An Empirical Analysis of Critical Success Factors for CD-ERP Model

Tareq Salahi Almigheerbi[1]*, David Ramsey[2], Anna Lamek[2]
[1] University of Tripoli, Tripoli, Libya.
[2] Wrocław University of Science and Technology, Wrocław, Poland.

* Corresponding author. Email: t.almigheerbi@uot.edu.ly
Manuscript submitted October 16, 219; accepted January 30, 2020.
doi: 10.17706/jcp.15.2.37-47

Abstract: Collaboratively-Developed ERP (CD-ERP) is an intermediate model between two well-known models, namely: closed- and open-source systems. This model is based on the community-source paradigm and ERP packages. It has been proposed for Libyan higher education (LHE) to be governed by a consortium of LHE institutions. To investigate the applicability of this model in LHE, this paper presents an analysis of critical success factor (CSFs) based on both a review of the relevant literature and lessons learnt from similar projects. Using the literature review, the title of the model was proposed on the basis of its fundamental elements. Cases of other projects which follow a similar approach show that community-source has been adopted by many universities and government bodies worldwide. Also, many of the issues observed in such projects are similar to those that appear in the literature, especially in regards to distributed and collaboratively working environments.

Keywords: Collaboratively-developed ERP, critical success factors, cloud computing, multi-tenancy, community-source, CD-ERP.

1. Introduction

Based on recent studies, Libyan higher education (LHE) has fallen to the bottom third of the countries surveyed [1]. One of the keys to boosting a Higher Education (HE) system is by enhancing the Information Systems (ISs) used within its institutes. Libyan universities have struggled to invest in the deployment of ISs. At the same time, obtaining reliable ISs that suit HE's unique needs is an issue at international level, not only in the Libyan environment. As in HE organizations worldwide, Libyan universities have been buying commercial ISs or developing their own in-house applications. The former may be hard to customize based on HE's specific needs, while the latter are often too inflexible and many organizations find them impractical [2]. Collaborative development (CD) - or what is also known as the "Community-Source" approach - provides a practical alternative to these solutions by combining effectively the benefits of in-house development and outsourcing. This approach pools institutional resources to develop open- source applications, which dramatically reduces the development costs in HE organizations [3].

Rebuilding systems in LHE institutes from scratch would require serious investment. Enterprise Resource Planning (ERP) systems are a promising alternative. However, many colleges and universities have chosen to follow a non-ERP approach believing that such a strategy involves less risk and cost than implementing new systems and are better aligned with their goals, culture, and directions. On the other hand, the

disadvantage of high establishment and operational costs involved in adopting ERP should be avoided if Libyan universities choose to follow a collaborative development approach. Such an approach needs a group of enterprises which are similar in organizational structure, data flow, and business processes. Also, we should not forget that the concept of Information Technology (IT) has been shifting from possessing to consuming ever since Cloud computing (CC) emerged as a sustainable and promising solution to the challenges associated with shrinking IT budgets and escalating IT needs. Hence, Libyan public universities should benefit from this approach under the direction of the Libyan Ministry of Education (LMoE). In this study, an empirical analysis of critical success factors (CSFs) for the CD-ERP model was conducted at two levels, findings from the literature, and lessons learnt from international experiences.

2. Theoretical Framework

In order to study the CSFs for this model, the following questions were asked: Firstly, *what similarities and differences are there between approaches to system development?* To answer this, a comparison of such approaches is conducted. There are also sub-questions which need to be answered, including: What are the benefits and drawbacks of the CD-ERP approach and non-ERP approaches in the context of LHE? If Libyan universities abandon a non-ERP approach in favor of a CD-ERP approach, what are the risks and consequences? What benefits can be gained and what difficulties are expected when using a collaborative-development approach compared to separate projects being carried out by individual universities? Secondly, *how successful have international CD-ERP projects been and what can be learnt from their experiences?* To answer this question, experiences from international projects following a similar approach were included. The framework used to answer these questions is presented in Table 1.

Table 1. Theoretical Framework of the Study

Theoretical Framework	**1) Literature Review** Deductive Approach Data obtained from a review of the relevant literature and documents
	2) Studying International Experiences Deductive Approach Data obtained by studying projects based on a similar approach, including their documentation

3. Literature Review

A literature review can be interpreted as either a process or a product which is descriptive in nature [4]. This study uses a literature review to draw conclusions about the subject of the research area. The topic of this research is rather new and very few studies have been run so far, especially in the Libyan context. To our knowledge, this is the first academic research to be conducted on a model based on community-source in the context of Libya. Consequently, the title of the model proposed is based on its fundamental elements. Specifically, this research introduces a new model to Libyan universities which is mainly based on both ERP and the collaborative-development approach. The precise name of this model is "Collaboratively-Developed, Cloud-based Multi-tenant ERP Systems (CD-ERP)". Accordingly, attention is paid to ERP systems. System development is also discussed with an emphasis on collaborative approaches. Both Cloud computing and multi-tenancy are also covered.

3.1. Critical Success Factors for the Implementation of ERP in HE

CSFs for successful ERP implementation were selected based on a literature review regarding: (1) CSFs faced by different kinds of organizations during the implementation of a new IS, together with a specific focus on the HE field, since there are many similarities between the problems encountered in various fields, and (2) ERP systems, considering the advantages, disadvantages and unsuccessful implementation of such

systems. In summary, it was noted that most CSFs are related to managerial aspects rather than technical ones. This confirms that ERP software itself does not lead to unsuccessful implementation. The CSFs selected are classified by factor type in Table 2.

Table 2. Summary of Critical Success Factors (CSFs)

No.	Factor	Type
1.	Commitment and support from top management [5], [6].	Managerial
2.	Management of change [5], [7], [8], [6].	Managerial
3.	Project management [7], [5].	Managerial
4.	ERP adapted to a business's processes [5], [9].	Managerial
5.	Customization should be avoided, or at least minimized, to achieve the full benefits of ERP [5], [7].	Managerial
6.	Training and education [5], [7].	Managerial
7.	The participation and involvement of affected users (employees) [6], [10], [7], [11], [9].	Managerial
8.	Effective and efficient communication between different groups [12], [6], [7], [5], [9].	Managerial
9.	Teamwork and composition of the ERP team [5], [10], [7].	Managerial
10.	Strategic IT planning, vision, clear goals and objectives [5], [10], [7].	Managerial
11.	Selection of consultants and relationships with them [5], [7].	Managerial
12.	Cultural Factors [7], [8], [6], [9].	Managerial
13.	The ability of ERP systems to support business processes in HE should be ensured [8].	Managerial
14.	Selection of an ERP system and the choice of ERP architecture [5], [7].	Technical
15.	Selection of a vendor and customer support [7].	Technical
16.	ERP integration [5], [13] compatibility of ERP with the legacy system(s) must be considered [7], [8].	Technical
17.	The level of existing technology/infrastructure [7].	Technical
18.	Software development, testing, troubleshooting [7].	Technical
19.	Post-implementation evaluation: Monitoring and evaluation of performance [5], [10], [7], [8].	Technical

3.2. Critical Success Factors for a Collaborative-Development Approach

The dimensions of an environment of distributed ISs (communication, coordination, control, development, and maintenance) must be considered. Table 3 presents a summary of the CSFs for collaboration in IS development. These CSFs are classified by factor type.

Table 3. Summary of CSFs for Collaboration in IS Development

No.	Factor	Type
1.	Differences between processes [14]	Managerial
2.	Challenges to knowledge management [15], [14], [16]	Managerial
3.	Coordination costs [14]	Managerial
4.	Records of "organizational memory"	Managerial
5.	Group awareness [17], [16]	Managerial
6.	Appropriate communication [15], [16]	Managerial
7.	Project management [15]	Managerial
8.	Leveraging modularity and the use of cultural mediation [18]	Managerial
9.	Developing an effective tool base [18]	Managerial
10.	Compatibility of ICT systems [14], [15]	Technical
11.	Recognizing the importance of decision making [16]	Managerial
12.	The integration of different modules [14]	Technical
13.	Development time [14]	Technical
14.	Following a single lifecycle or process model [17]	Technical
15.	Convergence towards a final architecture and design [19]	Technical
16.	Reducing dependencies between engineers [17], [19]	Technical
17.	Recording of errors and their resolution	Technical
18.	Choosing appropriate models of collaboration [20]	Technical

3.3. Critical Success Factors (CSFs) for CC Adoption in the HE Sector

CC was added to the CD-ERP model, since it is a great benefit to HE, especially when universities intend to

work jointly [21]. The literature has highlighted a number of challenges concerning CC in any field. Table 4 summarizes the CSFs for CC adoption in the HE sector. These CSFs are classified by factor type.

Table 4. Summary of CSFs for CC Adoption in the HE Sector

No.	Factor	Type
1.	Legal and regulatory issues [22], [23].	Managerial
2.	Data lock-in and standardization [22], [24], [25], [26].	Managerial
3.	Performance monitoring [27].	Managerial
4.	Controllability: CC providers, rather than HEIs control the data [24], [21], [26].	Managerial
5.	Complex integration [24].	Managerial
6.	Organizational support [25].	Managerial
7.	The specificity of a service level agreement (SLA) [23].	Managerial
8.	Security, privacy, and trust [22], [27], [23], [28], [21], [29].	Technical
9.	Availability, robustness, and the ability to recover from a disaster [22].	Technical
10.	Resource management and energy-efficiency [22].	Technical
11.	Not all applications run on clouds [25].	Technical
12.	Low speed or lack of an Internet connection [25].	Technical
13.	Failure of a cloud service or a lack of compatibility of products and services. In other words, dependence on a specific provider [29], [21].	Technical
14.	Some CC providers only support particular languages or platforms [26].	Technical

3.4. Multi-tenancy in the HE Sector

The powerful capability offered by multi-tenancy has been described in the literature. In particular, it enables a variety of organizations to securely take advantage of one application in order to reduce costs and increase operational efficiency, while various degrees of isolation or sharing may be implemented [30], [31], [32]. Table 5 summarizes the issues related to multi-tenancy (drawbacks and benefits).

Table 5. Considerations for Developing a Multi-tenancy Architecture

No.	Factor
1.	Strictly isolated tenancy is needed to ensure the highest level of data security and scalability. Isolation should be carefully considered in almost all parts of the architecture from various aspects, both functional and non-functional, e.g. security, performance, and availability [33], [34].
2.	Careful assignment of an equal amount of resources to each instance [30], [35].
3.	The service level agreement (SLA) has to provide very strong guarantees of data security [34].
4.	Knowledge of relevant regulations when allotting responsibilities to meet legal requirements [34].
5.	Re-engineering current software applications to the multi-tenancy architecture adopted [31].
6.	Selecting the appropriate multi-tenant architecture [31].
7.	The ability of an application to serve a large number of tenants via one instance [34].
8.	Customization should not impact other tenants during runtime [33].
9.	Ability of tenants to deploy their existing applications on the multi-tenant architecture without a large change in code [33].
10.	Monitoring of service delivery and availability. One error could interrupt service delivery [33].

4. International Experiences

"Community-based open-source" is also described in short as "community-source" and in this research is called "collaborative development". It was first proposed by Brad Wheeler and defined as a type of open-source project governed by a group of educational institutions or even firms. In a community-source project, a consortium of partners share their financial efforts and human resources. Such a project is managed via a model of consortium governance [36]. Community-source is an environment where each partner must view other partners as non-competing, in order to share costs, risk, and potential rewards [37]. The development of community-source is considered to be an approach to developing applications, resulting in a unique type of collaboration from multiple organizations in a virtual environment [3].

In this section, real-life projects based on the community-source paradigm are briefly discussed in order

to better understand this approach. These cases offer us a great opportunity to understand the research issues of community-source projects adopted in HE, as well as to confirm the findings of the literature review. Initially, the community-source paradigm was applied by the Kuali and Sakai projects in the United States. Implementations of similar projects in Europe, Asia, and Africa have also taken place including; development work on Sakai at both Cambridge and Oxford in the UK, on Kuali at Strathmore University in Kenya, the Cineca Consortium in Italy, the Sigma Suite (a student information system) and the CRIS Argos Suite for research management in Spain, FS University Consortium in Norway, the AMUE system in France, Ladok in Sweden, JISC and UCISA in the UK, Surf in the Netherlands, AXIES in Japan, ASAUDIT in South Africa, CUCCI in Canada and HisinOne in Germany. Information on these cases, especially on their technological status, is difficult to obtain. Most of the information is not in English and unpublished. This indicates that little attention has been paid to community-source in the literature. Table 6 provides a summary of international projects.

Table 6. Examples of Collaborative-Development Projects in HE

Name of the Project	Founded	Initiator of the project	Source of funding	governing members	Features
Kuali Project – USA [2]	2004	a public and non-profit consortium of Indiana University, The University of Arizona, the University of Hawaii, Michigan State University, San Joaquin Delta Community College, Cornell University, NACUBO, and the rSmart Group	membership fees and dues	74 member institutions and the Kuali Foundation	• modular architecture based on Java (J2EE) • managed by the Kuali Foundation and became a single-company open-source project in 2014
Sakai Project – USA [38], [39], [40]	2005	a public and non-profit consortium of University of Michigan, Stanford University, Massachusetts Institute of Technology (MIT), University of California, Berkeley (UC Berkeley), and Indiana University. It was funded by a Mellon Foundation grant.	membership fees and dues	74 member institutions, coordinated by the Apereo Foundation	• modular architecture based on Java (J2EE) • developed on a centrally-planned model • moved towards open-source development and joined the Apereo Foundation in 2012 • commercial affiliates develop, host, and support Sakai
The USOS Project and the MUCI Consortium– Poland [41], [42]	2002	17 Polish public universities	membership fees and dues	representatives from 50 different types of HE institutions in Poland	• the USOS project is governed by a commission of MUCI members. • system development is driven by user needs reported daily to system developers. • based on a centralized Oracle database
Oodi Project – Finland [43]	1995	a public and non-profit consortium of University of Helsinki, Helsinki School of Economics, Helsinki	consortium fees collected from the member universities	representatives from 9 member institutions	• used by the majority of Finnish universities • the Oodi consortium is

		University of Technology, University of Oulu and Sibelius Academy.			responsible for maintaining and developing the system
CINECA Consortium – Italy [44], [45]	1969	non-profit consortium of Italian universities and research institutions	membership fees and dues	67 Italian universities and 9 research institutions	• the quality management system of Cineca has complied with the international ISO 9001 standard since November 2001.
LADOK Project – Sweden [43]	1970	non-profit consortium of Swedish public universities	membership fees	a consortium consisting of 37 HEIs in Sweden	• modular architecture based on Java (J2EE5) • DB2 database

5. Discussion

Consortiums between universities have existed for a long time, e.g. the CINECA project in Italy dates back to 1969 [44], [45], while the community-source paradigm was first applied by the KUALI and SAKAI projects in the United States. Other similar projects have been deployed in a variety of countries from North America, Europe, Asia, and Africa [36], [37], [46]. From the literature review, many of the problems observed in these projects are similar to those encountered outside HEIs, especially in regard to distributed and working environments, including:

- Generally, community source projects (which by nature create a distributed environment) are more challenging than non-distributed projects [47]
- The challenges and complexity of joint development result from the diverse needs of partners [48].
- Governance and the ownership of code are key issues [49].
- Project management is a critical issue when developing an IS using a community-source approach [3].
- The costs of coordination and communication between multiple partners within and across projects increases as the number of members increases and projects develop [2].
- Knowledge management is crucial in such projects [36].
- A certain level of face-to-face communication is essential, which could be difficult in some cases [36].
- Security concerns [48].

Furthermore, a study of the experiences of international projects indicates the following:

- Little attention has been paid to the community-source approach in the literature, since the authors have struggled to find relevant and reliable sources of information. Thus, more research should be carried out in the future to cover all aspects of the community-source approach.
- Community-source is an environment where each partner must view other partners as non-competing in order to share costs, risk, and potential rewards [37].
- As a natural part of the life-cycle of collaboration, a change in goals among the members of a community can lower the level of cooperation, or in some cases terminate a project. Some authors consider community-source to be a transitional phase rather than a permanent model. For example, Sakai moved towards open-source development and joined the Apereo Foundation in 2012, while Kuali evolved in a similar direction. The authors would argue strongly against this point of view. Sakai and Kuali are only two of many cases worldwide. In fact, a considerable number of other projects continue to use the collaborative approach, which indicates its success. On the other hand, Sakai and Kuali have reported a high number of implementations, both in the US and internationally. Many institutes asked to join these two projects, which reflects that the Kuali and Sakai projects are

based on a successful model, rather than being a transitional phase [36], [37], [50]. However, this point should be taken into consideration very seriously.

- Building a system on existing code that works at least fairly well, as was the case for Sakai and Kuali, is a successful approach. Such an approach negates the need for hard negotiations in regard to basic decisions regarding the architecture of a system, as well as increasing the likelihood that a project can achieve rapid success to prove its potential value [49]. This supports the CD-ERP model, which is based on ERP and projected to achieve early success and thus build momentum. Libyan universities are projected to obtain faster and reliable results as they will not need to code from scratch, especially as the fieldwork in Libya has shown the low level of the ISs implemented in the universities studied.
- Some projects (e.g. Sakai) are difficult to integrate with other software systems, such as ERP [51]. Hence, our model is based on ERP. As a result, ERP software would be part of the system.
- To guarantee rapid development, a core team of selected partners should be created [47]. Too large a number of partners might negatively affect the development process, due to e.g. diverse needs [46]. It would be better to create a team from selected partners (e.g. the leading universities in each province) to develop the core systems (or at least guide their development), without neglecting other partners' needs. The possibility of customization exists to fulfill such needs.
- When adopting a collaborative development approach, the structure of the consortium should be considered carefully. As mentioned before, HE institutes are unique and their needs are not well understood. Hence, collaboration with a group of such institutes is complex. This provides an interesting topic for future research. Accordingly, the model proposed here reflects the structure of some of the international cases.
- Closed community-source is often the preferred way of development in such cases, since it exhibits the benefits of both commercial and open-source software. Indeed, our model suggests that the rights to ISs developed via an open-source project will be granted exclusively to members.

6. Conclusion

Since a comprehensive review of all topics related to CD-ERP is not possible within the framework of this article, the basic elements of the CD-ERP model have been covered. The CD-ERP model is based on the community-source principle in which systems are developed jointly. The findings have shown that software development is no longer limited to an individual developer, but has to rely on a distributed network that uses computer-mediated collaboration and shares information. Communication, coordination, control, development, and maintenance are aspects that must be taken into consideration when dealing with geographically distributed teams, as in community-source projects. In addition, most of the CSFs for ERP implementation are also related to managerial aspects rather than technical ones.

Both Cloud technology and multi-tenancy architecture are separate topics from collaborative development. However, it has been noted that they can be naturally combined with collaborative development to reduce the costs of both investment in infrastructure and human resources incurred by the universities participating in the consortium.

Moreover, community-source has been adopted by many universities and government bodies worldwide. This raises the question as to what makes community-source widely popular, particularly among HE institutes. Traditionally, HE is a sector based on knowledge sharing. From this point of view, community-source is considered to be a means of sharing, not just knowledge, but also e.g. costs or human resources. Hence, community-source has been described as a "perfect fit" to the philosophy and standards of research and education. Moreover, based on the examples of international experiences, it is clear that collaborative

approaches to developing open-source applications can give results which meet the needs of the participating institutions, as well as greater potential to benefit the broader community. In particular, some consortiums in the HE sector, such as Kuali or CINECA, have a fairly robust tradition of building their own software. The HE sector is considered a unique environment with complex and poorly understood requirements. However, the community-source approach is projected to overcome the historically encountered gap between software producers and the HE sector.

Hence, this paper is an introductory study with a general scope. More research should be conducted on more specific topics such as: distributed ISs, international experiences of using the community-source approach, and the challenges related to the two main approaches to adopting ERP systems, namely: reengineering into ERP or customization ERP, with an emphasis on collaborative development.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

This paper is the end result of an ongoing doctoral dissertation. Tareq S. Almigheerbi is the Ph.D candidate, David M. Ramsey is the senior supervisor and leader of the doctoral project, and Anna Lamek is the assistant supervisor. This paper represents the findings from the literature review and the findings from similar projects worldwide. The tasks carried out by the individual authors are as follows: the research was conducted by Almigheerbi and Ramsey; the data analyzed by Almigheerbi and Lamek; the paper was written and revised by Almigheerbi, Ramsey and Lamek. Finally, all the authors approved the final version.

References

[1] Elferjan, M. (2015). *Development of Training Programs Provided for Academic staff of Libyan University.* Ph.D. Thesis, University of Salford, Salford, UK.

[2] Liu M., Sean H., & Tu Q. (2015). Toward sustainable collaborative software development: A case in higher education. *Proceedings of Twenty-first Americas Conference on Information Systems, Puerto Rico*.

[3] Liu, M., & Qiang, T. (2011). Community-based open source: The phenomenon and research opportunities. *Proceedings of 13th International Conference on Enterprise Information Systems,* Beijing, China.

[4] Bangert-Drowns, R. L. (2005). Literature review. *Encyclopedia of Evaluation*, 232-233. New York, Sage Publications Inc.

[5] Rabaa'i, A. (2009). Identifying critical success factors of ERP Systems at the higher education secto. *Proceedings of 3rd International Symposium on Innovation in Information & Communication Technology.* Amman, Jordan.

[6] Rani, S. (2016). A review of ERP implementation in higher education institutions. *International Journal of Advanced Research in Computer Science and Software Engineering, 6(6)*, 542-545.

[7] Villari, B. C., & Jharkharia, S. (2015). Critical success factors for ERP implementation: A classification. *Modern Applied Science, 13(1)*, 106.

[8] Abugabah, A., & Sanzogni, L. (2010). Enterprise Resource Planning (ERP) system in higher education: A literature review and implications. *International Journal of Human and Social Sciences, 4(11)*, 2120-2124.

[9] Seo, G. (2013). *Challenges in Implementing Enterprise Resource Planning (ERP) System in Large Organizations: Similarities and Differences between Corporate and University Environment.* Master thesis, Massachusetts: Massachusetts Institute Of Technology.

[10] Shatat, A. S. (2015). Critical success factors in Enterprise Resource Planning (ERP) system implementation: An exploratory study in Oman. *The Electronic Journal of Information Systems Evaluation, 18(1)*, 36-45.

[11] Jr, R. K. R., & Cegielski, C. G. (2005). *Introduction to Information Systems*. Hoboken, NJ: John Wiley & Sons, Inc, 2011.

[12] Zornada, L., & Velkavrh, T. B. (2005). Implementing ERP systems in Higher education institutions. *Proceedings of the 27th International Conference on Information Technology Interfaces ITI, Cavtat, Croatia.*

[13] Noaman, A. Y., & Ahmed, F. F. (2015). ERP systems functionalities in higher education. *Proceedings of International Conference on Communication, Management and Information Technology,* Prague, Czech Republic.

[14] Sengupta, B., Chandra, S., & Sinha, V. (2006). A research agenda for distributed software development. *Proceedings of International Conference on Software Engineering,* Shanghai, China.

[15] Shrivastava, S. V., & Date, H. (2010). Distributed agile software development: A review. *Journal of Computer Science and Engineering, 1(1)*, 10-17.

[16] Hyysalo, J. (2014). *Supporting Collaborative Development, Cognitive Challenges and Solutions of Developing Embedded Systems*. Doctoral Dissertation, Oulu: University of Oulu, Oulu, Universitat Isouluensis.

[17] Gutwin, C., Penner, R., & Schneider, K. (2004). Group awareness in distributed software development. *Proceedings of The 2004 ACM conference on Computer Supported Cooperative Work.* Chicago, Illinois.

[18] Lings, B., Lundell, B., Ågerfalk, P. J., & Fitzgerald, B. (2006). Ten strategies for successful distributed development. *Proceedings of IFIP International Working Conference on the Transfer and Diffusion of Information Technology for Organizational Resilience.* Galway, Ireland.

[19] Whitehead, J. (2007). Collaboration in software engineering: A roadmap. *Future of Software Engineering.* Minneapolis, MN, USA.

[20] Rafael Prikladnicki, R., Rodrigues, C., Azevedo, R. R. d., Junior, I. H., & Meira, S. (2011). Collaboration models in distributed software development: A systematic review. *CLEI Electronic Journal*, *14(2)*, 1.

[21] Kiryakova, G. (2017). Cloud computing – A necessary reality in modern education. *International Journal of Scientific and Research Publications, 7(4)*, 158-164.

[22] Voorsluys, W., Broberg, J., & Buyya, R. (2011). Introduction to cloud computing. *Cloud Computing: Principles and Paradigms,* 3-41. Hoboken, NJ, John Wiley & Sons Inc.

[23] Pandey, A. (2013). Role of cloud computing in higher education. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3(7),* 967-972.

[24] Raj, P. (2011). Enriching the 'integration as a service' paradigm for the cloud era. *Cloud Computing: Principles and Paradigms*, 57-96. Hoboken, NJ, John Wiley & Sons.

[25] Mathew, S. (2012). Implementation of cloud computing in education – A revolution. *International Journal of Computer Theory and Engineering, 4(3)*, 473-475.

[26] Islam, M., Kasem F., Khan, S., Habib, M., & Ahmed, F. (2017). Cloud computing in education: Potentials and challenges for Bangladesh. *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, *7(5)*, 11-21.

[27] Mohan, T. S. (2011). Migrating into a cloud. *Cloud Computing: Principles and Paradigms,* 42-56. Hoboken, NJ, John Wiley & Sons, Inc.

[28] Yadav, K. (2014). Role of cloud computing in education. *International Journal of Innovative Research in Computerand Communication Engineering, 2(2),* 2320-9801.

[29] Kumar, B. P., kommareddy, S., & Rani, N. (2013). Effective ways cloud computing can contribute to

education success. *Advanced Computing: An International Journal (ACIJ)*, *4(4)*, 17-32.

[30] Bezemer, C., & Zaidman, A. (2010). Challenges of reengineering into multi-tenant SaaS applications. Software Engineering Research Group, Delft University of Technology, Technical Report Series, Delft, Netherlands.

[31] Kabbedijk, J., Pors, M., Jansen, S., & Brinkkemper, S. (2014). Multi-tenant architecture comparison. *Proceedings of The European Conference on Software Architecture.* Vienna, Austria.

[32] Jani, K., Kumar, B., & Shah, H. (2013). Degree of multi-tenancy and its database for cloud computing. *International Journal of Engineering Development and Research*, *1(3)*, 168-171.

[33] Mukundha, C., Kavya, M., Reddy, O., & Tejaswini, R. (2017). A comprehensive study on multi-tenancy techniques in cloud computing models. *International Journal of Engineering Research and Development*, *13(9)*, 59-64.

[34] Matthew, C. (2014). A review of multi-tenant database and factors that influence its adoption. *Proceedings of UK Academy for Information Systems 19th Annual Conference.* Oxford, UK.

[35] Pallavi G. B., & Jayarekha P. (2014). Multitenancy in SaaS: A comprehensive survey. *International Journal of Scientific & Engineering Research*, *5(7)*, 41-48.

[36] Hanganu, G. (2018). The community source development model. Retrieved from http://oss-watch.ac.uk/resources/communitysource

[37] Wheeler, B., & Hilton, J. (2018). The marketecture of community, educause review. Retrieved from https://er.educause.edu/~/media/files/article-downloads/erm1261.pdf

[38] Alves, P., Miranda, L., Morais, C., & Alves, E. (2012). Proposal of a learning styles identification tool for sakai virtual learning environment. *Computer Science and Engineering,* 47-54.

[39] Sakai, (2019). *The Sakai Project*. Retrieved from https://sakaiproject.org

[40] Ignjatovic, M., & Jovanovic, S. (2013). Implementing sakai open academy environment - pros and cons. *International Journal of Emerging Technologies in Learning (IJET)*, *8(1),* 64, 2013.

[41] USOS. (2019). *About USOS.* Retrieved from https://www.usos.edu.pl

[42] Czerniak, M. (2010). Student management information system for Polish Universities at its tenth anniversary. *Proceedings of The 16th International Conference of European University Information Systems.* Warsaw, Poland.

[43] Feasibility-study. (2012). *Final Report, Feasibility Study: Modernisation of Student Information Systems, by University of Tampere.* Retrieved from https://blogs.helsinki.fi/otm-esiselvitys/files/2012/11/OTM-final-report-en.pdf

[44] Cineca. (2019). *Cineca Project.* Retrieved from https://www.cineca.it/en

[45] Vertiv. (2018) *Vertiv Case Study.* Retrieved from https://www.vertivco.com/globalassets/documents/case-studies/vertiv-case_study-cineca-it-2018_engb_249727_0.pdf

[46] Almajalid, R. M. (2017). A survey on the adoption of cloud computing in education sector, Cornell University library. Retrieved from https://arxiv.org/abs/1706.01136

[47] Hubner, U., Duda, I., Merz, Natusch, M., D., & Weckmann, H. (2008). HISinOne - Development and early adoption partnerships. *Proceedings of EUNIS 2008: The 14th Congress in a Series of Conferences within the Framework of the European University Information Systems Organization*, Aarhus C, Denmark.

[48] Liu, M., Hansen, S., & Tu, Q. (2014). The community source approach to software development and the kuali experience. *Communications of the ACM*, *57(5)*, 88-96.

[49] Courant, P. N., & Griffiths, R. J. (2018). Software and collaboration in higher education: A study of open source software. Retrieved from https://www.campussource.de/opensource/docs/OOSS_Report.pdf

[50] Feldstein, M. (2014). *Community Source is Dead.* Retrieved from https://mfeldstein.com/community-

source-dead

[51] Monarch, (2010). Open-source learning management systems: Sakai and moodle. *Whitepaper.* Retrieved from https://www.monarchmedia.com/wp-content/uploads/2015/01/open-source-lms-sakai-and-moodle.pdf

Tareq Salahi Almigheerbi was born in Tripoli/Libya in 1982. In 2004, he obtained a bachelor's degree in computer science from the University of Tripoli, Libya. In 2008, he obtained a master of science degree in information technology from the Faculty of Information Technology and Quantitative Science, Universiti Teknologi Mara Malaysia (UiTM) "Mara University of Technology", Shah Alam, Malaysia. The author's major fields of study are information systems, community-source, ERP and software development. Since 2012, he has been working as a full-time lecturer at the Department of E-commerce, Faculty of Economics & Political Sciences, University of Tripoli. He has developed many offline and online systems for University of Tripoli. Mr. Almigheerbi has worked as the department head of the office of information and documentation at the Faculty of Economics. He is the member of the IT committee at University of Tripoli.

David M. Ramsey was born in Cardiff, UK in 1967. In 1990, he obtained a bachelor's degree in mathematics from the University of Bristol, UK. In 1994, he obtained a PhD in statistics from the same institution. In 2005, he obtained a higher degree in economics from Warsaw School of Economics, Poland. David Ramsey's main fields of study are decision theory, in particular game theory, probability and statistics. He has worked at the Faculty of Computer Science and Management, Wrocław University of Science and Technology, Poland since 2012, where he was promoted to the position of university professor in 2016. He has also worked at the University of Limerick, Ireland and the University of Bath, UK. His current research interests include behavioral economics, dynamic games and applications of decision theory to the management and biological sciences. Prof. Ramsey is a member of the international society of dynamic games and the polish economic society.

Anna Lamek is an assistant professor in the Operations Research Department at the Faculty of Computer Science and Management, Wrocław University of Science and Technology.

In 2009, she obtained a master's degree with a subject titled "Information Technology in Management" at the Faculty of Computer Science and Management, Wrocław University of Science and Technology. In 2014, she graduated from Wroclaw University of Technology and received her Ph.D in management. Her research interests are forecasting, economics, decision-making, big data, IoT, game theory, fair division procedures, ERP systems.