)											1						
3 (Programming languages)					1												

Then the value $\max_j(M_{nj}B_j)$ will be equal to 1 if at least one of the technologies of the family n and 0 is studied otherwise. To ensure resolution of the dependence of the technologies under study on technology families, the following system of inequalities must be implemented:

$$B_j \max_j(M_{nj}B_j) \geq B_j \; for \; \forall j,n \;\; or \;\; B_j R_{nj} \sum_j M_{nj} B_j \geq B_j \; for \; \forall j,n \qquad (9)$$

For clarity, you can display information on technology relationships from tables 2-4 in graphical form Fig. 2.

The solid lines in the Fig. 2 show the relationships from table 2 (unambiguous dependence), and the dashed lines show the data from tables 3 and 4 (requirements of at least one technology from the family). The arrows indicate the dependent technology to the supporting one. You may notice that in Fig. 2 some technologies or technology blocks are not connected to each other, which corresponds to independent technologies or technology blocks.
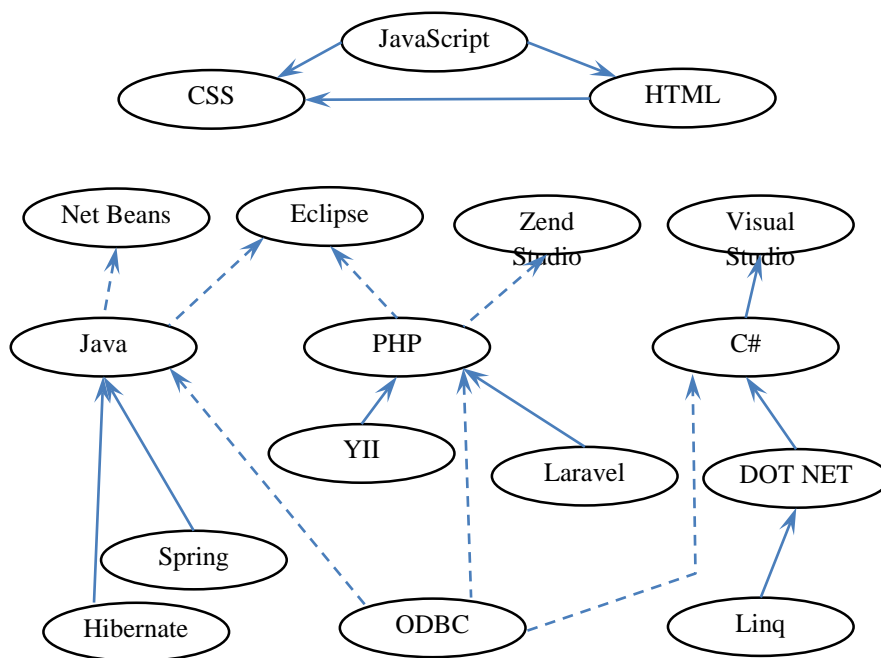

*Figure 2 **Technology interconnection graph***

So, it can be argued that any set of technologies studied, defined by a vector B_j and satisfying conditions (6 and 9), is joint, and if conditions (3) are satisfied, it is complete, that is, it provides the formation of all the required competencies for a given training profile and represents the set of possible training paths for specialists.

We turn to the problem of choosing the optimal training path. To do this, we introduce a restriction on the maximum complexity of the preparation. Let the vector H_j determine the complexity in hours of studying j of that technology. When the task of choosing the optimal trajectory from the position of minimizing training costs is to solve the optimization problem (10) subject to conditions (3, 6 and 9):

$$\sum_j H_j B_j \rightarrow min \qquad (10)$$

If it is necessary to choose the most useful trajectory for training specialists, equation 10 takes the form of a restriction (where H^{max} is the maximum available training time):

$$\sum_j H_j B_j \leq H^{max} \qquad (11)$$

By the criterion of the usefulness of the educational trajectory, we understand the degree of demand for specialists trained on this trajectory by the labor market. To do this, using recruiting data (job site in the field of information systems and technologies https://www.dice.com, January 7, 2020), we will build table 5, taking into account the data from table 1. Table 5 in the last column gives the normalized frequency of references. Rationing of the frequency of technology references was carried out within the framework of competencies to which these technologies relate. This method allowed us to come to comparable values and get rid of the economies of scale of the demand for competencies, which is absolutely justified, since condition (3) guarantees training in all competencies of the specialty. Thus, the last column of table 5 is a vector of technology E_j demand.

*Table 5 **The usefulness of technology in the labor market***

Technology	Number mention	Competence	Number of mention technologies within the competence	Normalized frequency of references,% (The vector of the utility of technology in the labor market E_j)
Visual Studio	269	experience with modern IDEs	404	67
Zend Studio	20			5
Net Beans	4			1
Eclipse	111			27
ODBC	37	knowledge and ability to work with relational DBMS	685	5
Linq	71			10
Hibernate	577			84

HTML	2972	knowledge of the client layer on average level	12394	24
CSS	1513			12
JavaScript	7909			64
PHP	832	knowledge of server programming language	14389	6
C#	3017			21
Java	10540			73
DOT NET	2288	knowledge of server frameworks	4494	51
Spring	2112			47
YII	25			1
Laravel	69			2

Then the choice of the most useful training path is reduced to the search for maxim (12), subject to conditions (3, 6, 9 and 12):

$$\sum_j E_j B_j \rightarrow max \qquad (12)$$

To set necessarily studied technologies, for example, for a specific employer, condition (13) can be introduced.

$$B_j \geq B'_j \ for \ \forall j \qquad (13)$$

B'_j sets the required technologies to be studied by unit values.

Results and Conclusions

A multifactor mathematical model is developed for constructing a multitude of consistent educational paths for training full stack specialists, the formal formulation of which is presented below

Initial Data
A_i – a set of required competencies,
B_j – a set of available technologies,
C_{ij} – connection i of that competence and j of that technology,
V_{ij} – the significance (weight) of technology j for competence i,
D_{jk} – the dependence of technology j from k,
M_{nj} – the relationship of technology j and technology family n,
R_{jn} – he dependence of j hat technology on the nth family of technologies,
E_j – vector demand for technology by the labor market.

Optimization Parametrs
P_i – minimum level of formation of *i-th* competency,

H_j – the complexity of j technology,

$\text{H}^{\max}$ – the most labor intensive educational trajectory,

B_j' – compulsory technology.

Limitations

$\sum_j V_{ij} B_j \geq P_i \; for \; \forall \, i$ – competency formation,

$D_{jk} B_j \geq B_j \; for \; \forall j, k$ – compatibility (consistency) of technologies,

$B_j R_{nj} \sum_j M_{nj} B_j \geq B_j \; for \; \forall j, n$ – compatibility (consistency) of technologies and technology families,

$\sum_j H_j B_j \leq H^{max}$ – restriction on the complexity of training,

$B_j \geq B_j' \; for \; \forall j$ – guaranteed study of required technologies.

Optimization Variables

B_j – binary vector trajectory preparation.

Objective Function

$\sum_j \text{E}_j \text{B}_j \; \rightarrow \; \max$ – efficiency from the standpoint of demand.

$\sum_j H_j B_j \; \rightarrow \; min$ – minimization the complexity of training.

Solved tasks:

- The construction of many joint and complete trajectories of specialist training;
- Search for the optimal trajectory of the training of specialists;
- Searching for the most useful trajectory of specialist training with a limited labor intensity.

A mathematically sound technology has been obtained that allows us to develop training plans for specialists that cover all the necessary competencies for the chosen field of training on the basis of a specialized multifactor mathematical model for constructing a multitude of consistent educational ones. This allows us to satisfy the need for such specialists, which has been continuously growing since 2013, especially in small high-tech companies and startup projects.

References

Bosch, J., Olsson, H., Björk, J., & Ljungblad, J. (2013). The early stage software startup development model: A framework for operationalizing lean principles in software startups. *Lean Enterprise Software and Systems. Springer, 167*, 1–15.

Dennehy, D., Kasraian, L., O'Raghallaigh, P., Conboy, K., Sammon, D., & Lynch, P. (2019). A Lean Start-up Approach for Developing Minimum Viable Products in an Established Company. *Journal of Gecision Systems, 28*(3), 224-232.

Gurcan, F., & Köse C. (2017). Analysis of software engineering industry needs and trends: *Implications for education. International Journal of Engineering Education, 33*(4), 1361-1368.

Guryanov, A., Kozlov, V., & Zhuravliova, O. (2019). Arranging University Educational Process based on the Regional Competence Profile. *Society, Integration, Education. Proceedings of the International Scientific Conference*, *I*, 205-212. DOI: http://dx.doi.org/10.17770/sie2019vol1.3829

Kozlov, V., Alontseva, E., & Guryanov, A. (2019). Information and Educational Environment for Monitoring and Forming a Cumulative Assessment of Academic Work in a Semester. *Society, Integration, Education. Proceedings of the International Scientific Conference. V*, 383-392. DOI: http://dx.doi.org/10.17770/sie2019vol5.3828

Kozlov, V., & Nasyrov, M. (2014). Automated information system of active positioning of students in the labor market. *International Research Journal, 3-2* (22), 25-26.

Münch, J., Fagerholm, F., Johnson, P., Pirttilahti, J., Torkkel, J., & Jäarvinen, J. (2013). Creating Minimum Viable Products in Industry-Academia Collaborations. *Lean Enterprise Software and Systems. Proceedings of 4th International Conference*, 137-151.

Northwood, C. (2018). *The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer*. Retrieved from http://www.xn--101-8cdo3any.xn--p1ai/pdf/The_Full_Stack_Developer.pdf

Park, T., & Wiedenbeck, S. (2011). Learning web development: Challenges at an earlier stage of computing education. *Proceedings of the Seventh International Workshop on Computing Education Research, USA, August 8-9*, 125-132

Savoskina, E., Domnina, S., & Kozlov. V. (2017). Mehanizm sozdanija regional'nogo kompetentnostnogo profilja vypusknika vuza. *Izvestija Samarskogo nauchnogo centra Rossijskoj akademii nauk. Social'nye, gumanitarnye, mediko-biologicheskie nauki, 19*(4), 41-45.

Yusupova, O., Alontseva, E., Kozlov, V., & Kulakova, N. (2017). Experience of a pilot project for the implementation of a score-rating system in a technical university. *In the collection: Traditions and innovations in construction and architecture. Social and humanitarian and economic sciences. Collection of articles: electronic resource. Samara State Technical University. Samara*, 150-153.