4-14-2014

MOTIVATING BUSINESS MAJOR STUDENTS TO LEARN COMPUTER PROGRAMMING – A CASE STUDY

Thomas Ngo-Ye
Dalton State College, tngoye@daltonstate.edu

Sung-Hee Park
Dalton State College, shpark@daltonstate.edu

MOTIVATING BUSINESS MAJOR STUDENTS TO LEARN COMPUTER PROGRAMMING – A CASE STUDY

Thomas Ngo-Ye
Dalton State College
tngoye@daltonstate.edu

Sung-Hee "Sunny" Park
Dalton State College
shpark@daltonstate.edu

ABSTRACT

Learning to program is viewed as difficult by many students. How to motivate and engage business major students to learn programming is challenging. In this paper, we report our attempt to teach business students programming. In our teaching method, first we orient students with the business and managerial aspect of programming by posing six original questions. By discussing these six questions, students gain appreciation of the high level responsibility of participating, contributing, assessing, and managing Information Systems (IS) as business professionals. Second, we adopt a Let Us Do It Together approach to deliver hands on labs to teach the technical aspect of programming. Inspired by the constructivism learning theory and the learning by doing and experimentation idea, our Let Us Do It Together approach mitigates students' anxiety and fear of programming. Overall, our teaching approach seems to enhance students' interest in programming.

Keywords

Learning programming, business and managerial aspect, Let Us Do It Together approach, constructivism learning theory

INTRODUCTION

Learning to write computer program has long been recognized as challenging to students, especially to novices who are introduced to programming for the first time (Blayney, 2009; Ramalingam, LaBelle and Wiedenbeck, 2004; Robins, Rountree and Rountree, 2003). In some undergraduate business programs, programming is included as a small component in a larger Principle of Management Information Systems (PMIS) course (Blayney, 2009). Given that the main objective of the course is to give students an overview of a broad spectrum of concepts and topics in MIS, there is very limited time available for teaching programming. In this paper, we report how we tackle this challenging issue of teaching undergraduate business students programming in just a few class meetings.

We organize the paper in the following manner. First, we briefly describe the background of this study. Next, we report our attempt in delivering the best possible learning experience and knowledge. We highlight two parts of our endeavor. In the first part, we focus on the business and managerial aspect of programming, which is centered around six original questions. We conduct extensive online research and then share our findings and perspectives with students. This activity is intended to reinforce students' understanding of the high level responsibility of participating, contributing, assessing, and managing Information Systems (IS) projects as business professionals (Kroenke, 2014). In the second part, guided by the constructivism learning theory (Liu, 2010), we adopt "Let Us Do It Together" approach to deliver hands on labs to teach students programming. Through learning by doing and experimentation (Neely and Pray, 2007), it reduces students' anxiety and enhance their self-confidence in programming. Finally, we conclude the paper by discussing the observed outcome and proposing future research direction of teaching programming.

BACKGROUND

In our program the PMIS course is required for all business majors. In particular, one of the student learning outcomes of our course is "demonstrate an understanding of the basic constructs of computer programming". Our PMIS is consistent with the "Core course: IS 2010.1 Foundations of Information Systems" (Topi, et al., 2010) recommended by ACM and AIS. It has been noticed that MIS curriculum is normally designed for MIS major students and the special needs of non-MIS majors are not well attended (He and Guo, 2011). In our PMIS classes, most students are non-MIS majors. Therefore, we have to tailor the course particularly for non-technical students.

In addition to the challenge (learning to program is difficult) mentioned in the literature (Blayney, 2009; Ramalingam, LaBelle and Wiedenbeck, 2004; Robins, Rountree and Rountree, 2003), we have a very tight time constraint – that we can devote at most four class meetings to teach programming. Each class meeting has 75 minutes. Moreover, in general students taking PMIS course are from diverse majors, with different existing skill levels, motivations and background (Neely and Pray, 2007). In our particular case, we have both some traditional college age students and a significant number of adult returning students. The adult students often have full-time or part-time job and family responsibilities. Taking together we encounter the extra challenge of teaching programming to students with diverse levels of readiness, motivation, interest, and competency.

Inspired by “The Last Lecture” story (Pausch and Zaslow, 2008), we raised the following practical question: “if this is our last lecture, what should we do to provide the most valuable information on introduction to programming to most students?” In fact, most of the students in our PMIS classes will never take another MIS course in the future. Therefore, technically they are attending the last lecture given by us, MIS faculties. Randy Pausch, a computer science professor at Carnegie Mellon, emphasized on the importance of helping students overcoming obstacles in his last lecture. In the same spirit, we pay special attention to help students overcome their fear of computer programming and boost their self-confidence. Thinking in the empathetic way, we imagine that we are the novice, none technical oriented students with busy schedule and limited time and attention (Kroenke, 2014). Then it is natural to arrive at our following teaching approach that aims to maximize the learning outcome and experience in a very limited time frame.

One practical question of teaching non-Information Technology (IT) business students programming is what programming language/tool should be used. Different programming languages and environments have been reported to be used in teaching programming (Blayney, 2009). Based on our antecedent experiences as well as the case presented in (Blayney, 2009), Microsoft Visual Basic for Applications (VBA)/Macro is selected as the language and environment for teaching programming. We articulate the following reasons that we choose VBA. First, we recognize that VBA is a very practical skill. VBA is very useful for certain types of Office automation tasks. Thus, business student can directly transfer what they learn of VBA from classroom to work. It is very applicable knowledge. Other types of computer programming language such as ALICE or similar pseudo programming language, although interesting in the sense of pedagogy, they are not practical for the real world. Second, VBA is easy to learn, easy to understand, and easy to use. VBA’s hierarchical object structure has elements such as document, paragraph, spreadsheet, selection, etc., which fits end users’ cognitive model. Business users can relate to VBA hierarchical object model and are familiar with the concepts, terms, and context (Blayney, 2009). Moreover, users are already familiar with the Microsoft Office environment, which may make them feel comfortable with VBA. Third, VBA is ubiquitously available. Since VBA is embedded in Microsoft Office suite and Microsoft Office is virtually installed on almost all business PCs, VBA is essentially available for most business professionals. Users can invoke VBA and enters VBA coding environment by simply click “View” tab and then click “Macros” button. Users do not need to download and install anything to run or code in VBA.

BUSINESS AND MANAGERIAL ASPECT OF PROGRAMMING

In an empirical study of what IT skills are expected from non-IT major business students, based on interviews with business recruiters, programming is ranked as No. 10 in the IT skills list (He and Guo, 2011). It is understandable that programming in a narrowly speaking sense (coding) is not the top IT skills sought after from non-IT major business students. After all, the hard-core coding tasks require substantial amount of programming theory, knowledge, and skills. They should be left to professional programmers, not casual business professionals. However, nowadays IT plays a critical role in the business operation and can contribute to organization objectives by either reducing cost or enhancing competitive advantage. Non-IT business professionals need to get involved, contribute, assess, and sometimes manage IT projects (Kendall and Kendall, 2011; Kroenke, 2014). Programming is an important component of IS (Topi, et al., 2010). Therefore, having a basic understanding and working knowledge of programming will help business professionals in many important fronts in their career.

Hence, in addition to teach students the hard-core coding aspect of programming, we intentionally guide students’ thinking from the business and managerial perspectives (ExcelForManagers.com, 2012), which might suit the long-term interests of most business students. We devote the first class meeting of programming to this business and managerial aspect. It sets the tone of introduction to programming and motivates students to learn programming in the following three more class meetings.

We create an Individual Report on Computer Programming assignment. The purpose of this individual assignment is to provide students the opportunity to better understand technical as well as business and managerial perspective of developing, planning, and managing IS. In this assignment, we pose six original questions that we believe that they best capture the elements of evaluating, assessing, applying, and managing IS in the context of programming. We provide students the materials about “Programming” collected by us. Students are also encouraged do additional online research on the topic. We post the assignment of six questions at the beginning of the semester. At the end of the semester, in our first class meeting of programming, we discuss these six questions with students and share with them all we have got so far to our best knowledge. In the following sub-section, we present the six original questions and our perspectives, which are written down when preparing the class meeting for business and managerial perspective of programming.

Individual Report on Computer Programming Assignment

1. Why business school students and non-technical knowledge workers need to learn Microsoft Visual Basic for Applications (VBA)/Macro?

Having some basic understanding of computer programming will be very helpful for a business professional or student to appreciate the work of IS professionals, especially programmers. This understanding will also better prepare business students to collaborate with programmers.

The late Apple CEO Steve Jobs said: "Everybody in this country should learn how to program a computer, should learn a computer language, because it teaches you how to think. It's like going to law school. I don't think anybody should be a lawyer, but going to law school can actually be useful because it teaches you how to think in a certain way[...] I view computer science as a liberal art" (Rosoff, 2011). It seems that there is a good reason that everybody should learn programming. At the fundamental level, programming may help one to become a better logic thinker.

In 1993, based on the disparate macro languages included with Word and Excel, Microsoft integrated them to create Visual Basic for Applications (VBA). Over the years, Microsoft ensures all Office applications support VBA. With VBA, programmers get access to each Office application's object model and develop extended Office applications. The extensibility model greatly contributes to the phenomenal success of Microsoft Office product (Anderson, 2013). VBA has its root in "classic" Visual Basic 6 (Nicholas, 2013; Zlatkovsky, 2013). In both Word and Excel, there is a macro recorder. VBA offers an in-product experience for automating Office task to get the job done. It is often an ideal choice for Office application development for internal/departmental employees (Zlatkovsky, 2013).

Microsoft VBA/Macro is arguably an easy and practical way to introduce business students the concepts of computer programming. With the knowledge of VBA, non-technical business students can use it to do some quick programming jobs to automate work tasks and also make their work better.

2. How VBA/Macro can help business organizations and individuals in their work?

Microsoft VBA/Macro is usually used for solving existing needs for existing users (Zlatkovsky, 2013). It is often used to automate routine repeated tasks (Anderson, 2013). It will save a lot of time and tedious labor intensive effort. With the freed time, employees can devote their energy and effort to other more productive and creative work activities. Moreover, VBA can help improve the interaction between users and computer applications such as Microsoft Office 2013 (Zlatkovsky, 2013). Using VBA, users can design customized dialog screens to help end users to bridge different applications such as Microsoft Outlook and Excel, making them as an integrated whole.

3. What are the Pros and Cons of User Developed Applications with VBA/Macro?

User developed applications with Microsoft VBA have several advantages. Users tend to have better knowledge and understanding of user requirements – what users really need and want (Kendall and Kendall, 2011; Kroenke, 2014). No communication with the IS staffs is needed, as users are the owners and creators of their own VBA applications. With this shortcut development approach, less communication and people related problems will occur, because fewer parties are involved. Moreover, users usually have strong motivation to make their own application work and thus more likely to adopt the application that they spent time on to develop. More satisfaction toward the application may be derived. Users will also maintain and support their own VBA application, with better feedback and less lead time for fixing problems and improving the application. Thus, user developed applications with VBA avoid many common problems associated with regular IS developed by professional IS staffs.

However, at the same time, user developed applications with VBA do have some limitations. First, users need to have certain level of knowledge and skills of Microsoft VBA to perform the development work. Not all the organizations have qualified employees who can do the VBA programming on their own, without the support from the professional IS staffs. One of the most serious shortcomings of user developed VBA embedded in office applications is the lack of thorough testing and big potential for error.

Another serious concern is related to the management of the scalability, compatibility, and security of the VBA applications. Since VBA code is often saved and distributed directly in a Microsoft Office document, many users will have a copy of the VBA code. This approach renders initial deployment and distribution very easy. However, for the same reason, it will create a huge challenge when the VBA code needs to be updated, added or deleted (Zlatkovsky, 2013). There is a risk that different versions of VBA applications exist in the same organization to address the same business problem. This will cause confusion among end user employees. Incompatible data and the difficulty to sort out which VBA application version represents the most current and correct information are very challenging. Security of VBA application is another big problem for organizations. VBA applications need to be carefully managed to ensure the clean deployment of the consistent application.

Moreover, the knowledge and business procedure/process/policy embedded in the VBA application codes should be shared among users. Another potential problem of user developed application with VBA is related to the ongoing maintenance and support of VBA applications. The original VBA developers know everything about the VBA applications. However, when deploying the written VBA applications to other employees (end users) in the organization, the end users may have some questions and need support and help. VBA developers may not have the time to help or support end users, because it is only part of the unofficial job duty. Documentation of VBA applications and training end users are critical to the success of the application usage, especial in cases that users are not the original VBA developers. Original VBA developers usually do not invest enough time and effort to produce professional quality documentation. When the VBA developers leave the organization, it will be very hard to figure out and maintain the VBA applications without existing proper documentation.

4. Is the skill of VBA/Macro still relevant?

Definitely Yes! VBA technology is still alive as of today (Anderson, 2013). "Microsoft Visual Basic for Applications (VBA) is still supported for individual Office 2013 application development."(Source: http://msdn.microsoft.com/en-us/library/office/jj229830.aspx). Microsoft for its own interest keeps supporting VBA. In both Windows version of Office 2013 and Mac version of Office 2011, they have the built-in VBA environment. In the foreseeable future, Microsoft will continue bundling VBA environment in its Office suite products and support VBA (Zlatkovsky, 2013).

The main reason for Microsoft to do so is that many organizations invested a lot of labor, time, and money to develop and maintain their home-grown VBA applications to extend the standard Office applications' capability to run their own businesses. Those VBA applications are an important part of their business IT infrastructure and play a critical role for the daily operation of the organization. If Microsoft withdraws its support of VBA, those organizations with heavy investment in VBA will face a very tough dilemma. Organizations can choose to phase out all their VBA applications and gradually convert them into newer Microsoft Dot Net based technology – Visual Studio Tools for Office (VSTO) or the cutting edge technology – Apps for Office, which is only available starting in Microsoft Office 2013. But this strategy will be extremely expensive and time consuming to implement, as thousands of lines of VBA codes need to be converted. Moreover, the process will also be enormously difficult, as it represents a paradigm shift (Zlatkovsky, 2013). Another choice is for the organizations to stay with their current version of Microsoft Office product, such as Office 2013, and not to upgrade to the newer version of Office product in the future. In this way, the organizations can continue using their existing VBA applications, but giving up the potential benefits of new features coming along with the newer version of Office product. To mitigate business clients' legitimate concern of VBA viability, Microsoft continues supporting VBA and recommends clients to carry on VBA applications if they are satisfied with the capabilities and tooling (Zlatkovsky, 2013). Moreover, VBA has the clear advantage that it does not involve the heavy overhead of learning VSTO and new languages such as HTML5, CSS3, and JavaScript for developing Apps for Office.

Although Microsoft offers newer technologies such as VSTO and Apps for Office, to develop Microsoft Office applications for Mac computers, VBA is still the only choice. Given the Apple's market share of PC, the cross-platform support (having an Office application working in both Windows and Mac environment) is still relevant and important (Nicholas, 2013).

Recently, Office 365 becomes an important product for Microsoft to complete in the cloud computing era (Kroenke, 2014). When Microsoft Office 365 is used as a cloud-based storage facility, users can upload an Office document containing VBA code. Users can open the document in desktop version of Microsoft Office and VBA code will still work (Das, Mistouflet and Rohn007, 2012).

From various websites, it seems that there is a reasonably sizable VBA programmer community (Blayney, 2009). Some of the VBA programmers are IT consultants, capitalizing their VBA skills by helping their business clients for solving their business problems and designing customized VBA applications. Many of them seem to make a comfortable living by doing VBA programming.

From a non-technical business professional's perspective, learning to write computer program or VBA in particular, not only improves one's future job prospect, but also enhances one's current standing among peers and supervisor. Knowing how to do VBA programming also makes the business professional's Office work more efficient and effective. Moreover, the fundamental programming concepts and even some syntax of VBA are consistent and relevant to Microsoft's new Dot Net technology. Moreover, in many organizations, VBA is used to customize and create reports from databases. Even the coding behind the "Crystal Report" has very similar syntax to VBA. Therefore, VBA skill is still relevant and useful today.

5. What factors will help a non-technical user to learn to program in VBA/Macro?

First, as pointed out in the online discussion forum, if a person is tasked with a project to use VBA to develop a user application for work, it will give the person a very strong motivation to learn VBA (Supdawg, 2008). Reading VBA books and practice VBA coding examples will be helpful. If one happens to have a mentor who can help and guide VBA development, it will be even better. In many cases, one can turn to online VBA community to seek help and support (ExcelForManagers.com, 2012). Like many other professional activities, it takes time and practice to learn and master VBA programming. However, it is doable, because VBA is designed for non-technical people and non-programmers from scratch.

6. From information systems development methodology perspective, what role does project plan play in ensuring the success of a VBA/Macro project?

Project planning is also important in VBA project. Just because end users develop VBA applications and professional IT staffs are not involved, it does not mean that the VBA project can be pursued carelessly. Users should follow the same IS development methodology such as SLDC to ensure the success of the VBA project. Users need to define the goals of VBA project and evaluate other alternative technologies and approaches. Feasibilities of VBA project should also be investigated before a full blown project is underway. Requirement analysis, communications with relevant business users, creating small prototype sample applications, and seeking users' honest feedback are all important just as in other IS development projects. VBA project may also take several iterations and corrections to finally meet users' requirements. It is never enough to emphasize the importance of proper project management and planning in VBA project (ExcelForManagers.com, 2012).

LET US DO IT TOGETHER APPROACH – HANDS ON LABS

In this section, we report our approach of teaching the technical aspect of programming in the remaining three class meetings. For the new generation of students who grow up with computer and Internet, it seems their use of technology is natural. But the appearance is deceiving (Lavy and Or-Bach, 2011). While the young generation is comfortable with information technology such as smart phone, they do not necessarily have technology proficiency, especially with academic programming tools such as VBA. Given the diverse background of student population in terms of computer competency and motivation, it is challenging to teach VBA in a way to provide most students a great learning experience with excitement. The limitations of traditional teaching method include perceived boringness of passive lecture and lack of social learning and interaction with the instructor. We observe that, if without intervention, students with low self-motivation withdraw their effort.

Let Us Do It Together approach is inspired by the constructivism learning theory, which recommends students actively engage in the learning and thinking process, and construct knowledge (Liu, 2010). From students' own learning experience, they construct a change in meaning (Newby, Stepich and Russell, 1996). The core idea is that knowledge cannot be simply transmitted from teacher to students through lectures, but via an active process of construction (Gonzalez, 2004). Learning by doing and experimentation is especially effective in teaching technology (Neely and Pray, 2007). The constructivism learning theory has been successfully applied in teaching Office Application software (Haruehansawasin and Kiattikomol, 2011), where teachers act as facilitators to enable students actively engage in their learning and offer help when students struggle.

To mitigate and address the challenge of students' anxiety and fear of programming, we explore and experiment a new teaching approach for programming. We take a very active and engaging approach, where we create a "Let us do it together" atmosphere in class by performing hands-on coding activity ourselves. Students watch and follow us on their computers step-by-step. Students see our actions, learn from our mistakes, and get active learning and real time explanation in the exact context. During the live demo, we pause, in every a few minutes, to walk around the classroom and check on students' progress in the activity. We also incorporate the lecture in the context or process of demo, to make it natural and integrated. This method is more like a tutoring, rather than mass lecture. Students feel the sincere effort from the instructor. During the process, students and the instructor worked together to overcome the unexpected issue. The interaction is the very source of learning, improvement, deeper understanding and creativity.

CONCLUSION

Based on the informal feedbacks from the students, we find that the hands-on approach is an effective means to achieve the learning goal of the technical aspect of programming. Many students express increased interests in learning programming and plan to explore more. Some students even prefer us devote more class time on VBA and assign more challenging coding project. Businesses that have employees with VBA skills can improve productivity, effectiveness, and efficiency of work. Through this study, we communicate and convince business students to invest time and effort to learn VBA. We provide essential introduction information of VBA to students. Students seem to change their belief about VBA and increase intention to learn VBA. In the future, we will further investigate 1) whether business students/professionals intend to continue learning programming and apply programming skills to work and 2) how Let Us Do It Together approach effects on their intentions. TAM may be a fruitful perspective in this line of inquiry.

REFERENCES

1. Anderson, T. (2013, March 4) *Office 2013 is for sale; Office App Store already dead; COM Add-ins alive and well.* Retrieved December 30, 2013, from http://www.add-in-express.com: http://www.add-in-express.com/creating-addins-blog/2013/03/04/office-2013-for-sale-office-app-store-already-dead/
2. Blayney, P. J. (2009) Knowledge gap? Accounting practitioners lacking computer programming concepts as essential knowledge. In G. Siemens and C. Fulford (Ed.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 151-159). Chesapeake, VA: AACE.
3. Das, R. S., Mistouflet, S. and Rohn007. (2012, October 18) *Is there a VBA or equivalent in office 365.* Retrieved December 30, 2013, from http://answers.microsoft.com: http://answers.microsoft.com/en-us/office/forum/webapps-excel/is-there-a-vba-or-equivalent-in-office-365/0393b6f7-d018-42ef-aced-33b8dd617adb
4. ExcelForManagers.com. (2012, January) *VBA For Managers.* Retrieved December 30, 2013, from ExcelForManagers.com: http://ebookbrowsee.net/vba-for-managers-pdf-d43865912
5. Gonzalez, G. (2004) Constructivism in an introduction to programming course. *Journal of Computing Sciences in Colleges*, *19*(4), 299-303.
6. Haruehansawasin, S. and Kiattikomol, P. (2011) Enhancing vocational learners' problem-solving and knowledge-seeking skills in effiently applying Office application software (A learning model evaluation). *ICER 2011: Learning Communities for Sustainable Development* (pp. 457-466). KKU, Thailand: ICER.
7. He, J. and Guo, Y. M. (2011) Should I take MISXXX? Implications from interviews with business recruiters. *Proceedings of the Seventeenth Conference on Information Systems* (pp. 1-9). Detroit, Michigan: Association for Information Systems.
8. Kendall, K. E. and Kendall, J. E. (2011) *Systems Analysis and Design* (8th ed.). Upper Saddle River, New Jersey, U.S.A.: Prentice Hall.
9. Kroenke, D. (2014) *Experiencing MIS* (4th ed.). Upper Saddle River, New Jersey, U.S.A.: Prentice Hall.
10. Lavy, I. and Or-Bach, R. (2011, June) ICT literacy education - College students' retrospective perceptions. *ACM Inroads, 2*(2), 67-76.
11. Liu, C. C. (2010) Evolution of constructivism. *Contemporay Issues in Education, 3*(4), 65.
12. Neely, M. P. and Pray, T. F. (2007, Spring/Summer) The case for a more rigorous approach to teaching spreadsheet and database applications. *Journal of Informatics Education Research, 9*(1), 121-140.
13. Newby, T. J., Stepich, D. A. and Russell, J. D. (1996) *Instructional technology for teaching and learning: Designing instruction, integrating computers, and using media.* NJ: Prentice Hall.
14. Nicholas. (2013, June 19) *The Future of Microsoft Office Add-In Development.* Retrieved December 30, 2013, from In the Flow BreezeTree Software: http://www.breezetree.com/blog/index.php/future-of-office-add-in-development/
15. Pausch, R. and Zaslow, J. (2008) *The Last Lecture* (1st ed.). Hyperion.
16. Ramalingam, V., LaBelle, D. and Wiedenbeck, S. (2004, September) Self-efficacy and mental models in learning to program. *Association for Computing Machinery SIGCSE Bulletin, 36*(3), 171-175.
17. Robins, A., Rountree, J. and Rountree, N. (2003) Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172.
18. Rosoff, M. (2011, November 11) *The most interesting things Steve Jobs said in a "Lost" interview showing next week.* Retrieved January 2, 2014, from http://www.businessinsider.com: http://www.businessinsider.com/the-best-quotes-from-the-lost-steve-jobs-interview-showing-this-weekend-2011-11
19. Supdawg. (2008, March 3) *How long does it take to properly learn VBA?* Retrieved January 2, 2014, from http://www.mrexcel.com/: http://www.mrexcel.com/forum/showthread.php?t=306705
20. Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K., Nunamaker, J. F., Sipior, J. C., et al. (2010, April) IS 2010: Curriculum guidelines for undergraduate degree programs in information systems. *Communications of the Association for Information Systems, 26*, 359-428.
21. Zlatkovsky, M. (2013, June 18) *Roadmap for Apps for Office, VSTO, and VBA.* Retrieved December 30, 2013, from http://blogs.msdn.com: http://blogs.msdn.com/b/officeapps/archive/2013/06/18/roadmap-for-apps-for-office-vsto-and-vba.aspx