https://doi.org/10.54989/msd-2023-0016

A KNOWLEDGE MANAGEMENT GUIDELINE: IDENTIFYING AND QUANTIFYING THE KNOWLEDGE LEVEL OF A TEAM THAT WORKS FOR A SOFTWARE ENGINEERING PROJECT

ȚÎȚU Aurel Mihail[1,2] and NIȚĂ Nicoleta Mădălina[3]
[1]Lucian Blaga University of Sibiu, 10 Victoriei Street, Sibiu, Romania, mihail.titu@ulbsibiu.ro
[2]The Academy of Romanian Scientists, 3 Ilfov Street, Bucharest, Romania
[3]National University of Science and Technology POLITEHNICA Bucharest, Faculty of Industrial Engineering and Robotics, 313 Splaiul Independenței, 6th District, Bucharest, Romania, madalina.nita12@yahoo.com

ABSTRACT: This scientific paper aims to explore the identification and quantification of the knowledge level within a team working on a software engineering project. The knowledge level of a team plays a crucial role in the success and efficiency of project outcomes. To achieve this objective, a comprehensive review of existing literature on knowledge management, team dynamics, and software engineering practices was conducted. This article reveals insights into the knowledge dynamics within software engineering teams. The research methodology consists of different quantitative and qualitative methods for data collection and analysis. The identified knowledge gaps and strengths can inform strategies for improving team performance, such as targeted training programs or knowledge sharing initiatives. Furthermore, the quantification of knowledge levels can serve as a benchmark for future projects, allowing for better resource allocation and team composition. Overall, this research is bringing benefits to the field of software engineering by providing a systematic approach to identify and quantify the knowledge level of teams, enabling organizations to optimize their team structures and enhance project outcomes.

KEY WORDS: Knowledge, Management, Project, Software, Engineering

1. INTRODUCTION

Knowledge management is critical in guaranteeing the success and efficiency of teams in software engineering projects. One crucial aspect of knowledge management is identifying and quantifying the knowledge level of a team. This process involves assessing the collective knowledge, skills, and expertise possessed by team members, enabling organizations to make informed decisions, allocate resources effectively, and foster continuous learning and improvement. Identifying and quantifying the knowledge level of a software engineering team is a multifaceted task that requires a structured approach. This guideline aims to provide insights and best practices for organizations seeking to undertake this endeavour. By following some guidelines, organizations can gain a comprehensive understanding of their team's knowledge assets, identify areas of strength and weakness, and develop strategies to enhance knowledge sharing and collaboration.

Assessing the knowledge level of a software engineering team is crucial for several reasons. It helps organizations identify gaps in expertise, determine training needs, and allocate resources appropriately. Additionally, it enables teams to leverage their collective knowledge effectively, avoid knowledge silos, and promote a culture of continuous learning and improvement.

To begin the process, organizations should define the knowledge domains and competencies relevant to their software engineering projects. These may include programming languages, software development methodologies, testing frameworks, field of work - specific knowledge, and soft skills such as communication and collaboration. By clearly defining these domains, organizations can ensure a comprehensive assessment of the team's knowledge.

There are various methods available to assess the knowledge level of a software engineering team. These can include self-assessment surveys, interviews, knowledge tests, peer evaluations, and performance evaluations. Organizations should select appropriate methods based on their specific needs, considering factors such as accuracy, scalability, and feasibility.

To quantify the knowledge level, organizations should develop assessment criteria aligned with the defined knowledge domains and competencies. These criteria can be based on proficiency levels, experience, certifications, completed projects, or any other relevant factors. It is essential to establish clear and measurable criteria to ensure consistency and objectivity in the assessment process.

Once the assessment methods and criteria are established, organizations can conduct knowledge assessments. This may involve administering surveys, conducting interviews, or evaluating performance based on predefined criteria. It is crucial to ensure confidentiality, encourage honest self-assessment, and provide opportunities for team members to showcase their knowledge and skills.

After collecting assessment data, organizations should analyse and interpret the results. This analysis can help identify areas of expertise, skill gaps, and patterns within the team. It is essential to consider both individual and collective knowledge levels to gain a holistic understanding of the team's capabilities.

Based on the assessment results, organizations can develop strategies to enhance the team's knowledge. These strategies may include training programs, mentoring initiatives, knowledge sharing platforms, cross-functional collaborations, or hiring additional expertise. The goal is to create an environment that fosters continuous learning, knowledge exchange, and skill development.

Knowledge management is an ongoing process, and it is crucial to monitor and evaluate the progress of knowledge enhancement strategies. Regular assessments, feedback mechanisms, and performance evaluations can help track improvements, identify new challenges, and refine knowledge management initiatives.

Identifying and quantifying the knowledge level of a software engineering team is a critical step in effective knowledge management. By following the outlined guidelines, organizations can gain valuable insights into their team's capabilities, make informed decisions, and foster a culture of continuous learning and improvement. Embracing knowledge management practices enables software engineering teams to maximize their potential, deliver high-quality projects, and stay ahead in an ever-evolving industry.

2. CONTRIBUTIONS TO THE DESIGN AND IMPLEMENTATION OF A MESH KNOWLEDGE NETWORK PROTOTYPE

The mesh type network diagram prototypes represented in Figures 1 and 2 are intended to facilitate and clarify the roles of the parties participating in software engineering projects, as well as in organizations in this sector in general.

The roles and competences of the parties participating in a project, but also in an organization that produces software systems and solutions in general, are highlighted in Figure 1. This sort of graphic was created to illustrate the significance of teamwork in order to share knowledge in order to achieve optimal results. This contribution arose from a perceived need in the software systems market, specifically a lack of knowledge about the other parties involved in a project. Most of the time, employees are focused on doing their duties successfully on their own, and when they find a difficulty, they spend hours attempting to fix it, losing sight of the comfort that comes from talking to a colleague who may have already handled a similar problem. Such a chart, customized according to the organization of the company to which it applies, with the ability to add concrete names and surnames, as well as the actual skills of each character appearing in it, could significantly reduce project downtime, employee burnout, and stress, while also increasing the level of communication and motivation among team members (Paganin et al., 2023).

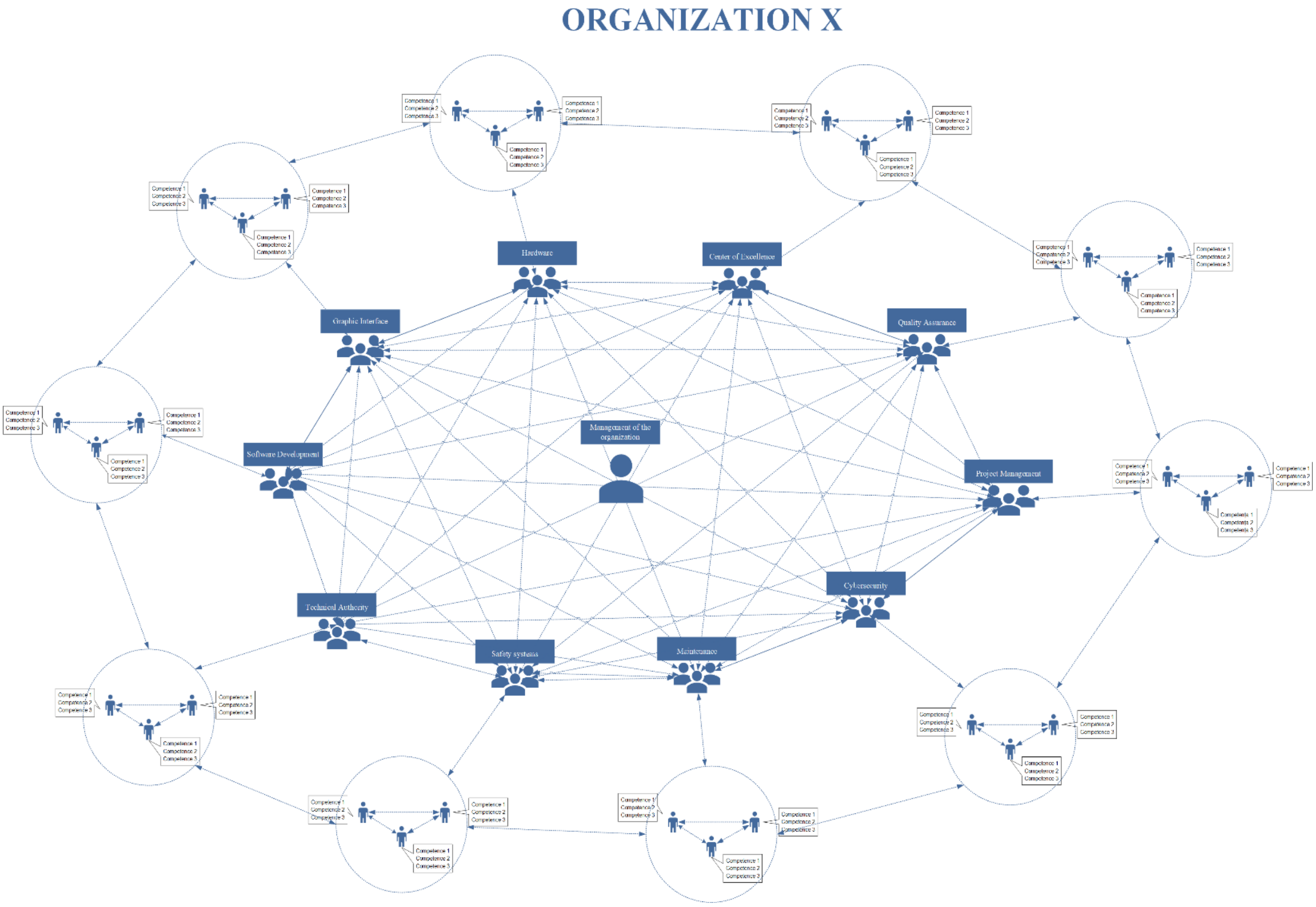

Figure 1. A knowledge mesh network prototype in a software engineering organization (Elzain, Wu, 2019)

A knowledge network in a software engineering organization refers to a system that facilitates the sharing, organization, and access to knowledge and information within the company. It serves as a centralized platform where employees can collaborate, exchange ideas, and access relevant resources to enhance their productivity and problem-solving capabilities (Epstein, Hundert, 2002). This network typically consists of various components such as a knowledge base, forums, wikis, document repositories, and communication channels. These tools enable software engineers to document their expertise, share best practices, and seek assistance from their peers when encountering challenges.

This diagram is useful for the organization's management team, particularly at the stage of contracting a new project, because it can be estimated whether the team possesses the set of knowledge and skills required to carry out that project based on the skills of the employees and their degree of load. Furthermore, as a method for future improvement, this chart can be enhanced with an IT solution so that it accurately reflects people's areas of expertise, level of training (if they are already involved in training programs to reinforce knowledge), and even load level (if they are already associated with a project and for how long).

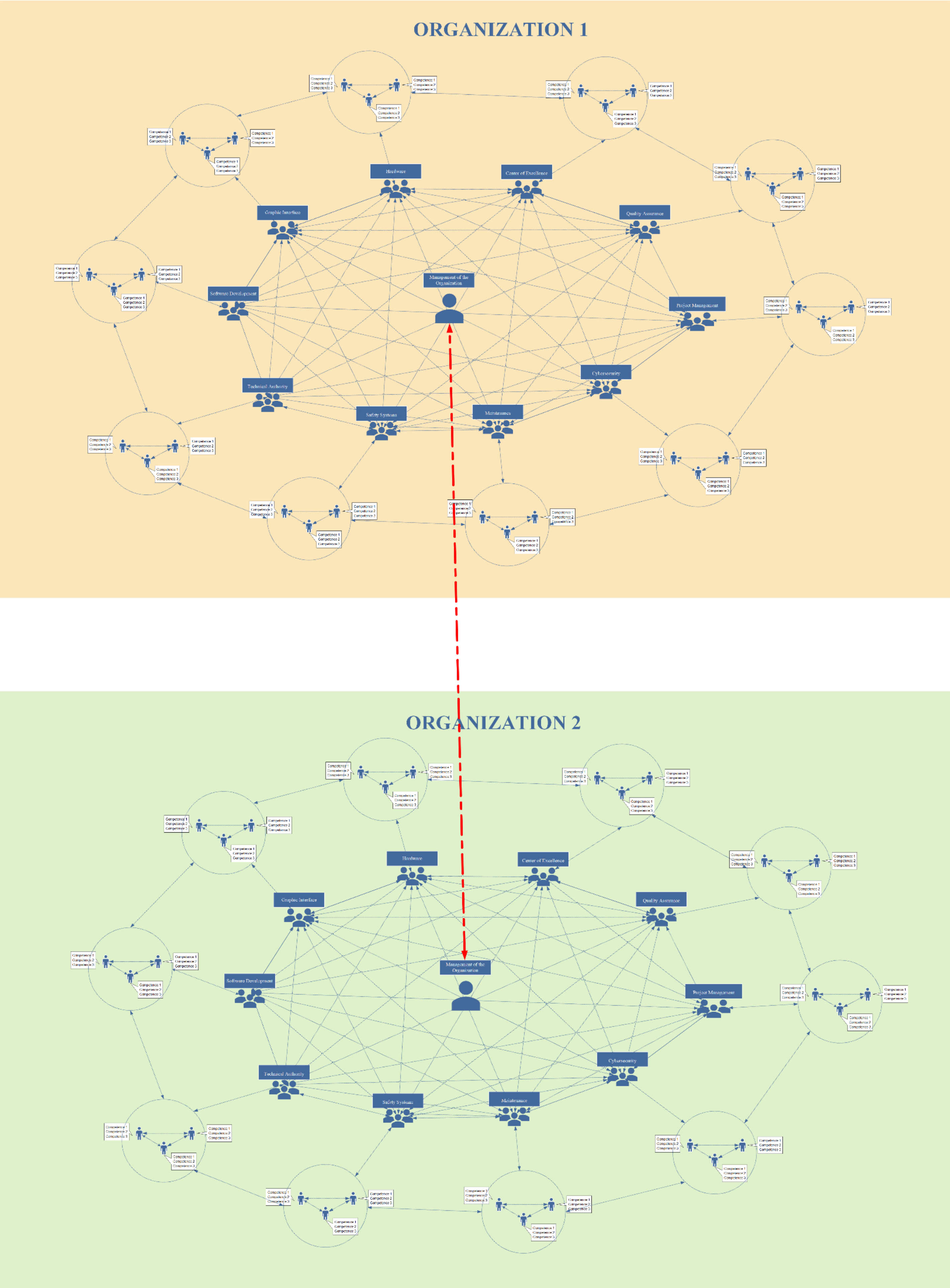

Figure 2. A knowledge mesh network prototype applied between two organizations

Figure 2 depicts the ease of information transfer even between two organisations that may not necessarily provide the same services or products. Departments can be altered and even upgraded to reflect the true condition of the organization's body of knowledge, much like a much clearer organizational chart.

3. CONTRIBUTIONS ON QUANTIFYING THE LEVEL OF KNOWLEDGE FOR STAKEHOLDERS IN SOFTWARE SYSTEMS ENGINEERING PROJECTS

Because the information, skills, and expertise of the parties involved in a project serve as the basis upon which the deliverables are formed, third parties are occasionally hired to cover knowledge gaps in specific areas of the project. Later, as the project progresses, this body of knowledge increases throughout the project's scope.

The parties involved in the project form a team, which, from some perspectives, can be considered to act like a small organization, implying a series of actions that must be carried out and considered in order for the knowledge that has been accumulated about that project to remain available, documented, and stored until the project is completed and even until new members join the team, leave it, or are absent for various reasons, which can translate as a threats.

Because of this risk, it is critical to know the level of expertise of all project participants at all times in order to establish measures to mitigate and enhance potential problem areas.

The level of knowledge of the persons participating in each project in the software systems engineering business determines the obstacles, as does their capacity to communicate these thoughts to the other parties through various modes of communication (Rauch et al., 2009). To objectively identify a project team member's level of competence and knowledge, something concrete and tangible must be converted into a form that is simple to grasp and interpret, and then evaluated using certain criteria. The existence of personal thinking influences of the evaluator, which are also a product of personal experiences, notions, information, data, and knowledge, facilitates the evaluation in the absence of a set of clear rules and principles on the quantification of knowledge from a project in all its forms and phases.

The most relevant knowledge must be safeguarded and supported, according to the flow diagram in Figure 3, which can even be transposed into the form of a questionnaire; hence, this illustration can be followed and evaluated both in the early stages of the project and at its completion. Because ineffective knowledge management entails risks, Figure 3 highlights the fact that developing employees' knowledge and skills does not constitute a risk-mitigation measure in projects until this knowledge is transferred to the rest of the team in order to achieve a skill balance (Powell et al, 2015).

Knowledge in a software engineering project, in my opinion, is generated by merging four domains:

- Elementary notions that we are born with and that we refine through education
    - Like the process of writing, reading, understanding a text, etc.
    - There are some people with innate genius who come with this extremely well-developed body of knowledge
- Technical knowledge acquired through education and professional training in the field
    - Bachelor's degree, Master's degree, doctorate in technical fields
    - Professional training courses for different IT systems, digital tools or the industry it addresses
- The experience
    - The more years of experience one has in a certain activity, the faster tasks are accomplished and issues are predicted and avoided.
- The emotional intelligence (EI)
    - Organizations are increasingly focusing on employees' emotional intelligence skills in order to strengthen their knowledge management strategies (as employees repeatedly share knowledge and experiences), to work more easily in teams (the collaboration principle), to improve communication, motivation, and creativity (via the brainstorming technique, etc.).

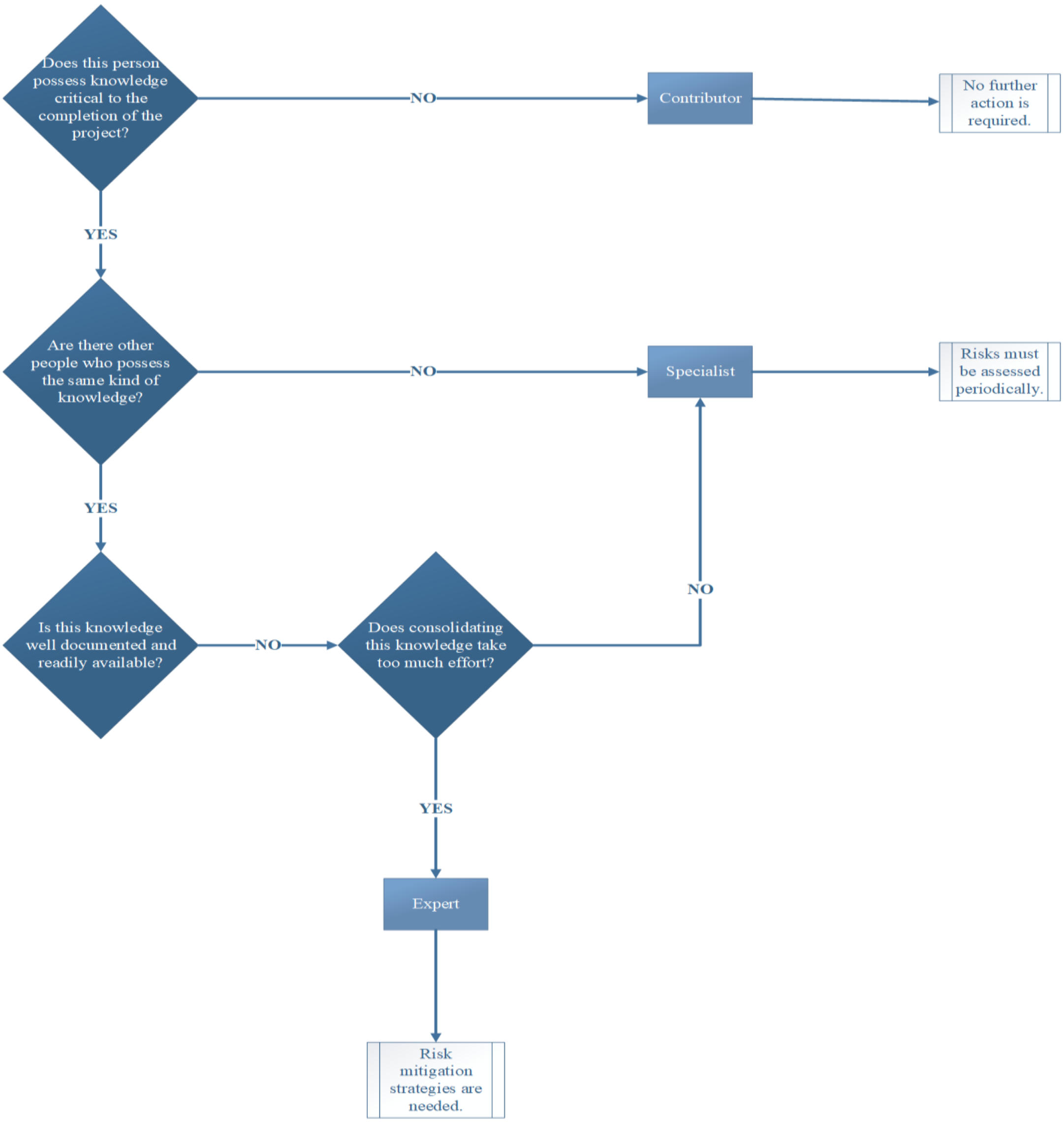
Does this person possess knowledge critical to the completion of the project?
NO
Contributor
No further action is required.
YES
Are there other people who possess the same kind of knowledge?
NO
Specialist
Risks must be assessed periodically.
YES
Is this knowledge well documented and readily available?
NO
Does consolidating this knowledge take too much effort?
NO
YES
Expert
Risk mitigation strategies are needed.

Figure 3. Flowchart of knowledge quantification for each project

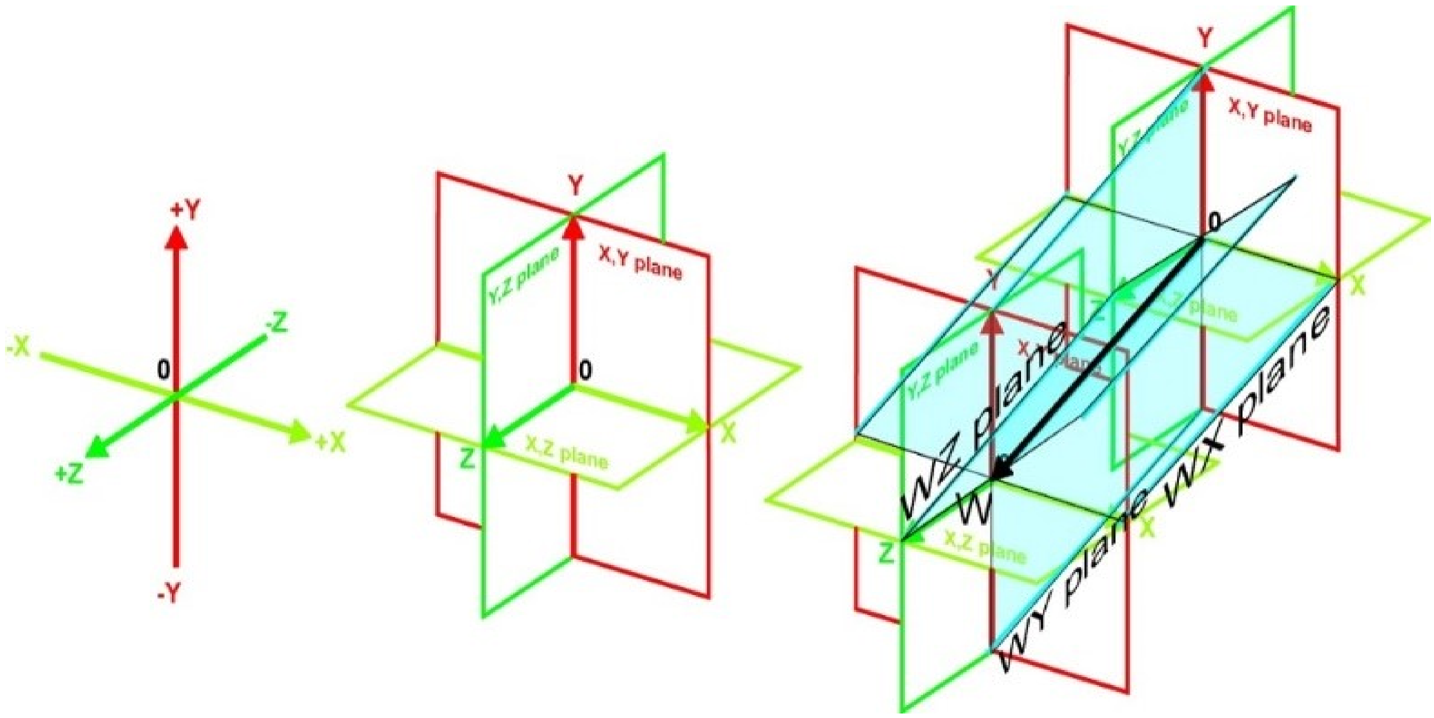

Figure 4. Example of drawing with 4 coordinate axes (FeltMagnet, 2022)

Thus, an interdependence relationship between these four major components exists, which can be represented in the form of a four-dimensional picture, as seen in Figure 4.

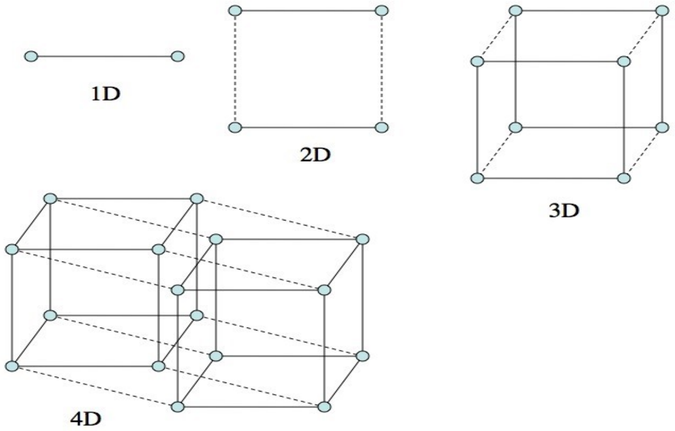

Figure 5. Example of 4D drawing (Radiya-Dixit, 2017)

To reinforce what was previously stated, we can see in the Figure 5 how each of the four components (presented in the Figure 5 under the names x, y, z, and w) complement each other, forming a plethora of planes, proving that any variation in one of the elements will automatically modify the others. Concerning the first two elements that comprise knowledge in a project, those elementary notions that we are born with and refine through schooling, combined with the technical ones acquired through training in this field, an indicator of quantification of this knowledge was established at the level of the European Union, namely the EQF (European Qualifications Framework), i.e. a European framework of qualifications. It is divided into eight tiers and it contains the knowledge, skills and responsibilities associated with each level.

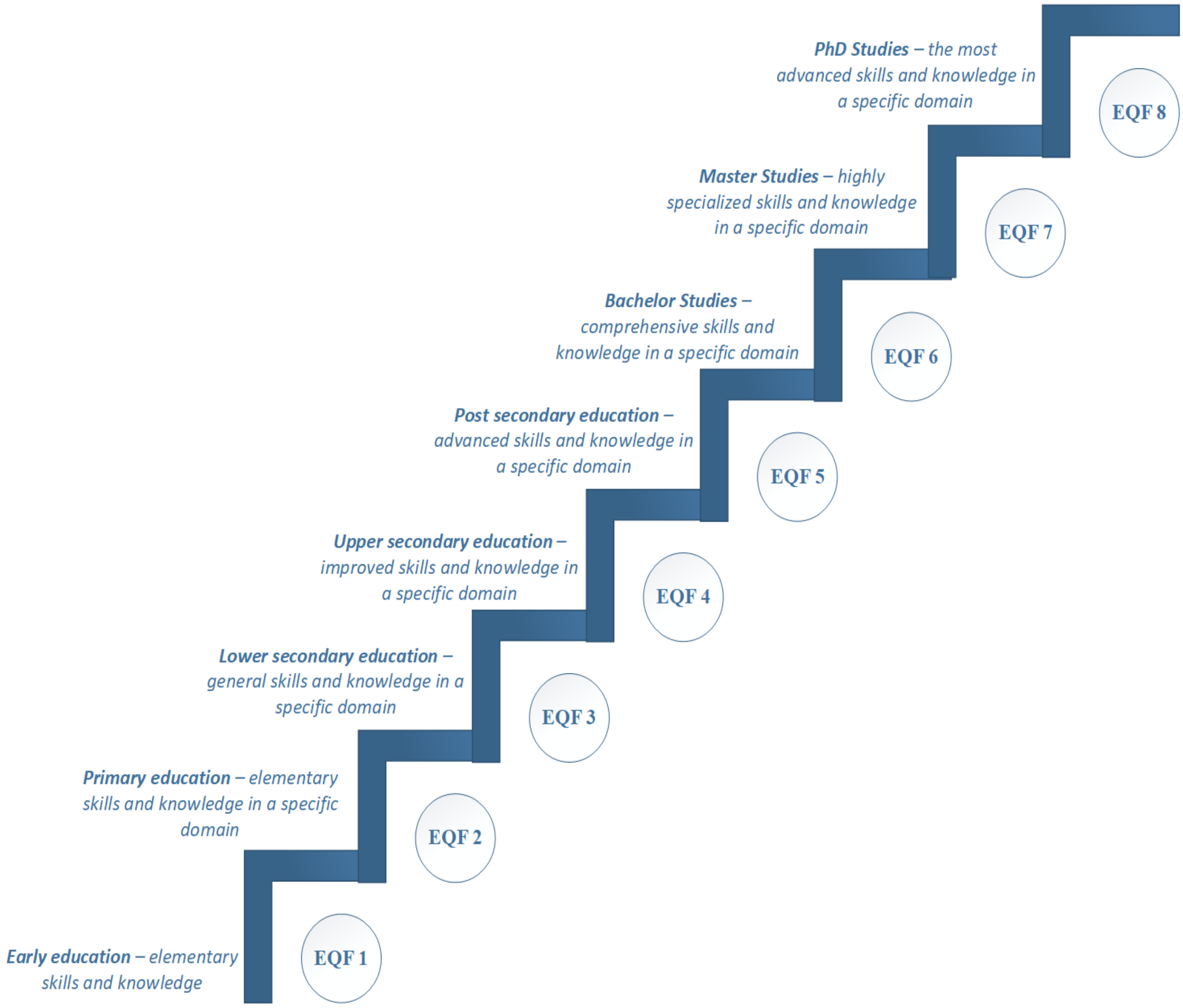
PhD Studies – the most advanced skills and knowledge in a specific domain
EQF 8
Master Studies – highly specialized skills and knowledge in a specific domain
EQF 7
Bachelor Studies – comprehensive skills and knowledge in a specific domain
EQF 6
Post secondary education – advanced skills and knowledge in a specific domain
EQF 5
Upper secondary education – improved skills and knowledge in a specific domain
EQF 4
Lower secondary education – general skills and knowledge in a specific domain
EQF 3
Primary education – elementary skills and knowledge in a specific domain
EQF 2
Early education – elementary skills and knowledge
EQF 1

Figure 6. The 8 levels of the European qualification framework (EQF) (Europass (n.d.))

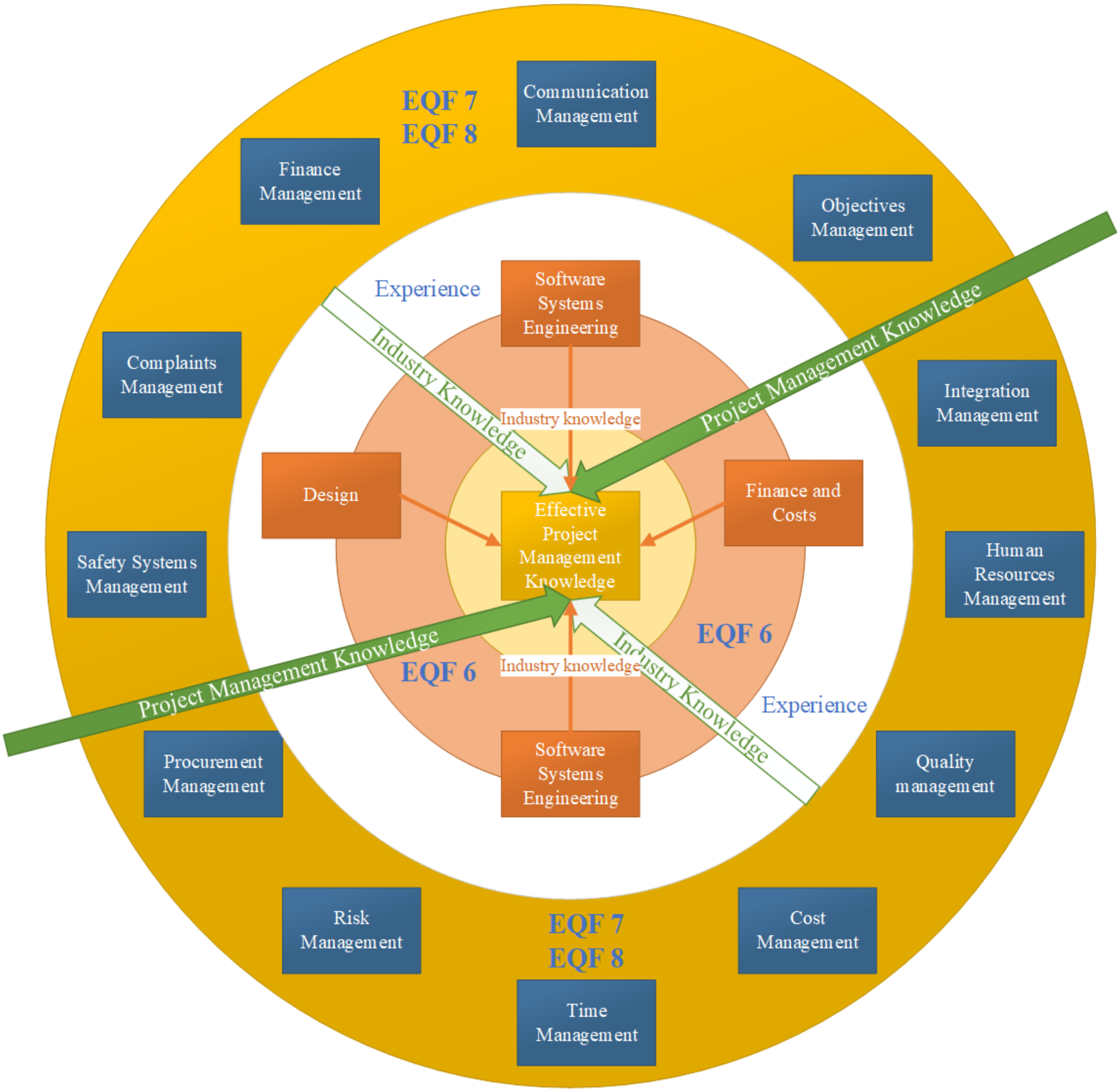

Figure 7. Schematic of an EQF application model in software systems engineering projects (Burger, Zulch, 2018)

Starting with the explanations in the Figure 6, Figure 7 aimed to emphasize the importance of knowledge qualifications for software systems engineering projects. A successful project in a sustainable organization requires the effective involvement of various departments other than those directly related to the execution of the products or services demanded by the client. On the first disc of the plan, some of the most significant participatory processes in project management are listed, such as financial management, quality management, human resource management, and so on (Deaconu et al., 2014). Because understanding the project from a general perspective requires a mix of knowledge acquired through schooling and deepened through academic research, which can only be quantified based on specific diplomas and accreditations, which refer to EQF 7 (master's studies) and EQF 8 (doctoral studies), but also project management knowledge, the competence of these departments increases with the hierarchical position of the employees (Bleiklie et al., 2017).

Moving towards the centre of the circle, you can see the shift to the more technical part of the project, which has been heavily influenced by experience and the lessons gained from it, and which has been transferred as a foundation in the creation of the project's objectives. At this point, the importance of project management can be emphasized, as it serves as an interface and point of contact between the organization's management and the entire project team.

The knowledge accumulated in the industry to which we refer represents a significant improvement in the quality of the project's deliverables, because it assists the project in developing solutions that can be concretely applied to the client's needs, eliminating the dead time for learning from mistakes, which translates into redoing the work at the expense of some resources (Florian-Gaviria et al., 2013). The minimum EQF 6 knowledge quotient, which refers to undergraduate studies in the topic, determines the pinnacle of knowledge in this phase, when the most technical knowledge is necessary. When discussing the industry, expertise is required not only in the software systems engineering domain, in order to design the product, but also knowledge of the industry in which the product will be implemented, which can be obtained through customer-provided training programs.

The minimum knowledge quotient necessary at that moment is presented in each of the phases specified in Figure 7, but it is encouraged to exceed it and train the personnel as much as possible in terms of education, know-how, and experience, to be able to supply new project progress definitions.

4. CONCLUSIONS

Identifying and quantifying the knowledge level of a team working on a software engineering project is crucial for effective project management and successful outcomes. This process involves assessing the collective knowledge, skills, and expertise of team members to determine their proficiency in relevant areas. By understanding the knowledge level of the team, project managers can make informed decisions regarding resource allocation, training needs, and task assignments.

The knowledge level of a software engineering team can be evaluated through various methods and techniques. One common approach is conducting a skills assessment, which involves assessing individual team members' technical competencies, programming languages, software development methodologies, and domain knowledge. This assessment can be done through interviews, self-assessment questionnaires, or technical tests. By analysing the results, project managers can identify strengths and weaknesses that require improvement within the team.

Another method for identifying knowledge levels is through knowledge mapping. This involves mapping out the knowledge domains and areas of expertise required for the software engineering project. By identifying the key knowledge areas, project managers can assess the team's proficiency in each domain and identify any gaps that need to be addressed. This can be done through surveys, interviews, or by analysing past project experiences and performance. Quantifying the knowledge level of a team can be challenging, as knowledge is often tacit and difficult to measure objectively. However, there are approaches that can help in this process. One such approach is using a knowledge maturity model, which provides a framework for assessing the team's knowledge level based on defined criteria and benchmarks. This model can include factors such as experience, certifications, training, and the ability to use knowledge effectively in practical situations. By assigning scores or levels to each criterion, project managers can quantify the team's knowledge level and track progress over time.

Additionally, project managers can leverage tools and technologies to support the identification and quantification of knowledge levels. Knowledge management systems can be used to capture and organize explicit knowledge, such as documentation, best practices, and lessons learned. These systems can also facilitate knowledge sharing and collaboration among team members, enabling the transfer of tacit knowledge. Data analytics and machine learning techniques can be applied to analyse project data, identify patterns, and extract insights about the team's knowledge level. Once the knowledge level of the team is identified and quantified, project managers can take appropriate actions to enhance the team's capabilities. This may involve providing targeted training programs, mentoring or coaching sessions, or assigning team members to projects that align with their expertise. By addressing knowledge gaps and fostering continuous learning, project managers can improve the overall performance and productivity of the software engineering team.

In conclusion, identifying and quantifying the knowledge level of a team working on a software engineering project is essential for effective project management. Through skills assessments, knowledge mapping, and the use of knowledge maturity models, project managers can gain insights into the team's proficiency and identify areas for improvement. Leveraging tools and technologies can further support this process. By taking appropriate actions to enhance the team's knowledge and skills, project managers can maximize the team's potential and increase the chances of project success.

REFERENCES

1. Bleiklie, I.; Frolich, N.; Sweetman, R.; Henkel, M. (2017) Academic Institutions, Ambiguity and Learning Outcomes as Management Tools. *European Journal of Education*, 52(1), 68-79. https://doi.org/10.1111/ejed.12200.
2. Burger, M.; Zulch, B. (2018) A construction project management knowledge model: The type and level of knowledge required. *Acta structilia* (Online), 25(1). http://dx.doi.org/10.18820/24150487/as25i1.4.
3. Deaconu, A.; Osoian, C.; Zaharie, M.; Achim, S. A. (2014) Competencies in Higher Education System: an Empirical Analysis of Employers' Perceptions. *Amfiteatru Economic*, 16(37), 857.
4. Elzain, H.; Wu, Y. (2019) Software Defined Wireless Mesh Network Flat Distribution Control Plane. *Future Internet,* 11(8), 166. https://doi.org/10.3390/fi11080166.
5. Epstein, R.; Hundert, E. (2002) Defining and assessing professional competence. *Jama-Journal of the American Medical Association*, 287(2), 226-35, doi: 10.1001/jama.287.2.226.
6. Europass (n.d.) Description of the eight EQF levels. Retrieved from https://europa.eu/europass/en/description-eight-eqf-levels (accessed Apr. 23, 2023).
7. FeltMagnet (Sep. 28, 2022) *How to Draw Four Dimensional Figures*. Retrieved from https://feltmagnet.com/drawing/How-to-Draw-Four-Dimensional-Figures (accessed Apr. 23, 2023).
8. Florian-Gaviria, B.; Glahn, C.; Fabregat Gesa, R. (2013) A Software Suite for Efficient Use of the European Qualifications Framework in Online and Blended Courses. *IEEE Transactions on Learning Technologies*, 6(3), 283-296. doi: 10.1109/TLT.2013.18.
9. Paganin, G.; De Angelis, M.; Pische, E.; Violante, F. S.; Guglielmi, D.; Pietrantoni, L. (2023) The Impact of Mental Health Leadership on Teamwork in Healthcare Organizations: A Serial Mediation Study. *Sustainability*, 15(9). https://doi.org/10.3390/su15097337.
10. Powell, B.J.; Waltz, T.J.; Chinman, M.J. et al. (2015) A refined compilation of implementation strategies: results from the Expert Recommendations for Implementing Change (ERIC) project. *Implementation Sci*, 10, 21. https://doi.org/10.1186/s13012-015-0209-1.
11. Radiya-Dixit, A. (2017) Visualizing the Fourth Dimension. *Duke University - Research Blog*, Apr. 26. Retrieved from https://researchblog.duke.edu/2017/04/26/visualizing-the-fourth-dimension/ (accessed Apr. 23, 2023).
12. Rauch, A.; Wiklund, J.; Lumpkin, G. T.; Frese, M. (2009) Entrepreneurial Orientation and Business Performance: An Assessment of Past Research and Suggestions for the Future. *Entrepreneurship Theory and Practice*, 33(3), doi: 10.1111/j.1540-6520.2009.00308.x.