On the Integration of Agile Practices into Teaching: *an approach to overcoming teaching and learning challenges of programming*

Bassey Isong, Ohaeri Ifeoma and Naison Gasela
Department of Computer Sciences
North-West University
Mafikeng Campus, South Africa
bassey.isong@nwu.ac.za, oh.ifeoma@yahoo.com, naison.gasela@nwu.ac.za

***Abstract*—In Computer Science Education, teaching of computer programming (CP) for beginners remains one of the greatest challenges. This is due to the nature of first year students characterized by under-preparedness, poor academic background and so on. Moreover, the traditional lecture-based approach is used and the computer laboratories are rarely used. Consequently, students are not motivated, engaged or no active participation. While this teaching approach has failed to achieve learning outcome, incorporating cutting-edge practices like agile methods into the teaching could be valuable. This stems from the widespread in practice and acceptance of agile methods in the software industries today. Though agile practices like pair programming have impacted students' programming projects, none has been reported on teaching. Therefore, this paper proposes a new approach of teaching CP using the agile methods. The goal is to assist beginners acquire needed programming skills, proffer teaching technique that maximizes the chances of engagement, active participation, and improving teaching by reflecting on what has been taught and what the students are learning. Additionally, beginners will be able to operate the computer, write correct code, gain better programming skills and assist teachers to manage large classes effectively. To achieve this, the study adopted the framework for decoding the discipline**.

Keywords— agile process, computer programming, learning, teaching, undergraduate students.

I. INTRODUCTION

In the educational world today, increasing students' diverse background, under-preparedness, language competences, large classes, numerous courses, poor attendance, poor teaching methods etc. are some of the intense challenges of teaching and learning that confronts teachers. These challenges emanates from efforts to achieve greater importance and full participation [1]. In years back when the university classes were limited to highly selected students, the traditional lecture-based teaching backed by tutorials was considered good enough. But today, as a result of diversity, levels of under-preparedness and so on, many students finds it difficult to survive with the system [2]. While this is common among several universities in developing nations, teachers have to play the role of assisting these students to achieve their learning outcomes. Teachers should not see the challenge as an impediment to teaching such students but to change their usual way of learning [2]. To this end, a more radical approach of teaching is needed to support students to achieve their learning outcomes or simply a teaching technique that maximizes the chances of engaging the students in their learning processes [2].

Today, several different teaching methods have been developed and utilized in the context of teaching and learning where information and communication technology (ICT) based approach is one of such that has been known for it improved efficiency [3]. ICT has the capacity to improve students' learning both inside and outside the classroom. It increases students motivation, deepens their understanding as well as improving their memory retention, etc. [4]. Furthermore, ICTs promotes learning collaboration, role playing and group problem solving activities [5]. Nevertheless, due to the rapid technological advancement witnessed today, the educational methods and practices has to evolve also since is now challenging to teach as a result increasing students' diversity [6]. In this case, a new technology-oriented approach is needed to cope with the diverse and dynamic needs of students [6][7].

One area where students' diversity and under-preparedness challenges has affected teachers immensely is the teaching of introductory computer programming (ICP) course for first year Computer Science (CS) students. The teaching of ICP poses a serious challenge to CS education [8] and a new technological improved approach is required to add to the teaching method. CP is an indispensable part of computing and is considered a useful skill which constitutes one of the core competences expected of every CS graduate [9]. It is a problem-solving process of formulating, planning the solution, designing the solution, translation, testing, and delivery [10].Moreover, to accomplish these processes, a programmers has to be equipped with skills such as learning the language, composing new programs, comprehending, reusing and integrating existing programs, modifying, and so on [10]. Nonetheless, CP has been considered difficult and challenging tasks involving huge cognitive activities [11].

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.153

Learning to program is known to be difficult due to the fact that the learning process is vulnerable to several risks [8]. One of such risks is the teaching approach that is used to deliver the instructions to the students. The existing CP teaching approach is the traditional lecture-based that has fixed instructional design in spite of the dynamic nature of computer technology and curriculum [12]. As a result, several students are not engaged, can't even use the computer or understand the working of software code, their syntax as well as how to code themselves. Given the newcomers of the CS education, the beginners constitutes about 80% of students who have just left the high school with little or no knowledge about computers. Therefore, the traditional lecture-based approach is not appropriate for teaching such students CP. This is because the teacher is seen as the primary source of knowledge where the leaning is described as information transmission [1][13]. Additionally, students are lectured only in the classrooms and the computer laboratories are hardly used. With their under-prepared, teaching CP only in the classroom does not promote active students participation and engagement. Learning outcome is hardly achieved irrespective of the teaching technique used.

CS teachers of ICP should not see teaching as just the transmission of knowledge in the form of information, rather it should be seen as enabling participation in knowing [1]. This is because "*access to knowledge can only be gained by enabling participating in the knowledge community*" [1]. Thus, a teaching method is required for ICP course that enables participation such that students will be able to gain the required knowledge or skills and core competences in their learning process as CS students. To provide a comfortable atmosphere that enable students get engaged and actively participate freely in ICP course, ICP teachers have to align their teaching with advanced ICT cutting-edge practices and processes. This paper therefore, proposes a new approach of teaching first year CS students ICP via the agile software development practices and processes using the decoding the discipline framework by [14]. The objective is to help students initiate into CP in the CS discipline through active participation and engagement in the course activities.

The rest of the paper is organized as follows: Section II introduces agile software development process, III is the proposed teaching approach, IV is the paper discussions and V is the conclusions.

II. Agile Methodology

Agile software development process involving several "*lightweight*" methods was founded in the 1990s. It emanated as part of the stands against the traditional software processes known as the "*heavyweight*" methods due to their rigid, slow, and inconsistent nature of software development [15][16]. Agile methods are specifically designed to develop software in short iterations, increase productivity, involve customers, high quality software, and development within time and budget [17]. Agile benefits ranges from effective communication to increased customer satisfaction [15]. Moreover, it is guided by agile manifesto that was articulated in 2001 as a declaration of agile practices and principles. The agile manifesto is as follows [18]:

i. Individuals and interactions over processes and tools
ii. Working software over comprehensive documentation
iii. Customer collaboration over contract negotiation
iv. Responding to change over following a plan

For more information on agile practices and principles, refer to [17]. Agile methods and principles has been extensively accepted in the software industry [19][20][21]. This is evident in several software development companies today. It is on this ground that this study advocate for agile practices to be integrated into the teaching of CP. Since agile focuses on work life balance and interactions among team members, integrating its practices into the teaching could go a long way to assist first year CS students to achieve their learning outcome in both technical and social skills [22]. In recent years, agile principles have been reported in lectures, academic conferences, journals including its successful application either in student's projects or practical. For example, the use of agile practices such as pair programming in students' software development projects has been commendable [7][22][23][24][25][26]. In pair programming, codes are programmed in pairs of developers to ensure the participation of everyone in all development tasks and the benefits ranges from collaborative learning to peer evaluation promotion [7][23][27]. However, nothing has been said about agile methods application in the teaching. Based on the successes of agile practices and processes, this paper aim is to extend agile values to the teaching of ICP to cope with the high expectations, under-preparedness and diversity.

III. Proposed Introductory Computer Programming Teaching Approach

In this section, discussion is based on the new ICP teaching approach that is intended to increase the chances of engaging first year CS students and assist the teachers to reflect on their teaching as well as their students' learning. The idea is not to condemn the existing traditional lecture-based approach but rather to re-engineer it using cutting-edge practices that enables coping with diverse backgrounds and under-preparedness. Moreover, it is intended to boost CS programs enrolment, productivity of CS graduates and so on.

In order to achieve this, our study adopted the model tagged "*Decoding the Disciplines*" by [14]. The model consist of seven important steps as shown on Fig. 1. The steps are processes involved in getting students deeply initiated into the details of thinking and learning in their disciplines [14]. The aim is to develop strategy for increasing learning and assisting students master a specific material in their discipline [14]. In the context of this study, it is adopted by the teacher for beginner students to master CP in the CS discipline. The model constitutes an expert attempt to improve post-secondary education of CS. The steps involve are questions designed to

be asked by teachers during the course of planning solutions to the specific challenges that affects learning in a given discipline. This study therefore, followed the steps modelled by [14] to provide a solution to ICP course challenges teachers faced in institutions of higher learning. These steps are discussed as follows:

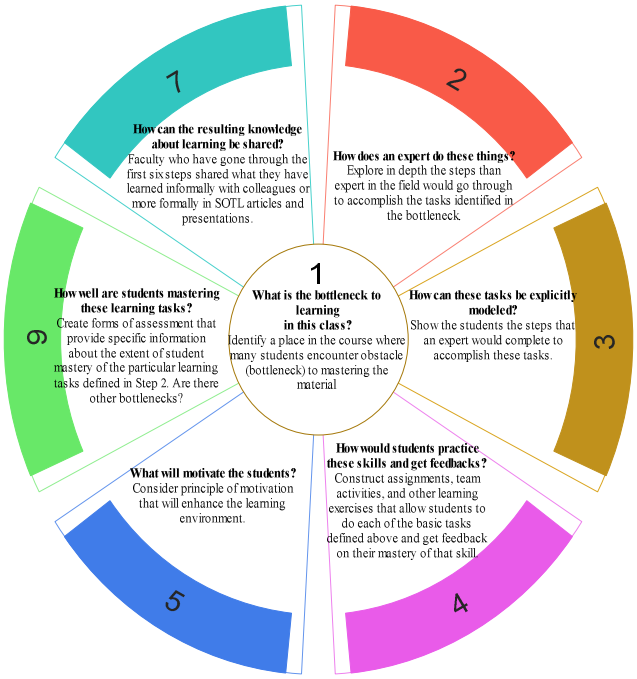

Fig. 1. Decoding the Disciplines: Seven Steps to Overcome Obstacles to Learning [14]

A. *Step 1: What Is a Bottleneck to Learning ICP in the First Year Class?*

This step starts by identifying the different bottlenecks or impediments to learning ICP course in the first year CS class. In this case, several challenges where identified that affects teaching and learning of ICP in most institutions of higher learning in developing nations. The challenges where known to impact both the teachers and students negatively and they include computer literacy, lack of access to essential resources, language competences, and ineffective teaching methodology. While some of the challenges are easily dealt with in developed countries, developing nations are struggling to contain them. These challenges are discussed:

1) Computer literacy: In most university system, apart from the remedial students, majority of the students, about 80% in the first year CS class are directly from secondary schools where CP was never taught. Consequently, some have not seen, touched or even operated computers before. They only touched computers for the first time in the programming class. This makes it difficult if not impossible for the students to operate the computer, understand programming concepts and structures, syntax and semantics of the language, the programming environment and actively participate in the CP classes.

2) Lack of computer access and other resources: This challenge is rooted from the poor socio-economic background of the students such as financial constraints. In this case, most students can't afford university accommodation, unable to purchase personal computers or laptops and essential discipline text books. In addition, the departmental computer laboratory is not always available or get the required resources needed to learning programming effectively.

3) Language competence: Due to the foreign nature of English language, some of the students have poor English background and can't communicate or write fluently. It is always a challenging task initiating them into the discourse of the discipline and as a result, teachers are forced to choose words when teaching or try to mimic their mother tongue before the students can understand. Moreover, students cannot participate or engage actively in the classroom as they cannot communicate effectively either in writing or speaking.

4) Ineffective teaching methodology: Although the teaching and learning of CP has been tagged difficult [28][29], it is not always true all the time. The fact is that, lack of experienced teacher or the use of poor teaching method used in delivering the lectures could be the reason why students are not actively engaged and participate in the class. In this case, students learning process is adversely affected. Thus, a teaching approach that is not productive should be replaced by a productive one. This forms the central focus of this study.

B. *Step 2: How Does an Expert Solve This Nature of Challenge?*

To answer this question which is aimed at finding viable solutions to the challenges faced in the teaching and learning of ICP for the beginners, this study has proposed a new teaching approach that is based on the agile manifesto and the pair programming practices. The approach is intended to engage the students in the CP class as well as ensure active participation which is not the case with the traditional lecture-based. This teaching method is expected to be aligned to agile manifesto and pair programming practices in its operations. However, for this proposal to be effective and successful, several requirements need to be satisfied such as:

- There should be available and well-furnished computer-equipped classrooms which can accommodate at least fifty (50) or 25 pairs of students offering ICP, taking pair programming settings into account.
- All programming classes should not be held in ordinary classrooms rather than computer-equipped classroom to avoid separate practical class schedules.
- Lecture notes and programming materials should be given to students before the commencement of the course. This is important because the student need not write separate notes in their classes since they have to strictly follow the teacher's demonstration.

- The teacher should be ready to actively maintain close contact with the students and not only teaching. In this case, he will be able to reflect on what he is teaching and what the students are learning.
- ICP classes should be scheduled to last for at least two hours per class in order to cover in-depth what is required.

These requirements should be taking into consideration for the successful implementation of this proposal.

C. Step 3. How Can This Proposed Solution Be Explicitly Modelled?

In order to explicitly model the agile methodology in this regard, the values implanted in agile manifesto has to be aligned with the teaching approach of ICP for beginners in their first year CS class. This is shown in Fig. 2. The rationale is that aligning ICP teaching in this manner could help motivate the students, getting them to engage and participate in the class.

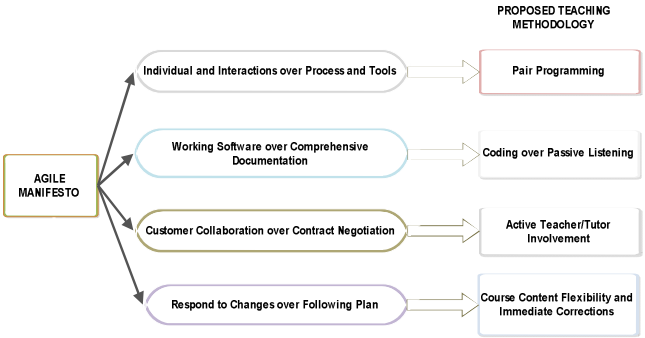

Fig. 2. Agile manifesto and ICP teaching alignment

Fig. 3. ICP class setup with pair programming

1) Pair Programming: To achieve "*individuals and interactions over processes and tools*", this study recommend pair programming practices as well as the use of one integrated development environments (IDE) for programming. Two students should be assigned to one permanent computer in the laboratory to be used in every CP class. The set-up is shown in Fig. 3. In this case, as one is actively involved in coding, the other is continuously observing the code to possible identify errors or provide more ideas to achieve a better solution to their task. This is idea is expected to be implemented interchangeably to maintain a balance and avoid one student being passive. Additionally, trust will be built through continuous interaction and collaboration. With the class set-up, students will be able to master programming skills through constant practices and use of a single IDE throughout the course.

2) Coding over passive listening: The agile manifesto, "*working software over comprehensive documentation*" can be accomplished by creating a comfortable atmosphere for students in pairs to directly and immediately practice whatever their teacher is doing on the whiteboard with their computer instead of just being passive listeners. Passive listening in this context refers to writing down what is said by the teacher as note which may be difficult to understand later during practice. If students practice by coding in this style, they will get to understand how the code operates on the IDE in terms of execution as well as memorize and retain the steps used in achieving the results. They will be able to master the coding, the language syntax, program execution and debugging where necessary. For instance, if a teacher says all variables in a program must be declared before being actually used, as in the code segment shown in Fig. 4.

```
.....
.....
public static void main(String[] args) {
  a = 1;
  b = 1;
  System.out.println(a);
  while (b < 50) {
    System.out.print(b);
    b = a + b; // new b
    a = b - a; // new a
  }
.....
```

Fig. 4. Sample code segment

The best way the students could understand the syntax and the statement is by running the sample program code with an undeclared variables to see if it compiles or not and be able to debug it where necessary. Consequently, this practice will be able to improve their understanding, master the coding and apply it whenever they are writing their own programs.

3) Customer collaborations over contract negotiation: In this proposed teaching approach, "*customer collaboration over contract negotiation*" as part of agile manifestos can be achieved by having the teacher available and in contact with the students throughout the ICP class. This involve meeting the students from time to time during the CP classes to see if they are doing the right thing or having some problems. During ICP class, since it is not possible to have external customer as done in agile, the teacher having sufficient domain and programming knowledge will assume the responsibility of the customer. Though *negotiation of contract* is not applicable here, the teacher has to give the students programming tasks and maintain frequent contact with the

students in the class to see that they are doing the right thing as well as assisting them when they get trapped. In addition, instead of submitting their solutions after performing their tasks, the teacher through active involvement will assess the students' work immediately where applicable based on what is displayed on the screen.

4) Course content flexibility and immediate corrections: Accordingly, in this proposed approach "*responding to change over following a plan"* can be achieved in two different ways in order as a way of ensuring students achieved their learning outcome. In one hand, the course contents should be adjustable to meet the needs of the students when the need arise. This is important, especially in the case of comprehension problems not rooted from the teacher's experience. The first year students being new to programming should be taught based on their level of understanding instead of following the course content serially. Adjustment should be made where necessary by following the Bloom's taxonomy [30]. On the other hand, there should be immediate correction of programming errors during ICP class. As teacher maintain constant close contact with the students, whenever they are found not doing the right thing, it should be pointed out and the changes effected immediately without having to wait for the task to be submitted for assessment before changes are made. When required changes are made at the right time, it will save the effort and time for rework on the part of students and the assessments on the part of the teacher.

D. Step 4. How Will Students Practice These Programming Skills and Get Feedback?

To answer this question, the practice of pair programming and direct coding as modelled in Fig. 3 are important to help the students master the skills of CP. As presented in Fig. 3, in the even programming tasks are given or a demonstration from the teacher, the students have to do it themselves in order to get the feelings of CP themselves rather than seeing it like the transmission of information from the teacher to the students. In this case, active participation through the course will be achieved.

E. Step 5. What Will Motivate the First Year CS Students?

In the context of the teaching approach in this study, what will motivate the CS student is when a teaching approach that offers the students opportunities for sustained participation in the first year class is used. Based on the mode of operation, aligning the teaching method to agile as shown in Fig. 2 will bring about continuous interactions, collaborations, mastering of programming skills and syntax and so on. Students are motivated as they are given opportunity to write and run code on their own and the teacher maintaining constant close contact on what they are doing as well as correcting the students when they go out of point. Thus, students and the teacher are seen as partners in the learning process.

F. Step 6. How Well Are the First Year CS Students Mastering These Programming Learning Tasks?

In this proposed approach, it is easy and simple to know if the students are mastering the programming skills or the use of the computer to solve problems. With the theory of *do it yourself*, the students constantly embark on exercise that offers them the chance to practice and obtain immediate answers. Perhaps, this provides the teacher who maintain constant close contact with the students a considerable amount of information about whether learning has occurred or not. In this case, teacher will give either a simple or complex programming tasks and would have to assess them on completion to see if they are doing the right thing or not and at the same time assist them in correcting their mistakes, where necessary. Based on this, the teacher would be able to tell if the students are mastering the programming skills or not.

G. Step 7. How Can the Resulting Knowledge About Learning in the CP Class Be Shared?

With the teaching approach discussed so far in this study and as part of the efforts to motivate the students, the teacher has to give the students real-world programming tasks in the form of a project using the pair programming practices. At the completion of such project, each pair of students will be given an opportunity to present their task and how it was achieved to the rest of the class members. In some cases depending on the nature of the task, the outcome could even be published in journals or presented in conferences. By this act, students will be motivated more to program on their own and be equipped with the advanced technical proficiency and core competences expected from CS graduates.

IV. DISCUSSIONS

Teaching and learning in institutions of higher learning is witnessing a shift in the paradigm that ICT usage has brought. However, issues of diverse background of students and their level of under-preparedness has rendered the existing ICT insufficient. This requires an improvement that is proactive to deal with the current situation. This paper has proposed and presented a new teaching approach for ICP course to first year CS students. The approach is based on the application of some cutting-edge ICT practices and processes like agile methods which could be productive in the enhancement of current teaching approach. By aligning agile principles and practices to the teaching of ICP course, the students will be able engaged and actively participate in ICP classes. The learning benefits ranges from being able to learn how to operate the compute to mastering programming skills through active team collaboration and interactions. For the teacher, this approach will go a long way to assisting them in managing large classes effectively by getting students engage and participate actively.

The justification for the incorporation of pair programming into the teaching method as a core practice in the agile process stems from the successes attributed to it either on students' projects or in the industries as reported in the literature. For instance, [23] applied pair programming as learning method in

several programming courses. In a similar study, Bipp et al [19] found that pair programmer students can produce more readable code that non-pair programmer. Furthermore, a study by [24] established that students who practiced pair programing have increased skill in individual programming activities and retain positive attitudes towards programming courses. The same findings was obtained by Perera [7]. Additionally, Perera [7] considered agile practices as a reliable method of helping weaker students to improve their programming skills. In the perspective of peer evaluation, Gatfield [27] emphasized that, as students assesses each other's work they become satisfied and their learning becomes effective. In general, pair programming has the capacity to improve novice programmers understanding.

With the few highlights on pair programming coupled with the relevance of the agile practices, the intuition of this study is that, integrating agile practices into the teaching of courses like ICP could assist in creating a learning environment that is problem-based which is effective in helping students to engage in the actual learning of CP. It could also go a long way to eliminate the challenges first year CS students faced during the course of learning CP.

V. CONCLUSIONS

Computer programming is a useful skill that constitutes one of the core competences required in CS Education. Yet, the teaching and learning poses a great challenge for both teachers and students respectively. This paper has presented a new approach of teaching CP courses to first year CS students using agile principles and practices of pair programming and by adopting the framework for decoding the discipline. Albeit CP is difficult coupled with the nature of first year CS students, this paper believed that by aligning the teaching approach to agile manifesto and pair programming practices, students can be motivated to achieve their learning outcome. The approach requires having all programming classes in the computer-equipped classrooms with students sitting in pairs. Conclusively, it could create an effective teaching framework that will bring about improved understanding, active engagement and participation on the part of the students to achieve their overall learning goals. Moreover, it will assist teachers in managing large classes as it offers a problem-based approach.

REFERENCES

[1] Northedge, A. Rethinking Teaching in the Context of Diversity Teaching in Higher Education, Vol. 8, No. 1, 2003

[2] Biggs, J. What the Student Does: teaching for enhanced learning, Higher Education Research & Development, 18:1, 57-75, 1999

[3] Jones, G. and Knezek, G. Non-commercial radio-satellite telecommunications: affordable options for technology educators. Cited in S. Romi (2000) Distance Learning and Non-formal Education: Existing Trends and New Possibilities of Distance Learning Experiences. Educational Media International. 37 (1), 39-44, 1993

[4] Dede, C. Learning about teaching and vice versa. Paper presented at Conference of Society for Information Technology in Education. Washington D.C., USA.1998

[5] Forcheri, P. and Molfino, M. T. ICT as a tool for learning to learn. In Watson, D. M. and Downes, T. (Eds.) Communications and Networking in Education. Boston, MA: Kluwer Academic. pp 175-184, 2000

[6] Gunter, G. A. "The Effects of the Impact of Instructional Immediacy on Cognition and Learning in Online Classes", International Journal of Social Science , vol. 2 no. 3, pp. 196-202. 2007

[7] Perera, G.I.U.S. Impact of using agile practice for student software projects in computer science education. International Journal of Education and Development using Information and Communication Technology (IJEDICT) Vol. 5, Issue 3, pp. 85-100, 2009

[8] Lui, A. K., Kwan, R., Poon, M., & Cheung, Y. H. Y. Saving weak programming students: Applying constructivism in a first programming course. SIGCSE Bulletin, 36, pp.72–76, 2004

[9] Kris M.Y., Victor C.S. & Lee Y.T. Yu. Learning motivation in e-learning facilitated computer programming courses. Computers & Education 55, pp. 218–228, 2010

[10] Deek, F.P., & McHugh, J. Problem Solving and Cognitive Foundations for Program Development: An Integrated Model. Proceedings of the Sixth International Conference on Computer Based Learning in Science (CBLIS), Nicosia, Cyprus, pp. 266- 271, 2003

[11] Lam, M. S. W., Chan, E. Y. K., Lee, V. C. S., & Yu, Y. T. Designing an automatic debugging assistant for improving the learning of computer programming. Lecture Notes in Computer Science, 5169, 359–370, 2008

[12] Rolandsson, L. Changing computer programming education: the dinosaur that survived in school. An explorative study about educational issues based on teachers' beliefs and curriculum development in secondary school. The 2013 Learning and Teaching in Computing and Engineering, IEEE., 2013

[13] Wulf, T. Constructivist approaches for teaching computer programming. SIGITE'05, Newark, New Jersey, USA. October 20–22, 2005

[14] Middendorf, J. and Pace, D. Decoding the Disciplines: A Model for Helping Students Learn Disciplinary Ways of Thinking. New Directions for Teaching and Learning, No. 98, Summer 2004 © Wiley Periodicals, Inc.

[15] Perera, G.I.U.S. and Fernando, M.S.D. "Enhanced Agile Software Development — Hybrid Paradigm with LEAN Practice, In Proc. of 2nd International Conference on Industrial and Information Systems, ICIIS 2007, IEEE , pp. 239 – 244, 2007

[16] Highsmith, J., Cockburn, A. "Agile Software Development: The Business of Innovation", IEEE Computer, vol. 34, pp.120-127, 2001

[17] Maurer, F. and Martel, S. "Extreme programming: Rapid development for web-based applications," IEEE Internet Computing., vol. 6, no. 1, pp. 86–90, Jan./Feb.2002

[18] Manifesto for Agile Software Development: http://www.agilemanifesto.org (17/09/2015)

[19] Bipp, T., Lepper, A., and Schmedding, D. "Pair programming in software development teams – An empirical study of its benefits, Information and Software Technology, 50, pp. 231–240, 2008

[20] "A decade of agile methodologies: Towards explaining agile software development", The Journal of Systems and Software 85 (2012) pp. 1213–1221

[21] Balijepally, V., Mahapatra, R., Nerur, S., Price, K.H. Are two heads better than one for software development? The productivity paradox of pair programming. MIS Quarterly 33, 91–118.2009

[22] Schroeder, A. Klarl, A. Mayer, P. and Kroiss, C. Teaching Agile Software Development through Lab Courses, Global Engineering Education Conference (EDUCON), IEEE, pp.1-10, 2012

[23] Williams, L., McDowell, C. Nagappan, N., Fernald, J., and Werner, L. "Building pair programming knowledge through a family of experiments", In Proc. of International Symposium on Empirical Software Engineering, ISESE 2003, pp. 143-152, 2003

[24] Williams, L., Yang, K., Wiebe, E., Ferzli, M., and Miller, C. "In Support of Pair Programming in Introductory Computer Science Course", Computer Science Education, vol. 12(3), pp.197–202, 2002

[25] Devedzic, V. and Milenkovic, S.R. Teaching Agile Software Development: A Case Study, IEEE Transactions on Education, Vol. 54, No. 2, pp.273-278, 2011

[26] Williams, L. Wiebe, E. Yang, K. Ferzli, M. and Miller, C. "In Support of Pair Programming in the Introductory Computer Science Course," Computer Science Education, vol. 12, no. 3, pp. 197-212, 2002

[27] Gatfield, T."Examining Student Satisfaction with Group Projects and Peer Assessment". Assessment and Evaluation in Higher Education, Vol. 24(4), pp. 365-377, 1999

[28] Gomes, A., & Mendes, A. J. Learning to program – Difficulties and solutions, In International Conference on Engineering Education – ICEE 2007, Coimbra, Portugal

[29] Jenkins, T. On the difficulty of learning to program. In 3rd annual conference of LTSN-ICS, 2002.

[30] Krathwohl, D. R. A revision of Bloom's taxonomy: an overview. Theory into Practice 41, p. 212-218, 2002