Position papers of the 2014 Federated Conference on Computer Science and Information Systems pp. 111–115
DOI: 10.15439/2014F467
ACSIS, Vol. 3

Proprietary versus Open Source Software in Support of Learning in Computer Science

Antoine Melki
University of Balamand
Department of Computer Science,
Faculty of Sciences
P.o.box 100, Tripoli, Lebanon
Email: amelki@balamand.edu.lb

***Abstract*—the topic of learning Computer Science had been the subject of many researches. From time to time, this topic flows to the surface especially when Computing instructors report students' difficulties in acquiring the necessary knowledge. On the other hand, different studies had tackled the issue of open source software in computing education from different perspective. These studies, in general, conflict with the positions taken by the different suppliers of proprietary software. This paper will investigate the contributions of both categories of software in the process of computer science learning, and then will compare these contributions to the principles of active learning in computer science, to conclude which of the 2 categories is more advantageous.**

I. Introduction

DIFFERENT studies had tackled the learning model in computer science and came with different results. Despite that, active learning finds the highest amount of support among the scholars who engaged in this line of studies. In the technology field, the proprietary software allies, who are mainly the companies that produce it, claim relying on active learning and providing the best computer science education. On the other hand, with the increase in the popularity of open source software, the allies mark the same claim. This position paper represents a descriptive study of what both software categories claim to provide to computer science education. A comparative revision concludes the paper. The value of this study lies in providing a theoretical background to support the adoption of open source software in teaching computer science.

II. Learning

A. Definition and theories

Learning is the process of acquiring modifications in existing knowledge, skills, habits, or tendencies through experience, practice, or exercise. Learning includes associative processes, discrimination of sense-data, psychomotor and perceptual learning, imitation, concept formation, problem solving, and insight learning.

Gestalt-psychology researchers drew attention to the importance of pattern and form in perception and learning, while structural linguists argued that language learning was grounded in a genetically inherited "grammar." Developmental psychologists such as Jean Piaget highlighted stages of growth in learning. More recently, cognitive scientists have explored learning as a form of information processing, while some brain researchers, such as Gerald Maurice Edelman, have proposed that thinking and learning involve an ongoing process of cerebral pathway building. Among the related topics of research is transfer of training which is defined as the activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something [1].

Learning theories tend to fall into one of several perspectives or paradigms, including behaviorism, cognitivism, constructivism, and others [1].

Cognitivism is based on the idea that the mental function can be understood and the learner is viewed as an information processor. Cognitivism focuses on inner mental activities. It is necessary to determine how processes such as thinking, memory, knowing, and problem-solving occur. People are not "programmed animals" that merely respond to environmental stimuli; people are rational beings whose action are a consequence of thinking. Cognitivism uses a metaphor of mind as computer where information comes in, is being processed, and leads to certain outcomes.

Constructivism's basic idea is that learning is an active and constructive process where the learner viewed is an information constructor. People actively construct or create their own subjective representations of objective reality. New information is linked to prior knowledge, thus mental representations are subjective.

Another concept in learning which is relevant to this study is the concept of model which is a theoretical construct or mental picture that helps one understand something that cannot easily be observed or experienced directly. Models are testable ideas created by people that capture a story about what is happening in nature [2].

B. Learning Computer Science

The literature on learning in Computer Science education is large and much diversified. What is common among most

of the paradigms is the emphasis on active and collaborative learning. This is based on a common trait of almost all the educational psychology literature on learning in recent years which admits that learners construct their knowledge by interacting with their environment and other people. This is close to constructivism where some focus primarily on the individual learner and others focus primarily on the social nature of knowledge construction. In either case, the consensus is that education is not the mere transmission of knowledge from the teacher to the student but requires that students be active. Furthermore, collaborative learning is becoming a key component in many college classrooms with several benefits: improved achievement, enhanced critical thinking competencies, improved attitudes towards the subject area, and reduced student anxiety.

In the case of Computer Science, students spend the majority of their time solving problems that require them to learn skills that are applications of the concepts from the readings of manuals or classes. Neither the teacher nor the teaching Assistant is a pure lecturer, but assumes the role of facilitator in many cases, especially in Lab settings. Solutions require students to learn and practice new skills. Each problem builds on previous skills and concepts, extending the range of the students' capabilities. Generally, especially with the presence of social networks, students spend some time discussing possible solutions to the problem in groups, in pairs or individually before attempting to solve the problem. It is known that students compare results and discuss problems they encounter and solutions they discover. The transfer of learning is a major issue in learning Computer Science, since the application of learning is always a real life case; where requirements and specifications are not explicitly clears [2].

III. Proprietary Software And Learning

Companies providing proprietary software generate revenue by selling products in the form of software licenses and additional services and support. Accordingly, their business model is built with a very small space left for free offering. This has an effect on the companies views of education and consequently on how they see learning and strive to support it. In this section, the 3 major proprietary companies are selected as an example to support the analysis: Cisco, apple and Microsoft.

A. Proprietary Companies' Education

Cisco sees that there are pressures on education in the 21st century, mainly because education needs to change in order to adapt to the needs of the 21st century. One driver of these pressures is digital technology. Because the new technologies demand a new set of skills, digital technologies exert pressure for change but at the same time provide opportunities for transforming pedagogy because they offer access to information, networks for communication, and new means of presenting learning. A set of other factors also apply pressures on education. Globalization is one factor that exerts social and economic pressure, and provides opportunities for wider, richer learning. Other factors include economic recession, demographic pressures, and environmental stability. In responding to these pressures, educators are coming up with innovations that enrich learning and help them in dealing with specific challenges.

Meanwhile, these pressures and opportunities require people to acquire new kinds of literacy including information literacy, cross-cultural literacy, and ecological literacy. Learners are expected to be lifelong learners, because technology, politics, economics, and the environment are changing so quickly. This demands a shift away from focusing on engagement in school, to engagement in learning. It also requires an examination of what sorts of environments are most conducive to learning in the 21st century.

Cisco proposes a strategy that yields to the formation of a learning ecosystem. It claims building its strategy on the latest thinking and research about innovation, and it believes this strategy is of practical use to system leaders in education. This proposal is built on the distinction between formal learning and informal learning contexts, and education provision delivered by existing providers and new entrants. Then in another stage comes the application of new resources and new sources of insights to the education landscape, specifically, the new resource of digital technologies, which have the potential to radically transform learning, though they are not often used to their full potential. The proposal also addresses learner ownership, because when learners feel ownership of their learning, they are able to apply their own insights about how they learn best, and become "co-producers" of learning rather than just "consumers." On the other hand, system leaders need to reposition themselves so that rather than being primary providers of education, they provide a platform for a diversity of providers [3].

Apple concentrates more on hardware and emphasizes the services rendered to instructors who can customize students' devices with materials that fit their level and learning style so that the machine becomes a more powerful learning tool. Apple claims that its services allow more creativity, less time of preparation, and teaching with content from top institutions. For students who learn best by listening, the instructor is advised to download a podcast from iTunes. And for those who learn through tactile interaction, the instructor is advised to find an app. Apple also claims that the wide range of content across subjects and grades also makes it easy to tailor apple machines for students at a variety of learning levels so that the instructor can teach all the students the same lesson in different ways [4].

Microsoft focus is more on systemic education in order to produce a more efficient and effective learning environment, advocating sophisticated metrics to measure results. Microsoft looks at making the teachers relying on it better at their jobs than another and how can best practices be shared?

Technology enables analysis and is also the delivery mechanism.

Microsoft is concerned more with the science of education than the art of learning. Its products are claimed to facilitate students' ability to consume and create content and collaborate across multiple devices so that the learning process is extended beyond the classroom [5].

As it is made clear, proprietary organizations deal more with products than with learning per se. Even learning is a product. In conclusion, the principles of learning presented by proprietary organizations can be summarized as follows:

1. Emphasis is placed on altering learning patterns and models instead of moving learners
2. The level of freedom given to the users does not go beyond changing some predefined options; accordingly, the user's contribution is limited and directed
3. Technologies have predefined roles.

B. Proprietary Software and Learning Computer Science

The majority of the companies that sell proprietary software have their own professional schools. Many educational institutions contract with these companies for an exchange of technology with the curriculum. So the educational institution becomes bound to teach the curriculum supported by the corresponding technology of the partnering proprietary company. On the other hand, the company offers its product for free to the institution. Some institutions go further than that by adopting a curriculum structure analogous to that of the certificates offered by the proprietary partnering company. The result of this process is graduates who are specialized in the product of these companies.

In addition, these companies release, from time to time, offers of cheap or free software for students. All this ends in enlarging the share of the corresponding companies on the human capital that include developers, software engineers, database designers, systems analysts, information systems specialists, etc.

Further than that, proprietary companies create their own communities, competitions and events that are targeted at attracting students to their product. Examples include: Microsoft Imagine Cup and Cisco Networking Academy NetRiders.

Accordingly, the features of proprietary companies' education include:

- Content is added, edited and updated by the company
- Materials are the product of one author which is the company itself
- Releases and updates the result of a process controlled by the owning company.

In summary, the learning provided by the companies that sell proprietary software is a controlled learning.

C. Open Source Software and Learning

The prevalence of the Internet as of the early nineties has facilitated the collaboration between software programmers from different poles of the world, allowing an easy distribution of their production. In addition to that, the distinct advantages offered by Free/Libre and Open source software (FLOSS) resulted in an increasing recognition and adoption of this category of software.

The impact of FLOSS is discussed in many studies. International Data Corporation (IDC) gives an estimate of 22.4% growth in revenues in 2013, reaching 8.1 billion dollars. Comparisons show an increase in the rate of adoption of FLOSS as compared to proprietary software in operating systems and web servers, while proprietary, mainly Microsoft and Apple, are keeping the lead in desktop usage [6].

The studies describing the use of FLOSS in education show a set of principles [7]:

- Content is not something static but dynamic
- Learning resources are manifold
- Users are also active creators
- Support and learning resources are closely connected
- Open and transparent structures foster re-use and discourse, but also continuous improvement and evolutionary growth
- Existence of a wide range of possible activities to engage at around the core product
- Self-studying and learning from what others did are the pre-dominant form of learning.

D. Open Source Software in Computing Education

The penetration of open source software in computing schools and departments is natural, and so it is increasing, with some of the major players in the software market increasing the amount of revenue they make from activities that use FLOSS. A non-exhaustive summary of the advantages of FLOSS in teaching Computer Science can be:

- Lower costs for both the university and the students
- Projects are more beneficial for research since the user of FLOSS is free to get the source and implement new ideas.
- FLOSS is valuable for teaching as it offers students the opportunity to share their contributions to projects [8].

The main disadvantage in using FLOSS is the lacks of hardware drivers especially for wireless, graphic cards and suspend/sleep functionality in laptops. On the other hand, studies show that students' feedback on using FLOSS is mixed. While many students get excited to open source tools, others consider that learning a proprietary tool gives a better chance to get a job.

As for the instructors, studies showed that adopting FLOSS is a challenge due partly to some misconceptions. A non-exhaustive list of the issues that have to be overcome in teaching FLOSS can include [9]:

1. Misconceptions concerning the adoption of FLOSS, like:
   a. Only hobbyists use FLOSS.
   b. It is a niche market with limited diffusion.

A number of studies showing current FLOSS penetration rates in the market prove the fallacy of these misconceptions.

2. Lack of managerial support at the level of department or school, which is usually overcome by a decision to substitute some or all proprietary products by open source.

3. A misconception concerning the quality of FLOSS is that proprietary software is better and students learn more by using better software. This is not true as some FLOSS is among the leaders in their market segment as shown by reviews from publications in the field. On the hand, from an educational point of view, there should not be any difference in what can be taught using FLOSS or proprietary software, as concepts are the same [10].

On the other hand open source in computing education show that a learning environment is formed around communities. While this applies more to informal education settings, schools and departments can still participate and contribute, as this belongs to the core principles of FLOSS. These communities can be described as follows [11]:

- Dealing with up to date content where everyone can add, edit and update the content
- Materials are usually the product of many authors with many contributions from people other than original authors
- Relying on frequent releases and updates where product features and community structures are the result of a continuous process of renegotiation and reflection within a continuous development cycle
- Reusing of prior learning outcomes and processes, as these are systematically available through mailing lists, forums, commented code and further instructional materials
- The community members represent a large support network that voluntarily function in a collaborative manner
- New solutions are adapted early by the community.

IV. Active Learning and Computer Science

In Computer Science, it is crucial to create an active learning environment to improve students' comprehension and retention of material, allow students to take control and regulate their own learning, and eventually empower them with necessary skills to solve problems outside of the classroom. Over the years, various strategies have been developed by Computer Science instructors to promote active learning [12].

A literature review would find many definitions of active learning; however, several essential components are common between the different definitions:

1. Active learning is not the passively listening to lecture, where students apply material to "real life" situations [13].

2. Usually, an active learning environment allows students to talk about what they are learning, write about it, relate it to past experiences, and apply it to their daily lives, "they must make what they learn part of [14].

3. Supporting active learning is necessary because "what students learn is greatly influenced by how they learn, and many students learn best through active, collaborative, small group work inside and outside the classroom" [15].

4. As Briggs points out, in a rapidly changing field as Computer Science, "students tend to be active and visual learners" and it is beneficial to provide a learning environment where students can interact with the material. "Active learning is especially effective for CS students who tend to be visual/intuitive learners." [16]

5. Active learning can incorporate many instructional methods, such as collaborative/ cooperative, project/ problem-based learning, role play, debates, etc. and even the use of functional emerging instructional technology teaching tools. Lindquist et al. [17] demonstrates how mobile phones can be used to "broaden and enhance the use of active learning in large classrooms."

V. An Assessment of The Contributions

The goal of this position paper is achieved through the comparison of the support provided by proprietary software to active learning in Computer Science to the support offered by open source software. For this reason, the following 6 characteristics of active learning are derived from the literature review

a. Students applying material to "real life" situations: The scope of open source real life situations for Computer Science students is much larger than that of proprietary software. In the latter case, an internship or a project at an organization requires that this organization employs the same tools and software as those used at school; otherwise a proprietary training is needed. In the case of open source, both parties, the organization and the trainee, are not bound to any specific software. In case of any restriction, training and related materials are usually free and available through many communities.

b. Relating what is learned to past experiences: The concept is so similar to software reuse, which is possible in the realm of open source at a much wider scale than that of proprietary. In open source software, reuse can be at all levels, from operating systems to interfaces, which does not hold true in the case of proprietary.

c. Visual/intuitive learning: Both proprietary and open source software can offer the same learning. There is not enough literature to support the superiority of any category over the other.

d. Collaborative/cooperative learning: The community nature of open source gives it advantage in the issue of collaboration and cooperation. As mentioned before, proprietary companies form their own communities, but that lack the level of freedom and wide coverage existing in the open source communities. Project/ problem-based learning: There is no literature that gives advantage to any category over the other in problem-based learning. As for project learning, the facts verify that more initiatives and projects can be initiated in open source. A clear example is the Linux operating system.

e. Role playing: The level of freedom and independence provided by open source software gives it an advantage over proprietary software in the case of role playing. The possibilities of accessing the source codes, studying the designs, and the option to apply operations like reverse engineering, allow the learners to play all the roles played in the life cycle of software.

VI Conclusion

In closing, open source software backed by communities is clearly a better supporter of active learning than proprietary software backed by companies. By accepting that active learning is an effective model to learn Computer Science, it can be concluded that open source software provides more opportunities to learn computer Science, than what proprietary software provides.

References

[1] D. H Schunk. Learning theories: an educational perspective. Boston: Pearson, 2012, p. 8.

[2] O. Hazzan, T. Lapidot, and N. Ragonis, Guide to Teaching Computer Science: An Activity-Based Approach, Springer-Verlag London Limited, 2011, pp. 35-45.

[3] Cisco. The Learning Society. San Jose: Cisco, 2010, http://www.innovationunit.org/sites/default/files/The%20Learning %20Society%20White%20Paper.pdf, retrieved on 22/5/2014.

[4] Apple Inc.Apple and Education. http://www.apple.com/education retrieved on 22/05/2014.

[5] Microsoft Inc. Microsoft in Education. http://www.microsoft.com /education/ww/products/Pages/Products.aspx. Retrieved on 22/05/2014.

[6] D. Lipsa and R. Laramee, "Open Source Software in Computer Science and IT Higher Education: A Case Study", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.1, January 2011.

[7] A. Meiszner, "Learning the Open Source Way: FLOSS as a learning environment", OpenLearn 2007 conference, Milton Keynes, UK, 2007.

[8] R. Bosch and S. Vila-Marta, "Current Trends in Free Software Research", Research report LSI-08-36-R. 2009, retrieved from http://www.lsi.upc.edu/dept/techreps on 22/5/2014.

[9] Morelli, A. Tucker, N. Danner, T. de Lanerolle, H. Ellis, O. Izmirli, D. Krizanc, and D. Parker, "Revitalizing Computing Education Through Free and Open Source Software for Humanity", CACM, 52, 8, (2009). Retrieved from http://doi.acm.org/10.1145/ 1536616.1536635 on 22/05/2014.[doi:10.1145/1536616.1536635]

[10] A. Cerone, and S. Sowe, "Using Free/Libre Open Source Software Projects as E-learning Tools", Proceedings of the Fourth International Workshop on Foundations and Techniques for Open Source Software Certification (OpenCert 2010), Electronic Communications of the EASST, Volume 33, 2010.

[11] S. Sahami, A. Aiken, and J. Zelenski, "Expanding the Frontiers of Computer Science: Designing a Curriculum to Reflect a Diverse Field", SIGCSE'10, Milwaukee, Wisconsin, USA, March 10–13, 2010, retrieved from http://portal.acm.org on 10/03/2014. [doi=10.1.1.157.8431]

[12] D. Schweitzer and W. Brown. Interactive visualization for the active learning classroom. ACM SIGCSE Bulletin, 39(1), 2007, pp. 208–212. [doi=10.1.1.108.247]

[13] D. Paulson and J. Faust. Active learning for the college classroom. 2002. Retrieved from http://chemistry.calstatela.edu/Chem&Bioch/active/ main.htm on 22/05/2014.

[14] A. Chickering and Z. Gamson. Seven principles for good practice. American Association Higher Education Bulletin, 39, 1987, pp. 3–7.

[15] T. Briggs. Techniques for active learning in CS courses. Journal of Computing Sciences in College, 21(2), 2005, pp. 156–165.

[16] D. Lindquist, T. Denning, M. Kelly, R. Malani, W. Griswold, and B. Simon. Exploring the potential for mobile phones for active learning in the classroom. Proceedings of the 38th SIGCSE Technical Symposium on Computer SCIENCE Education, 2007, pp. 384–388. [doi:10.1145/1227310.1227445]