Aug 26th, 4:55 PM - 5:20 AM

Role Of Industry Technologies in Information Systems Education

Shaloam Mutetwa
University of Cape Town, shalomutetwa@gmail.com

Gwamaka Mwalemba
University of Cape Town, gt.mwalemba@uct.ac.za

Role Of Industry Technologies in Information Systems Education

Shaloam Mutetwa, University of Cape Town, shalomutetwa@gmail.com

Gwamaka Mwalemba, University of Cape Town, gt.mwalemba@uct.ac.za

ABSTRACT

Determining the appropriate industry technology or software products[1] to use for practical teaching or demonstration purposes is a well-known challenge in technology-related education. As industry technologies continue to evolve and considering the vast number of existing proprietary as well as extant open source technologies, the task of selecting the best one for a particular competence, lesson, or course is also becoming more complicated. This research explores the selection and implementation of industry technologies in an Information Systems educational context. Data was collected through interviews with lecturers and students as well as analysis of course documents in an Information Systems Department at a South African university. The findings point to alignment with industry needs, affordability, availability of supplementary resources, and software functionality as some of the important criteria used by educators when selecting software products.

Keywords:

Industry technologies, software, information systems education

INTRODUCTION

The automation of business processes and the integration of technology in various business practices have increased exponentially over the last decade (Moscardini et al., 2020). Businesses are increasingly demanding not only a technologically-savvy workforce, but also trained information technology specialists who are equipped to design, implement, and maintain the various IT systems that support their business operations (Van Belle, Scholtz, Njenga, Serenko, Palvia, 2019). This demand has increased the demand for technology-focused faculties and departments in universities to produce graduates who are 'industry-ready' (Tan, Nakata, & Paul, 2018). It is now common knowledge amongst students and educators alike that the more industry-ready a graduate, the better their chances of getting hired. This has incentivised universities to increasingly prioritise the development of practical skills and competencies that align with industry needs (Ayalew et al., 2012).

One way of achieving the academia-industry alignment is by ensuring that students are exposed to not only theoretical knowledge, but that they also get a chance to experience working with technologies that are currently being used to support business operations in the industry. For some universities, this has meant incorporating

[1] The terms industry technologies and software products will be used interchangeably herein to refer to database, programming languages, operating systems and other (software) tools.

internship programmes into formal academic programmes. This allows students to leave campus to go spend time in the industry so as to be exposed to commonly used software products and practices (Dasuki et al., 2017). Another increasingly common approach used to expose students to what is needed involves incorporating industry or practitioner certification into a degree programme. With this approach, the university enters into a formal or informal arrangement with software vendors, who in turn grant students access to their certification programmes (Tan et al., 2018). This practice is increasingly being used to boost the appeal and international recognition of academic programmes, especially for newly established and/or privately owned universities. The third approach involves university educators incorporating as much practical work as possible into their courses, either in the form of tutorials or projects. The practical work involves interaction with complex business software that can expose students to scenarios typical of what they would be expected to encounter at the workplace. These could be tasks related to software development, viz. configuring or customising a business application, building data models, or assisting with system threats, to name just a few. In order to accomplish this, the educator needs to select and ensure access to software or a computer application with the necessary functionalities, using it to replicate a business or operational scenario that can be used for teaching purposes (Mwalemba 2019). This study aims to explore this third approach, used to expose students to industry technologies. Knowing that educators are often faced with many options and considerations when having to decide which software to use, this study attempts to answer the question W*hat influences the choice of industry technology to be used for practical purposes in an information systems course?* By answering this question, the study hopes to highlight the complexities, strategies, and challenges that form part of exposing Information Systems students to industry technologies in an attempt to create graduates with prerequisite technical know-how. The study will also be of use to educators and other industry stakeholders who might consider this approach.

The study begins with a brief literature review on the use of industry technologies in Information Systems education. Thereafter the research methodology is described followed by presentation of findings. The study ends with a discussion and conclusion chapter, which together synthesises and summarises both the findings and implications.

LITERATURE REVIEW

Information Systems Education and Industry Technologies

It is now common that any form of teaching typically makes use of some form of technology. Most commonly used technologies for teaching include the basic word processing software used to communicate instructional material to students, Powerpoint™ for presentations, and spreadsheets for basic calculations and number processing. Most institutions will also have learning management systems (LMS), which are typically used as a medium of interaction between learners and educators, but also as a repository for learning materials. These industry technologies, and many others, are commonly used across disciplines to facilitate teaching and learning.

However, technology-leaning disciplines such as information systems, computer science and engineering make use of additional advanced software packages and platforms to expose students to practical technology-related skills that often constitute part of their professional life. These could include technologies such as operating systems, database systems, programming languages and environments, enterprise systems and many other technology stacks. The aim is for students to become intimately familiar with these technologies, and to be able to not only use them, but also to design, build, implement, and maintain them. It is these technologies that constitute the focus of this study. Depending on the context, these could be open-source products or well-known proprietary software. Examples include making use of SAP™, Microsoft Dynamics™ or Odoo™ for teaching enterprise systems (Mwalemba, 2019; Zadeh et al., 2020), IBM BigInsights™ for teaching Data/Business Analytics (Zadeh, Zolbanin, Shadra, 2021), Python™ for programming (Smith & Jones, 2021), and AWS™ for cloud computing (Katiyar & Bhujade, 2018).

Selection of Industry Technologies for Teaching Purposes

The increasing number of software open source and proprietary software applications means that Information Systems educators are required to apply some form of selection criteria in order to decide which product or industry technology to use for their courses. In a few cases, educators can teach without having to rely on a specific technology. However, for educators to be able to coherently run their courses, they are often forced to select one or few industry technologies, and to use these to develop examples and practical exercises that can be assessed both competently and fairly (Mwalemba 2019). While there is a dearth of research exploring these criteria, a review of literature pointed to several aspects that can be attributed to influencing which industry technology is meant to be used in class.

Functionality

One of the criteria attributed to the selection of software products for teaching is the alignment between the intended competencies and the functionalities provided by the software. Scholtz et al. (2012) attribute the choice of SYPRO ERP to its alignment with the ERPEd framework used to design the MIS course. Similarly, Mwalemba (2019) highlights the demand for exposing students to advanced ERP skills as one of the motivations for partnering with SAP™, and creating a learning environment that will allow students to explore the full range of functionalities offered by the SAP ERP™ software.

Cost

Not all concepts can be sufficiently taught using open-source software. And even for open-source software, there are often costs associated with setting up and hosting the application before it can be used in the classroom (Mwalemba 2019). Hence, some educators' choices, especially in developing countries, are constrained by what they can afford. Certain software companies have specific initiatives that offer software to universities at discounted prices in some cases without cost (Hawking, 2010). However, most discounted or free proprietary products often come with limitations in terms of users, time, or functionality, which can significantly limit the academic value of the software application.

Ease of Use

Some of the software products used need educators to have significant product-specific technical knowledge for the product to be appropriately set up and used for academic purposes. This is often challenging, since educators have multiple responsibilities, such as research and administration, and therefore do not have enough time that can be dedicated to mastering one product solely for the purpose of using it for teaching (Mwalemba, 2019). On the other hand, Scratch's user-friendly interface is noted as a strength that makes it useful for teaching basic programming and computational skills (Scullard et at. 2019).

Industry Relevance

This refers to the degree to which the software product under consideration has been embraced by the professional community (Westfall, 2012). There is a strong belief among some educators that students ought to be taught content that aligns with the current practices, and offers them skills in software products used in the industry (Bain, 2017; Recker & Rosemann, 2009). Oracle™ is one of the most common software products to be used in undergraduate database courses now. This is due to its widespread use in industry and the edge it gives students when looking for an entry-level job (Beise, 2006).

Gap in Literature

The issues outlined above provide some insight regarding some of the thinking behind the choice of technology used to support teaching. However, most studies reviewed that involve the use of industry technologies do not explicitly disclose the reasons for selecting a specific software product from a vast pool of other similar ones. It is also clear from our literature review that existing research on the use of industry technologies in Information Systems education aims for the most part to share experiences with using a particular technology for teaching certain concepts (Hwang, 2018). Other studies have also focused on evaluations or analyses of the suitability of specific

technologies when teaching specific concepts from either a pedagogical or affordances lens (Murire, Cilliers, Viljoen, 2019; Zadeh, Zolbanin, Shadra, 2021).

This study expands on prior research by determining the criteria relevant to the appropriate industry technology while teaching in a given context.

METHODOLOGY

The research was conducted as a case study focusing on a Department of Information Systems at a South African university. Typical to case study methodology, data was primarily collected through semi-structured interviews with academic staff, as well as via the analysis of curricular documents. Most of those interviewed were 'course conveners', a title used within the university to refer to the academic staff responsible for the overall running and administration of a specific course. While a course may have multiple academics teaching on it, it only has one convener providing educator oversight. Interviews were recorded, transcribed, and analysed using a thematic analysis approach to identify themes that emerged from the collected data. NVIVO software was used to facilitate the data analysis process specifically for organising transcripts, documents, and the relevant codes.

Situated at the intersection between a business/commerce and information technology faculties, the Information Systems Department used as a case study offers various Information Systems related qualifications at undergraduate as well as postgraduate levels. The study interviewed ten academic staff. The next section presents a synthesis of the findings obtained from the analysis of the relevant documents and interviews.

RESULTS

Results point to more than 25 different software products (including programming languages) used across different undergraduate information systems courses offered by the department. A majority of the software used was proprietary. Figure One shows the percentage share of proprietary versus open-source products. There is also a substantial Microsoft® footprint in the products used across different courses.

Course	Content	Technologies Used
Foundation of Information Systems	Introduction to Information Systems, Introduction to Data Analysis, Introduction Databases, Introduction to Web Development	Microsoft Excel™, Microsoft Access™ Python™, HTML, Anaconda™, Navigator™
Commercial Programming	Planning and Developing Software Programmes Using C#	Microsoft Visual Studio Code™, Apache™ Studio IDE™, Miro™
Accounting Information Systems	IS for Accounting	Pastel™
Business Intelligence and Analytics	Business Intelligence and BA, Data Warehousing, Data Marts, Decision Support Systems, OLAP, Data Mining, Data Visualisation	Tableau™
Applying Database Principles	Database concepts, Advanced Database Design and Implementation, Database Architecture, Modelling	Microsoft Excel™, SQL™, Microsoft Access™
System Analysis	Object-oriented Systems Analysis	Microsoft Visio™, Miro™
Systems Design and Development	Implementation of Systems Development Process	Microsoft Visual Studio Code™

IT Architecture	Security and Networking, Cloud Services, E-Commerce	IBM Security Module™, Wix™
Systems Development Project	Project Management, People Management, Capstone Project (Problem Analysis and/or Software Development)	Microsoft Visual Studio 2019™, Microsoft Visual Studio Code™, C# Microsoft Azure Cloud™, Aris™, Microsoft Projects™
Electronic Commerce	Website Design, Business Models and Strategies, Business Use Cases, Web Development	HTML, CSS, Wix™, WYSIWYG, Vue.JS™
BPM and Enterprise Systems	Enterprise Systems, Business Process Modelling	SAP, Aris™, Microsoft Visio™, Tableau™

Table 1: The List of Software Products and Programming Languages Used Across Undergraduate Information Systems Courses

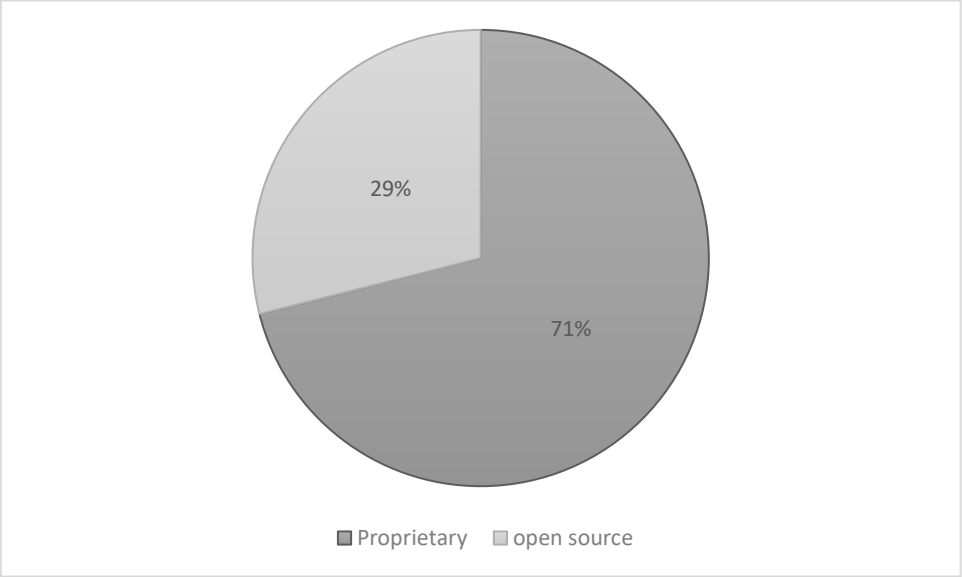

Figure 1: Proprietary vs. Open-source Software Products Used by Undergraduate Information Systems Educators

Criteria Influencing Selection of Software by Undergraduate Information Systems Educators

Industry Needs

The most frequently mentioned criteria when selecting software products is examining which software is widely used in the industry. Educators hope to expose students to software products that they are likely to encounter when they graduate and join the industry. Similarly, educators want to make their graduates marketable and increase their chances of securing competitive, high-paying jobs. Hence. there is always an attempt to use software products that industry partners may deem essential when screening students for potential employment. Responses are cited verbatim.

> *L03: "That project was designed over 10 years ago with Ernst and Young. And they have a big division, which audits particularly in the SAP space, so we were always supportive of using SAP as a tool because we know Ernest and Young, and most firms wanted their consultants to be able to work in that space."*

> *L07: "So there was no other department saying you need this when it came to ERP tools, SAP is the one with the high salaries, are most demand most used, so it's sort of the market leader but also, when I started lecturing, the university had an agreement that we could use our training system at the university at no cost, so it was easy."*

Some educators cited the popularity of the product as a reason for selecting it. It is however not clear how this popularity is assessed.

> *L08: "It's very much around the popularity. OK, two reasons. The first one is the popularity, in that Python is quite a popular coding language, not only for IS or technology practitioners, so it made sense because it's a service course that deals with most other BCom and humanities students who learn a coding language that's relevant to them. You know, in the work environment that was the one thing."*

Overall, it was reported that ensuring students graduate with familiarity with in-demand software products in the industry is an important consideration across the Department. There is an active effort by educators to keep up with the latest developments in the industry and try to tailor the curriculum according to the industry's requirements.

Institutional Relationship with Software Vendors

Some software vendors have formed institutional wide relationships that have allowed their products to be easily accessible within the University. Such arrangements often means there are also internal resources and or relationships that can easily be leveraged to obtain additional software products, and most importantly, technical support. This influences educators' choices, where making use of a given product often means less effort to set up the computing environment for teaching (e.g., installing software in computer labs), as well as the availability of necessary support, should the need arise.

> *L05: "To some extent, it also depends on the universities' choice of technologies. So, the university has an ICT service firstly. And then the faculty level. We have Commerce IT* [IT Support Unit]*, and so they responsible for the ICT infrastructure and labs and so on, and software and so to some extent we use. Uh, whatever is offered so in terms of like the productivity suites I mean, using Microsoft Office, why do we use Excel in the test your programme. Because that is the tool that's available to teach data analysis."*

Cost

For software that has cost implications for the university, it is often the role of the lecturer to negotiate costs with the software vendor and motivate the department or university to cover the costs for such software. There are also circumstances in which, for one reason or another, the University may elect to reject the request to purchase a particular piece of software.

> *L03: "The only problem is that the University doesn't have a licence, it's a very expensive product. But it's brilliant. ..because it's such as unique and specialised application, not all departments are using it. But what the company did for me, which I really appreciate was fantastic. They give free education licenses for when you have 100 users. And when I explained to them my situation with 400 students. And what we going to do, they increased my free subscription to 200 users."*

Whenever possible, educators prefer to avoid the administrative burden that comes with having to continuously motivate and prepare the necessary paperwork required for the university to cover the software costs. In such cases, educators often opt for that which is either freely available or is included in another university-wide arrangement with the relevant software vendor (see the institutional relationship with software vendor theme].

Availability of Support and Resources

Using a software product in class often requires a lot of effort, not only in setting up the software, but also in troubleshooting for students. There is also the concern that most of this software is not designed for teaching purposes and hence might require a special set-up for them to be suitable for classroom use. Furthermore, most of the software is constantly being upgraded and requires educators to have easy access to the training and learning/teaching materials for upskilling and updating course material.

> *L01: "With ERP systems, lecturers have to come up with the teaching material, which can sometimes be difficult. SAP has a wonderful university alliance programme, where SAP provides comprehensive support to faculty, e.g., faculty have access to an online curriculum repository, free faculty workshops, as well as the ESEFA programme. Here, lecturers are able to exchange teaching material which includes class exercises and tutorials."*

The availability of appropriate resources and support from either software vendors or a community of other academic users is an important criterion when selecting software. Furthermore, the availability of online tutorials, forums, troubleshooting guides, and support on the internet also makes it a better option, as students can easily turn to documentation as support when they encounter difficulties.

Functionality/ Feature set

Simply put, educators prefer to use a tool that has the appropriate functionality in order to demonstrate or cultivate competencies that form part of their course's learning outcomes. In addition, educators are also looking for software products that align with the pedagogy used, making things like collaboration, monitoring, and assessment possible, or less cumbersome.

> *L08: "I would say, because of the practical aspects of systems analysis and systems development and what we have now done in systems analysis to make it alive is we've used Miro. A wonderful application, where students can collaborate on whiteboards in the cloud. And I found Miro as quite effective for that, because you can see how you draw on the mirror board and everyone can see it so they can go into teams they can open a Miro board. And they can talk while they are on MS teams, and they don't have to share their screens. They can all have the Miro application open and then collaborate on that while talking. So, it's almost like you have a scenario that you would have in a practical workshop, which we do with systems analysis, where in real life students get to also do the work in teams. The tool mimics that, and allows students to master systems analysis."*

What can be inferred from this is that even if a software product is free and easily accessible, if it doesn't have the necessary functionalities, then it is not useful. This is often the case for software that is available only as a free trial.

DISCUSSION

The findings of this study highlight that simply having access to free or cheap software is often insufficient, especially if you intend on cultivating advanced technical competencies through exposure to complex systems and use cases. Technologies are continuously being updated, and they require advanced product-specific skills to set up and support in an academic setting. This is often challenging for educators who have multiple priorities. What is often needed is the appropriate technical support that understands the academic needs, training for educators, and the availability of additional resources that can be utilised in an academic context. These can be either set up by the software vendor or can be organised by a community of educators using the software for academic purposes. It can also be a combined effort between the software vendors and a community of educators (Mwalemba, 2019).

Alignment with industry needs is an important, but complex issue. It is clear from findings as well as from literature that using top-tier, widely-used software products helps to address the well-documented challenge of alignment between industry needs and what is taught at university. It also directly benefits students by ultimately making them more appealing to employers upon graduation (Dasuki et al., 2017 & Tan et al., 2018). On the other hand, it is an approach that privileges the status quo. Large companies with big budgets that allow for discounts or the funding of education initiatives that rely on their software have an upper edge. This could create a situation where students are rarely exposed to either open-source software products or software products developed by SMMEs without enough resources to set up initiatives to support complex university needs without any short-term benefits. The existence of a pool of skills on a particular product improves its business case and subsequently can positively influence its adoption rate (Sewchurran et al, 2012).

Another important issue highlighted in the findings is the lack of specific guidelines for determining how software ought to be selected. This is consistent with the literature, as we struggled to find examples where institutions or departments had a clear policy or guide on how educators can select software products for teaching purposes. This leaves educators to make software choices in their individual capacity. On the one hand, this makes the process of choosing software less bureaucratic. On the other hand, it is possible that, as pointed to by some of the findings in this study, educators tend to opt for that which they find either valuable or most convenient to themselves, and not necessarily for something that has an objective improvement or is aligned to the departmental or institutional vision, mission, and values. As illustrated in this study, the lack of policy in the department studied has inadvertently led to it using predominantly proprietary software from a single large software vendor. It is a question as to whether this aligns with the Department's vision, mission, and educational values. More pointedly, it is a question as to whether such exposure in any way shapes students' future preferences, perceptions, and values as far as software choices and usage are concerned.

CONCLUSION

In light of the increasing usage of software products in Information Systems education, this study set out to explore the criteria used by educators or institutions to select appropriate software products to use in the classroom for demonstration or practical purposes. Findings point to alignment with industry needs, affordability, availability of supplementary resources, and software functionality as some of the important criteria. However, what is also clear is that these criteria are often intertwined, making the decision even more complex. That which proves cost-effective sometimes lacks the necessary functionalities, and in some cases, that which is popular in the industry is either too expensive or too complex to be used in academic settings. There are also underlying interests and values that are associated with these decisions. An analysis of the impacts of these underlying interests and embedded values lay outside the scope of this study but may prove an interesting area for future studies.

Lastly, it's important to acknowledge that this study was conducted at one of the top-ranking universities on the Continent. This university has more extensive access to funding compared to most in the region. It also has a strong brand that is well-recognised in the industry. The circumstances discussed in the study should therefore be read within that context.

REFERENCES

1. Ayalew, Y., Renken, J., Mgaya, K. V., & Nkgau, T. Z. (2012). Developing a contextualized information systems curriculum for an emerging economy. The Electronic Journal of Information Systems in Developing Countries, 54(1), 1-19.
2. Bain, L. Z. (2017). How do Information Systems (IS) programs prepare students for entry level occupations in the computer and IT industry? *Issues in Information Systems*, *18*(3), 78–88.
3. Beise, C. M. (2006). Revisiting database resource choice: A framework for DBMS course tool selection. *Association for Information Systems - 12th Americas Conference On Information Systems, AMCIS 2006*, *4*, 2121–2126.
4. Dasuki, S. I., Quaye, A. M., & Abubakar, N. H. (2017). An evaluation of information systems students internship programs in Nigeria: a capability perspective. *The Electronic Journal of Information Systems in Developing Countries*, *83*(1), 1-19.
5. Fedorowicz, J., Usoff, C., & Hachey, G. (2004). Teaching Tip: Twelve Tips for Successfully Integrating Enterprise Systems Across the Curriculum. *Journal of Information Systems Education*, *15*(3), 235.
6. Hawking, P. 2010 The Dilemma Of Addressing SAP Skills Shortages In Developing Countries. In Proceedings of the Proceeding of the Sixteenth Americas Conference on Information Systems (Lima, Peru, 2010), 1-7.
7. Hwang, M. I. (2018). Relationship between Teamwork and Team Performance: Experiences from an ERPsim Competition. *Journal of Information Systems Education,* 29(3), 157-168.
8. Katiyar, N., & Bhujade, R. (2018). A Survey : Adoption of Cloud Computing in Education Sector.

International Journal of Computer Trends and Technology, *60*(1), 15–25. https://doi.org/10.14445/22312803/ijctt-v60p102

9. Murire, O., Cilliers, L., Viljoen, K. (2019). An Evaluation of Social Media Use in the Classroom at a Traditional University. In: Kabanda, S., Suleman, H., Gruner, S. (eds) ICT Education. SACLA 2018. Communications in Computer and Information Science, vol 963. Springer, Cham.
10. Mwalemba, G. (2019). Confronting Challenges Facing Enterprise Systems Education in Africa. *2nd International Conference on Next Generation Computing Applications 2019, NextComp 2019 - Proceedings*. https://doi.org/10.1109/NEXTCOMP.2019.8883674
11. Recker, J., & Rosemann, M. (2009). Teaching business process modelling: Experiences and recommendations. *Communications of the Association for Information Systems*, *25*(1), 379–394. https://doi.org/10.17705/1cais.02532
12. Scholtz, B., Cilliers, C., & Calitz, A. (2012). A Comprehensive, Competency-Based Education Framework Using Medium-Sized ERP Systems. *Journal of Information Systems Education*, 23(4), 345-358
13. Scullard, S., Tsibolane, P., & Garbutt, M. (2019). The role of scratch visual programming in the development of computational thinking of non is majors. In Pacis (p. 79). https://core.ac.uk/download/pdf/ 326833249.pdf
14. Sewchurran, K., Mwalemba, G., de la Harpe, A., & McKinnell, J. (2012, May). Making sense of the IT/IS industry to enable better business enablement and competitiveness: A case-study of the Western Cape in South Africa. In *Proceedings of the 2012 4th IEEE Software Engineering Colloquium (SE)* (pp. 18-20). IEEE.
15. Tan, Y. L., Nakata, K., & Paul, D. (2018). Aligning IS Masters Programs with Industry. *Journal of Information Systems Education*, 29(3), 169-182.
16. Van Belle, JP., Scholtz, B., Njenga, K., Serenko, A., Palvia, P. (2019). Top IT Issues for Employers of South African Graduates. In: Kabanda, S., Suleman, H., Gruner, S. (eds) ICT Education. SACLA 2018. Communications in Computer and Information Science, vol 963. . Springer, Cham.
17. Westfall, R. D. (2012). An employment-oriented definition of the information systems field: An educator's view. *Journal of Information Systems Education*, *23*(1), 63–69.