A Roadmap for Implementing the Sustainable Software Engineering Curricula in Ukraine According To European E-Competence Framework and Student-Centered Learning Conception

Ihor Turkin [1][0000-0001-6456-1120] and Olena Kirilenko[2][0000-0002-4496-3787]

[1] Software Engineering and Business Faculty, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine
[2] Software Engineering and Business Faculty, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine
i.turkin@khai.edu., o.kirilenko@khai.edu.

Abstract. The paper focuses on and propose the semi-formal method of constructing individual educational students trajectories for the professional training of future software engineers in Ukrainian universities. The method for the student-centered learning conception support presented by the authors was developed using graph theory. The set of educational paths that the student is invited to choose corresponds with the requirements of IT enterprises and provides the student with the opportunity to choose his or her educational path for the level of education legally enshrined in Ukraine. The article presents the result of educational trajectories constructing for the process of the professional training of future software engineers from a Junior Bachelor level to Doctor of Philosophy. Ukrainian universities can use the proposed educational trajectories of the professional training of future software engineers for curricula and syllabus preparation. The practical development of this work is an interactive online assistant that will recommend to the student the choice of the educational path from the already achieved level to the target level, which the student will choose.

Keywords: competence, curricula, student-centered learning, sustainability, software engineering, standards.

1 Introduction

In our volatile, intimately connected, and dangerously vulnerable world educational system must be anticipative because the future is far from a linear extension of the past [1]. The article [2] introduces an emerging sustainable learning and education philosophy of learning as a fundamental principle of the sustainable curricula building with appropriate methods of learning and teaching.

M. Knowles is the creator of the concept of self-directed learning, proved that all individuals by nature could be self-directed, even if they need help to get started [3].

The paradigm shift from teaching to learning has contributed to the transfer of power from teacher to student. The teacher should be a "facilitator of the learning process." In the 1980s, this student-centered learning (SCL) concept was transformed by C. Rogers into the theory of education [4].

The works of John Dewey [5] and Carl Roger [6] about personality formation have significantly influenced to SCL. They criticized the obsolete model of education, in which the teacher assigned the central role, and the student is a passive participant in the educational process. Scientists focused that students are as equal partners as a part of the academic community, and coproduces of knowledge is essential for giving student ownership and responsibility for their learning.

SCL has repeatedly demonstrated superiority over the traditional teacher-oriented approach. The research of the Erasmus+ [7] shows that SCL is introduced in different professional fields, different geographical areas, and that it can be practiced even in big classes. The impact of SCL on learning outcomes is confirmed by [8]. The authors confirm advances in designing an SCL process using cutting-edge methods, tools, and artificial intelligence. The authors [9] proved that SCL means that learning outcomes must meet the needs and expectations of students and society while providing employment and personal development.

Nowadays, a joint project of the Union of Higher School Employees (EI) and the European Union of students (ESU) [10] has proposed the main principles and recommendations that underpin in SCL. The project aims to re-evaluate the progress of implementation of SCL, highlight best practices, and establish peer assessment procedures for the implementation of the conception in European higher education institutions. Inspired by the growing interest and progress in implementing SCL, the European Students Union (ESU) developed a new project called Peer Assessment of Student-Centred Learning, funded with the support of the European Commission [11]. Through the research done in this project, the conception of learning outcomes forms the conceptual core for SCL higher education system.

Competencies are one of the main categories of SCL. Fundamental to our study is the competency-based approach (CBA), which based on the concepts of competency and competence. Competence as a complex object includes substantial components, based on knowledge, skills, experience [12]. European e-Competence Framework (e-CF) [13] provides a reference of 40 competencies applied to the field of Information and Communication Technologies (ICT) using a common language of competences, skills, and capability levels.

The work [14] presents an approach to the curricula building as a combination of academic competences and professional competences from the e-CF. The authors developed a competences description for 16 disciplines for proposed a Joint Master Program in Software Engineering. Zaitseva L. [15] pays our attention to the definition of 9 competencies in the Bachelor and 15 competencies in the Master study formed during the acquisition of the study course "Software Engineering" at Riga Technical University on the e-CF base. Evaluation of practical assignments allows assessing the achievement level of the required competencies. Fitsilis P. et al. [16] propose to use a conceptual grid to address different skill needs in different workforce segments in different sectors, depending on different subsets of the technologies. The proposed competence framework consists of six dimensions (proficiency, technology, industry sector, product lifecycles, job profile, and transversal skills) that need to be combined in order to produce the training needs for each specific case or to calculate the skilled coverage when the readiness of enterprises for introducing Industry 4.0. Liagkou V. et al. proposed an innovative online methodology to evaluate student's competencies and achieved learning outcomes [17]. The core of the methodology is an Engineering Competence Evaluation Internet Portal. The portal provides to the students and professors an innovative research-based online tool, which facilitates the exchange of information on learning outcomes, referring to a graduate's knowledge, skills, and e-CF competence upon completion of their Master of Science in Software Engineering Program.

The European ICT Professional Role Profiles make a crucial contribution to increasing transparency and convergence of the European ICT Skills landscape [18]. Incorporating the competences of the European e-Competence Framework as the main component of profile descriptions, the 30 ICT Professional Role Profiles provide a generic set of typical roles performed by ICT Professionals in any organization, covering the full ICT business process. Complementary to the e-CF, the European ICT Professional Role Profiles contribute to a shared European reference language for developing, planning, and managing ICT Professional needs in a long-term perspective and to maturing the ICT Profession overall.

However, educational standards and training programs at various levels are often not harmonized, based on incompatible competency systems, which leads to problems. It does not allow ensuring the continuity of educational programs at various levels, the possibility of reducing the learning duration and supporting individual educational trajectories of students. The presence of an extensive list of competencies inevitably complicates the development of appropriate assessment tools and methods to test the level of their formation among student

Vocational education and training reform and the Law on Professional Pre-higher Education establish a Junior Bachelor's Degree, which aims at training skilled workers in Ukraine. Junior Bachelor's Degree corresponds to the sixth level of the National Qualifications Framework (Initial level – the short cycle of higher education) [19]. The National Qualifications Framework agreed upon the Qualifications Framework of the European Higher Education Area and the European Qualifications Framework for Lifelong Learning. The Junior Bachelor's Degree (named associate's or foundation degrees in different countries) is an academic program taken at the undergraduate level. The Junior Bachelor's in Software Engineering must acquire the necessary technical and academic knowledge and transferable skills they need to go on to employment or further study.

In the formation and implementation of SCL university professional standard for training software engineers at Ukrainian universities, one must proceed from the fact that each level of higher education and the corresponding professional standard (educational-professional and educational-scientific) is an integral part of the process, which provides for consistency, continuity, and regularity. SCL in the professional standard cannot be fully implemented without individual educational trajectories for training software engineers. Actually, in the education system, individual educational trajectories of training should begin at the level of the Junior bachelor's programs. University educational programs at the Bachelor's level should continue the Junior bachelor's programs, be associated with them, but do not repeat them. At each next level of higher education, the role and practice of applying an individual learning trajectories students should increase.

In this paper, we adapt our previously developed general approach [20] to the changes that have taken place over the past 1-2 years in educational and professional spheres. We solve the task of constructing educational trajectories for training software engineers at Ukrainian universities. These trajectories must fit the qualifications framework and must support the SCL conception.

The paper is structured as follows. Section II presents the theoretical basis for the reasoned choice of the individual educational student trajectories according to the European e-competence framework. Section III presents the results of the proposed method according to the European e-competence framework for Software Engineering specialty.

2 Method of Constructing Individual Educational Students Trajectories According to European E-Competence Framework

2.1 Input Assumptions and Limitations.

The target competencies for training specialists in Software Engineering (SE) must provide training for profiles "Developer" and "Test Specialist" because these profiles fully meet the qualifications of Degree "Specialist in the development and testing of software" by the National classification of professions Ukraine. Since the profile "Digital media specialist" is very close to the two profiles above for the necessary competencies [21], we assume that these three profiles will form the core and start point for the specialty "Software engineering."

The target competencies that Developers, Digital Media Specialists, or Test Specialists must possess are limited to the third level of E-CF, which corresponds to the bachelor's level both in the European Qualifications Framework and the Law on Higher Education of Ukraine. Therefore, to continue training to the level of a master or a doctor of philosophy is pointless.

The choice of further educational paths should be justified, proceeding, first of all, from the desired target profile. The criterion for the validity of the choice of an educational trajectory is the minimum difficulty in acquiring new necessary competencies and increasing the level of competencies already achieved at the previous level of education. In our opinion, this complexity can be quantified by summing up the number of new competencies required, taking into account their level and the number of the new deliverables for which the future graduate will be responsible.

Only training paths for which the difficulty measured by the above method is small enough can be recommended for training. Obtained with this method of measuring assessments of the complexity of training will be more likely to be qualitative, relative, and will never become accurate, quantitative. In our opinion, the result we obtained is entirely consistent with the stated goal of the work: to formulate recommendations for the student on the choice of the educational path in terms of "it is easy," "it is harder," and "it is almost unattainable."

We are in no way trying to express numerical recommendations of the form "Training to achieve at the level of the philosophy doctor of competencies required to work on profile X will be many times Y more difficult than preparing you for work on profile Z." In our opinion, this is justified, the reasoned approach to the maximum extent corresponds to the concept of Student-Centered learning, as it allows each student to plan personally and, if necessary, adjust their path in their education.

2.2 The Formal Statement of the Problem.

The initial data are the interrelations matrixes of the "ICT profiles – E-competences" and "ICT profiles – Deliverables." In the mathematical representation, these tables define two bipartite graphs (mappings):

$$\Phi_1: P \to C, \Phi_2: P \to D, \tag{1}$$

where $\mathrm{P} = \{\mathrm{P}_1, \mathrm{P}_2, \ldots \mathrm{P}_{30}\}$ – the profiles set,

$\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}_2, \ldots \mathrm{C}_{40}\}$ – the competences set,

$\mathrm{D} = \{\mathrm{D}_1, \mathrm{D}_2, \ldots \mathrm{D}_{76}\}$ – the deliverables set.

The mapping of the profiles set to the competences set we present as $G = \langle P, C, E \rangle$, where E denoting the weighted edges set of the bipartite graph. The weight of each edge $e_{p,c}$ we define as the level of the competence with number c, which is needed to profile with number p. Similarly, the mapping of the profiles set to the deliverables set is $H = \langle P, D, K \rangle$, where K denotes the edges of the bipartite graph and each edge$K_{p,d\in}\{0,1\}$.

The proximity function between any two profiles s and t we define as distance in an undirected graph:

$$\Pr(s,t) = \alpha \sum_{\forall c, c \in C} \min\,(e_{s,c}, e_{t,c} + (1-\alpha) \sum_{\forall d, d \in D} k_{s,d}\, k_{t,d}, \tag{2}$$

where α - coefficient that allows controlling the importance of proximity metric according to common competencies relative proximity metric according to shared deliverables. The complexity function of the transition to a new profile t in the presence of competencies of the existing profile s we introduce as a distance in a directed graph:

$$Dif(s,t) = \alpha \sum_{\forall c, c \in c \cap (e_{t,c} > e_{s,c})} (e_{t,c} - e_{s,c}) + (1-\alpha) \sum_{\forall d, d \in D \cap (k_{t,d}=1)} (k_{t,d} - k_{s,d}) \tag{3}$$

The assumption that training is carried out in consistent progress through the educational levels leads to a search all possible pairs s and t within a stepwise education system with the suggestion that these pairs must satisfy the following condition:

$$\max_{\forall c, c \in C} (\, e_{t,c}) - \max_{\forall c, c \in C} (e_{s,c}) = 1 \tag{4}$$

We introduce the reachability matrix of the next target profile, starting with the existing one. In graph theory, reachability refers to the ability to get from one vertex to another within a graph. A vertex s can reach a vertex t, and t is reachable from s if there exists a sequence of adjacent vertices (i.e., a path) which starts with s and ends with t.

Taking into account constraint (4), the reachability matrix we present as:

$$Reach(s,t) = \begin{cases} \frac{1}{Dif(s,t)}, if\ condition\ (4) true \\ 0, otherwise \end{cases} \quad (5)$$

2.3 The Constructing Individual Educational Trajectories Set and Curricula Set

The method of constructing individual educational trajectories set and curricula set as a result is a sequence of following steps.

Step 1. Clustering profiles, as the process of ordering of profiles graph structure into relatively homogeneous groups, where the proximity function between any two profiles is calculated according to (2). We used the modularity metric proposed in the paper [22] for clustering. The metric value lies between 0.5 and 1, calculated by the formula:

$$Q=\frac{1}{2m}\sum_{s,t}\left(\Pr(s,t)-\frac{d_s d_t}{2m}\right)\delta(k_s,k_t), \quad (6)$$

where di is the degree of the i-th node, Ki is the cluster in which i-th node is located, m is the number of edges in the graph. $\delta(k_s,k_t)=1$, if $k_s=k_t$, otherwise 0.

Using the modularity metric (6) allows us to split the initial list of profiles into profile clusters, which are characterized by a strong connection of profiles within classes and weak for profiles located in different clusters (strong link inside and weak ones among groups). A change in the coefficient makes it possible to evaluate the stability of the clustering result when changing the evaluation criterion from proximity by competencies to proximity by deliverables.

Step 2. The reachability graph describes the ability to get from obtained profile to the next one by educational level with quantity estimation of this step complexity. The reachability graph is constructed by using (5) and is become a background for building individual educational trajectories set.

Step 3. The curricula set is constructed using individual educational trajectories set. The start point for each curricula building is the needing competencies for goal profile, knowledge, and skills examples, cataloged in the e-CF.

3 Individual Educational Student Trajectories According To The European E-Competence Framework

We performed a profile's clustering from e-CF with $\alpha=1$ from (2), taking into account only the quantity of common competencies required for each pair of profiles. As a result, we got seven closely related groups (fig. 1). Each group has its color, and the line thickness between each pair of profiles is proportional to the number of shared competencies.

The aimed group includes next roles:

- Developer.
- Digital Media Specialist.
- Test Specialist.
- DevOps Expert.
- System Administrator.
- Network Specialist.

Grouping of profiles by Deliverables/Outcomes proximity principle with $\alpha = 0$ from (2) gives the partially different result (fig. 2). A Deliverable, according to e-CF, is a predefined result of a task in a job context. Deliverables describe typical outcomes of a solved task in terms of accountable, responsible, or contribution. When clustering according to the proximity by Deliverables/Outcomes, the aimed group includes next roles:

- Developer.
- Digital Media Specialist.
- Test Specialist.
- DevOps Expert.
- Systems Analyst.
- Systems Architect.
- Digital Transformation Leader.

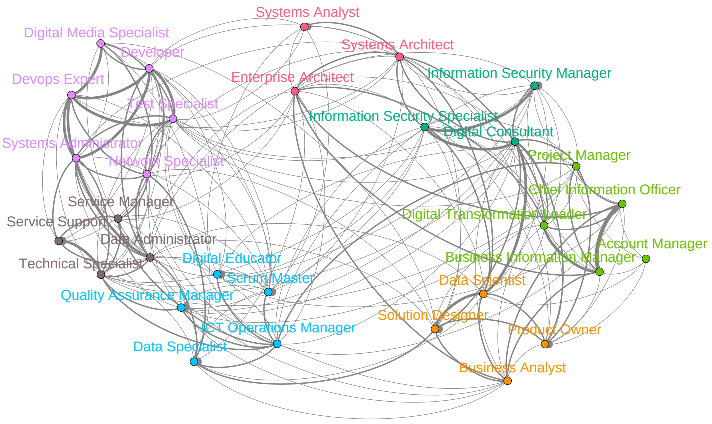

Fig. 1. Seven profiles groups (highlighted by different colors) which are closely related by the list of required competence

All recommended trajectories for achieving the desired profiles for software engineering are presented in fig. 3. The line thickness characterizes the number of competencies achieved at the previous level that will be claimed at the next one, that is, the thicker the connecting line, the easier it will be to achieve the target profile.

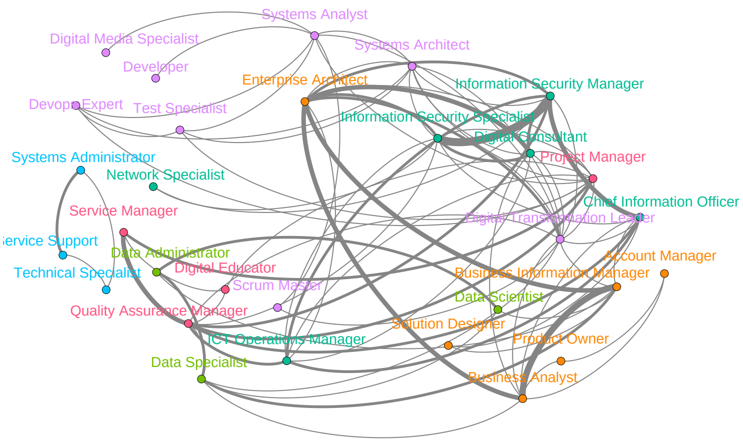

Fig. 2. Seven profiles groups (highlighted by different colors) which are closely related by the deliverables

Table 1 details recommended individual educational students trajectories, starting from the software engineering profiles (Developer, Digital Media Specialist, Test Specialist), which are available from junior bachelor level.

Table 2 contains reference information about deliverables and competencies, which must be formed or improved during the transition from systems architect role profile to digital transformation leader role profile.

4 Conclusions and Future Work

The key performance indicator for Student-Centred Learning is student satisfaction with the incorporation of sustainable curricula. The e-CF has proved to be very useful in the process of curricula design because one of the critical challenges of effective curricula design is managing how different stakeholders communicate and cooperate in designing curricula that meet both educational and employer objectives.

The article proposes a partial formalization of the method of constructing educational trajectories for training software engineers at Ukrainian universities. Improving student knowledge, skills, and competence without a clear plan and goal as to how that future Bachelor, Master, or Ph.D. will correspond to the organizational needs of an employer generally has minimal effect. Students should understand the general learning objectives and not learn in fragments without any apparent purpose. The effective use of individual SCL should be coordinated holistically to provide adequate support to the needs of the employer organization.

Table 1. Recommended individual educational students trajectories according to Software engineering profiles (only for profiles inherited from Developer, Digital media specialist or Test specialist with Junior Bachelor level)

<table>
<tr><th>Profile</th><th>ICT Profile Families</th><th>Proximity by competencies</th><th>Proximity by deliverables</th><th>Educational level</th></tr>
<tr><td>Developer</td><td rowspan="3">Development</td><td>+</td><td>+</td><td rowspan="5">Bachelor</td></tr>
<tr><td>Digital Media Specialist</td><td>+</td><td>+</td></tr>
<tr><td>Test Specialist</td><td>+</td><td>+</td></tr>
<tr><td>System Administrator</td><td rowspan="2">Service & Operation</td><td></td><td>+</td></tr>
<tr><td>Network Specialist</td><td></td><td>+</td></tr>
<tr><td>Systems Analyst</td><td rowspan="2">Design</td><td>+</td><td></td><td rowspan="3">Master</td></tr>
<tr><td>Systems Architect</td><td>+</td><td></td></tr>
<tr><td>DevOps Expert</td><td rowspan="2">Process Improvement</td><td>+</td><td>+</td></tr>
<tr><td>Digital Transformation Leader</td><td></td><td>+</td><td>Doctor of Philosophy</td></tr>
</table>

Table 2. Summary table describing the needed competencies and deliverables of the transition from systems architect role profile to digital transformation leader role profile

<table>
<tr><th></th><th>Systems Architect</th><th>Digital Transformation Leader</th></tr>
<tr><td rowspan="6">Deliverables</td><td>Solution Specification</td><td></td></tr>
<tr><td>Integrated Solution</td><td></td></tr>
<tr><td>Development Process</td><td></td></tr>
<tr><td colspan="2">Solution and Critical Business Process Integration Proposal</td></tr>
<tr><td></td><td>Digital Transformation Roadmap</td></tr>
<tr><td></td><td>Digital Transformation Strategy</td></tr>
<tr><td rowspan="7">Competencies</td><td>A.7. Technology Trend Monitoring</td><td></td></tr>
<tr><td>B.2. Component Integration</td><td></td></tr>
<tr><td>B.6. Systems Engineering</td><td></td></tr>
<tr><td colspan="2">A.5. Architecture Design</td></tr>
<tr><td colspan="2">A.9. Innovating</td></tr>
<tr><td></td><td>A.3. Business Plan Development</td></tr>
<tr><td></td><td>E.7. Business Change Management</td></tr>
</table>

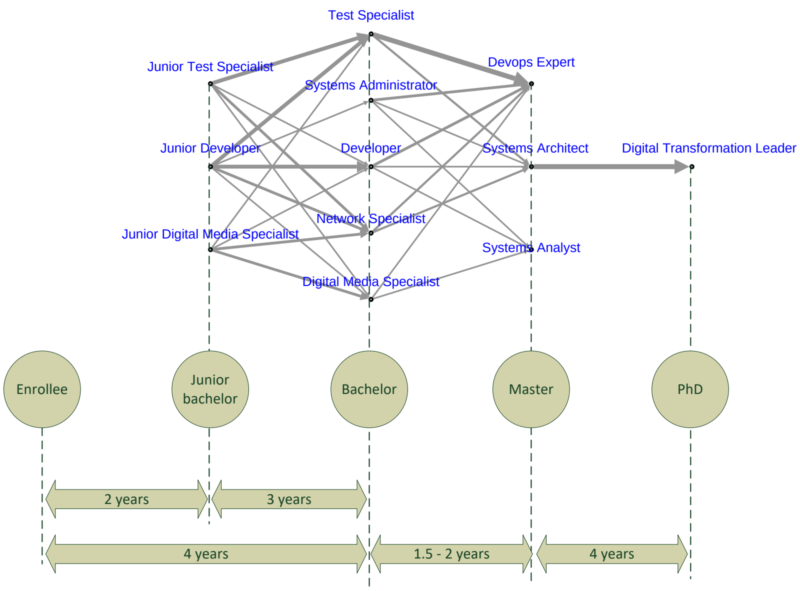

Fig. 3. Trajectories (possible paths) for achieving the desired profiles for software engineering with accounting common competencies and deliverables (the line thickness characterizes the number of competencies achieved at the previous level that will be claimed at the next, that is, the thicker the connecting line, the easier it will be to achieve the target profile)

The set of educational trajectories (individual paths) that a student is offered to choose must meet the requirements of IT enterprises, fit the qualifications framework, and must support the SCL concept, that is, enable the student to choose his or her educational path for one from the educational levels legally enshrined in Ukraine.

The following features of software engineering and higher education in Ukraine are taken into account.

1. The target competencies that Developers, Digital Media Specialists, or Test Specialists must possess are limited to the third level of E-CF, which corresponds to the bachelor's level both in the European Qualifications Framework and the Law on Higher Education of Ukraine. Therefore, to continue training to the level of a master or a doctor of philosophy is pointless. The choice of further educational paths should be justified, first of all, from the competencies desired target profile.
2. Vocational education and training reform and the Law on Professional Pre-higher Education establish a Junior Bachelor's Degree, which aims at training skilled workers. Junior Bachelor's Degree corresponds to the sixth level of the National Qualifications Framework (Initial level – the short cycle of higher education) [23]. The National Qualifications Framework agreed upon the Qualifications Framework of the European Higher Education Area and the European Qualifications Framework for Lifelong Learning. The Junior Bachelor's Degree (named associate's or foundation degrees in different countries) is an academic program taken at the undergraduate level. The Junior Bachelor's in Software Engineering must acquire the

necessary technical and academic knowledge and transferable skills they need to go on to employment or further study.

The proposed method for the set constructing of the software engineer's educational trajectories in Ukrainian universities uses graph theory. The nodes of the weighted graph are the roles proposed by the European e-Competence Framework, and the weights of the edges characterize the number of competencies achieved at the previous level of education that will be necessary to achieve the next level. Each trajectory is a path starting from the enrollee-node and ending on the master level or Ph.D. level node. An additional metric of the distance between nodes calculates the total number of deliverables shared by these roles.

Each person selects their trajectory personally; therefore, the mathematically solved problem belongs to the class of tasks in which it is required to list all implemented trajectories for future choice. After that, the student must choose his learning path, and the university must build the educational programs necessary for the implementation of the required trajectories.

The article presents the result of the synthesis of educational trajectories for software engineering from a junior bachelor level. It is shown that the subsequent levels of education should be oriented toward acquiring the competencies necessary for the DevOps Expert Role, Systems Architect Role, or Systems Analyst Role at the master's level. If a student plans to achieve the level of Doctor of Philosophy, starting from the three listed above Junior Bachelor's Degrees, he or she should choose only the Systems Architect Role at the Master degree, since only this role is firmly connected by common competencies and deliverables with the Digital Transformation Leader Role.

Further development of the work is planned in the practical implementation of an interactive online assistant, which will formulate recommendations for the student on the choice of disciplines, forming a set of trajectories from the already achieved level to the target one that the student will choose.

5 Acknowledgment

This research is supported by the project STARC (Methodology of SusTAinable Development and InfoRmation Technologies of Green Computing and Communication) funded by the Ministry of Education and Science of Ukraine.

References

1. Sterling, S.: Sustainable education - towards a deep learning response to unsustainability. Policy & Practice: A Development Education Review, 6, 63-68 (2008).
2. Hays, J., Reinders, H.: Sustainable learning and education: A curriculum for the future. International Review of Education, 66, 29–52 (2020).
3. Knowles, M.: Self-Directed Learning: A Guide For Learners and Teachers. New York: Association Press (1975).
4. Rogers, C.: Freedom to Learn for the 80's. New York: Charles E. Merrill Publishing Company, A Bell & Howell Company (1983).

5. Dewey, J.: Democracy and education: An introduction to the philosophy of education. New York: WLC Books (2009).
6. Rogers, Carl R., Lyon, Harold C., Tausch, R.: On Becoming an Effective Teacher – Person-centered Teaching, Psychology, Philosophy, and Dialogues with Carl R. Rogers and Harold Lyon. London: Routledg (2013).
7. Empowering Teachers for a Student-Centred Approach, https://ec.europa.eu/programmes/erasmus-plus/project-result-content/c55c65b9-7799-482e-b941-cfe94b3f3924/2%20SCL%20RESEARCH%20IN%20ENGLISH.pdf, last accessed 2020/05/21.
8. Proceedings of the 1th ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, https://doi.org/10.1145/3340435.3342716/, last accessed 2020/05/21.
9. 2019 ASEE Annual Conference & Exposition, https://peer.asee.org/31934, last accessed 2020/05/21.
10. Student-centered learning – Toolkit for students, staff and higher education institutions, http://www.aic.lv/bolona/2010/Reports/SCL_toolkit_ESU_EI.pdf, last accessed 2020/05/21.
11. Student-Centred Learning -Toolkit for Students, Staff and Higher Education Institutions, European Students' Union and Education International, https://www.esu-online.org/wp-content/uploads/2016/07/Overview-on-Student-Centred-Learning-in-Higher-Education-in-Europe.pdf, last accessed 2020/05/21.
12. Gervais, J.: The operational definition of competency-based education. Competency-based Education 1(2), 98–106 (2016).
13. European e-Competence Framework (e-CF), http://www.ecompetences.eu/e-cf-3-0-download/, last accessed 2020/05/21.
14. Jusas V., Ros J., Misnevs, B.: Software Engineering Competence Remote Evaluation Process Model. Baltic Journal of Modern Computing, 5(3), 317 (2017).
15. Zaitseva, L.: e-CF Competences in Software Engineering Course. Applied Computer Systems, 24(1), 32-37 (2019).
16. Fitsilis, P., Tsoutsa, P., Gerogiannis, V.: Industry 4.0: required personnel competences. Industry 4.0. 3(3), 130-133 (2018).
17. Liagkou, V., Stylios, C.: An E-competence evaluation portal for software engineering master course. International journal on information technologies& security, 9, 27-38 (2017).
18. European ICT Professional Role Profiles, http://www.ecompetences.eu/wp-ctent/uploads/2018/05/CWA_Part_1_EU_ICT_PROFESSIONAL_ROLE_PROFILES.pdf, last accessed 2020/05/21.
19. National Qualifications Framework, https://mon.gov.ua/eng/osvita/nacionalna-ramka-kvalifikacij, last accessed 2020/05/21.
20. Turkin, I., Narozhny, V., Vykhodets, Y.: The approach to drawing up a list of professional competences in the field of software engineering for harmonization of educational standards. In: The proceedings of international scientific conference "Information technologies and computer modelling, pp. 73-77. Ivano-Frankivsk (2017).
21. European ICT Professional Profiles, ftp: // ftp.cen.eu/CEN/Sectors/List/ICT/CWAs/CWA%2016458.pdf, last accessed 2020/05/21.
22. Girvan, M., Newman, M.: Community structure in social and biological networks. PNAS. 99(12), 7821–7826 (2002).
23. National Qualifications Framework, https://mon.gov.ua/eng/osvita/nacionalna-ramka-kvalifikacij, last accessed 2020/05/21.