ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 9 , September 2022

K-12 Agile Learning with Educational Software and Robotics Technology

Antonios Konstantaras, Kanellia Ioanna Dalaka, Maria Kourasi, Emmanuel Maravelakis

Associate Professor, Department of Electronic Engineering, Hellenic Mediterranean University, Romanou 3, Chania, GR73133, Greece
Educator and Author, Stamoulis Publications, Averof 2, Athens, GR10433, Greece
Doctorate Student, Department of Electronic Engineering, Hellenic Mediterranean University, Romanou 3, Chania, GR73133, Greece
Associate Professor, Department of Electronic Engineering, Hellenic Mediterranean University, Romanou 3, Chania, GR73133, Greece

ABSTRACT: This article discusses the application of agile learning sprints targeted to K-12 primary education students using educational robotics. Educational robotics are suitable for learning via experimentation, trial & error and retrial, which when inflicted in a form of game-play becomes pleasantly adopted by students. This has been conducted by implementing carefully designed cumulative sprints which are repeating a certain agile learning process, that breaks a larger problem into smaller parts/milestones, designing algorithms as to how to proceed step-by step towards solving the larger overall problem. This is done by evaluating and re-evaluating the progress of every step, until an outcome has been reached and reflected upon. The aforementioned process applies both and equally to tutors and students, respectfully. Results of applying agile learning using K-12 educational robotics demonstrate the ability of young students to cope and realize educational robotics construction, system understanding and algorithm design and implementation via Scratch-3 programming.

KEY WORDS: K-12 Software Technology, Educational Robotics, Agile Learning, Scratch Programming, Learning Sprints.

I.INTRODUCTION

The work presented investigates the application of agile learning methodologies using sprints based on Scratch 3 programming language and WeDo2 educational robotics, targeted at K-12 students in primary education. Standard teaching methods have emerged from excellent practices of the previous century [1]. These practices rely upon a unidirectional flow of information from the tutor to the students. The tutors are suppling knowledge to the students who act solely as receivers, reproducing and coping what is being taught. Traditional teaching methods are mainly governed by rather inflexible programs of study with set and often strict deadlines and evaluations only from the side of the teacher.

The latest pandemic brought rise to an opportunity for most students, and parents alongside the, to obtained resources on information and communication technologies (ICT) along with access to the internet, which enabled them, from very young, to familiarize themselves with these technologies [2]. This abrupt infliction of distance learning in K-12 students' life left little time to prepare for the transition between these two modes of study. As a result, most of the learning on how to use ICT occurred mostly by trial and error. These benefits though, were not in vain and are now being carried along as skills by the students that are now returning in person to on site school classes.

Educational robotics provide a suitable [3] tool for continuing the learning process via experimentation, trial & error and retrial in a laboratory classroom environment. Should it be introduced in a form of guided game play it then becomes a pleasantly acceptable task gladly followed by the students [4]. This is conducted via the implementation of carefully designed agile sprints, working cumulatively to one another, that repeat the same agile learning process [5].

The agile process [5] focuses on:
- breaking a wider problem into smaller parts and setting milestones, designing algorithm for each smaller part working complementary towards solving the overall problem,

- evaluating and re-evaluating the progress of the solution on every step along the way,
- reaching an outcome and reflecting upon it.

This process applies the same both to students, who learn to self-evaluate their performance and the solutions they are working upon, as well as teachers, who are being required to adjust their teaching approaches on either agile sprint accordingly in order to meet not only the needs of the class as an overall but the needs of individual's as well.

II. SIGNIFICANCE OF THE SYSTEM

This research work presents findings of a three-years period of applying agile learning with educational robotics, face-to-face, with in-class interaction to K-12 students. The development process that led to the establishment of an accredited teaching module is outlined along with developed materials both in terms of hardware (WeDo2 educational robotics) and software (Scratch 3 K-12 programming language) that were used in the process. The results being presented analyse the capability of students in three different age groups, six to eight, eight to ten and ten to twelve years old, to cope with building robotic devices with 3D graphics guidance, to construct algorithms for solving real-world problems and to implement algorithms using K-12 Scratch 3 programming.

III. LITERATURE SURVEY

In the last decade, the fourth industrial revolution (Industry 4.0) brought rise to a sudden need of increased ICT and programming skills. This rapid request enforced learning through real world problem solving which in turn made agile learning methodologies all that more favourable. Emerging from North America countries and now gaining ground in Europe and the rest of the world, agile methodologies are propagating downward in terms of age, to as early as in K-12 primary education.

Educational robotics provide a suitable and intriguing platform for K-12 students to play and learn how to solve a real-world problem. Learning is being conducted in the process where K-12 students learn how to build a robot, understand the system in terms of inputs and outputs, e.g. sensors and actuators, and devise, design and implement working algorithms via programming.

Evripidou et al., 2018 [6], investigated multiple educational robotics platforms and relevant competitions that promote them in order to assess the expected learning outcomes from the use of either platform. Tsalmpouris et al. (2021) [7] introduced a low-cost framework for stem education using open tools to address the substantial cost of the materials necessary for educational robotics to be introduced to public K-12 primary schools. Kakaras et al. (2022) [8] proposed a holistic tool for STEM education incorporating educational robotics. Sophokleous et al. (2021) [9] progressed a step further introducing computer vision to educational robotics. Evripidou et al. (2022) [10] made an effort towards establishing reliable criteria for selecting suitable robotics platforms for agile education. Feijoo et al. (2022) [11] combined hardware and software platforms involving educational robotics and Scratch programming for the development of STEM skills. Schifferle et al. (2021) [12] attempted agile rapid product development in K12 classrooms aiding students to learn via hands on programmable problem solving. Saleh et al. (2019) [13] reported K-12 students' evaluation of the agile learning methodology followed after completion of an adapted software engineering short course. Pahl et al. (2020) [14] studied the infiltration of programming principles to K-12 students in programming and non-programming classes. Last but foremost, it was Pinheiro et al. (2018) [15] who paved the way, having attempted to propose a holistic approach relating agile learning and educational robotics, for teaching basic principles of software engineering to K-12 students.

IV. METHODOLOGY AND MATTERIALS

The educational robotics materials used were the Lego WeDo2 programmable robotic construction kits. These kits were selected specifically as they serve nicely the following purposes: a) they are easily acceptable as toys by K-12 students, hence casting a sense game-play while learning at the process at the same time, b) because of safety issues due to critical ages of K-12 students. Furthermore, low energy wireless (Bluetooth) communication between the educational robots and personal computers was ensured via the use of low energy dongles. Scratch 3 programming language was deployed for implementing structured algorithms for programmable educational robotics by K-12 students [16].

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 9, Issue 9 , September 2022

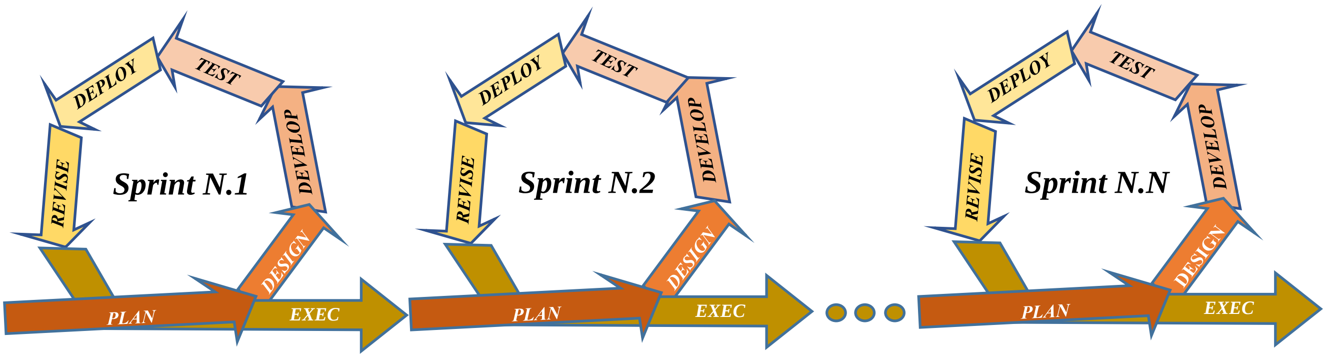

Fig. 1 Cumulative sprints in the agile learning cycle.

Teaching materials were developed targeted to support an agile learning approach where learning is conducted by problem solving, and students via trial and error, evaluation, re-evaluation and adjustment learn to devise asses and implement real-world problem-solving algorithms. A particular agile learning cycle (Fig 1) is being repeated in the form of short sets of exercises comprising each sprint. The agile cycle includes in order: planning a solution for solving a problem, designing an overall algorithm on how to do that often comprised of smaller collective algorithms that solve individual parts of the overall task at hand, development and testing of that algorithm every step across its development stages, deployment of the overall algorithm and revision of its outcomes, assessment and reassessment till final execution of the end product satisfies the goals of the project.

Classes were conducted both in primary schools and in the educational robotics laboratory of the Centre of Education and Life Long Learning of the Hellenic Mediterranean University in Chania, Crete, Greece for a period of three years. Two years were conducted before the pandemic and a year was conducted after the pandemic. All lessons were conducted face-to-face in suitably equipped laboratory classrooms. Students aged six to eight, eight to ten and ten to twelve participated in the programme, counting thirty students overall, ten at each different age group.

(a) (b)

Fig. 2 Instances of (a) 3D educational robots' building instructions and (b) control algorithm implementation via Scratch programming.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 9 , September 2022

Educational materials developed included, although not limited to, 3D building instructions for constructing educational robots using 3D graphics environments, as well as algorithm development exercises guiding the K-12 students in constructing step-by-step working Scratch programmable solutions to real-word problems. The purpose was for the K-12 students to realize that programming exceeds the console of their personal computer and can control programmable devices elsewhere in the real world. The above figure (Fig 2) displays instances from the construction (subplot ‘a’) of a line-follower robot and the control algorithm (subplot ‘b’) being implemented using Scratch.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Figures 3 to 5 present outcomes for each different age group (six to eight, eight to ten and ten to twelve years of age) from the assessment of students’ performance in three distinct categories: i) step-by-step 3D computer aided guided construction of an educational robot, ii) understanding the overall system and iii) algorithm design and structured programming implementation with Scratch 3. Percentages with respect to the weekly overall attendance were used for presenting the obtained results so as to compensate for scarce students’ absences. Fig 3 presents students’ performance with respect to 3D computer aided guided construction of an educational robot, Fig 4 presents students’ performance with respect to understanding the underlying system and Fig 5 presents students’ performance with respect to algorithm design and implementation using Scratch 3 to solve each sprint’s real world target problem.

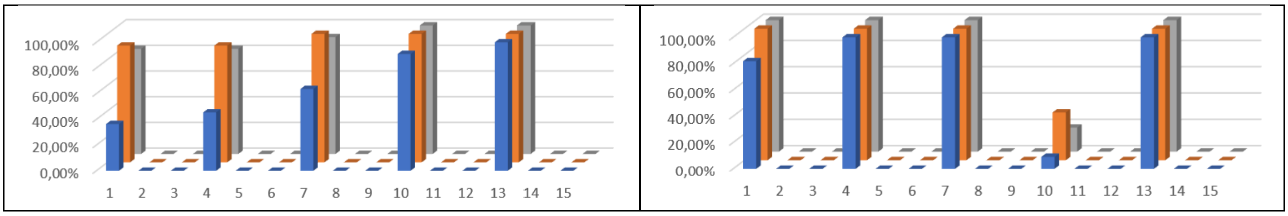

Fig. 3 Students’ weekly performance in guided robot construction. Age groups: [6-8) blue bar, [8-10) orange bar, [10-12) grey bar. Subplots: a) first 5 sprints (one week each) of a 15-week time period, b) last 5 sprints (one week each) of a 15-week time period with less tutor guidance.

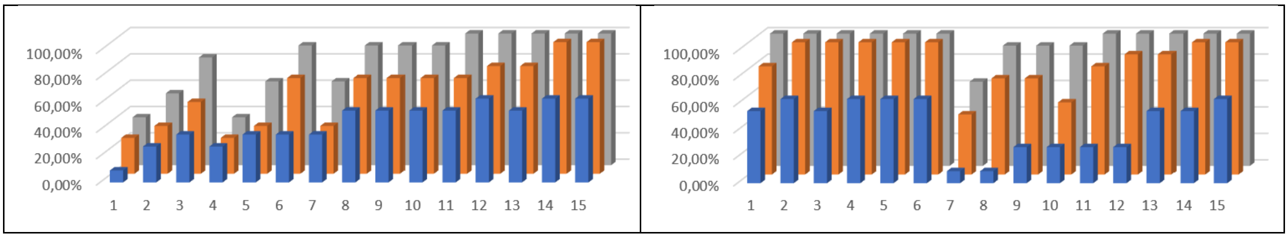

Fig. 4 Students’ weekly performance in system understanding. Age groups: [6-8) blue bar, [8-10) orange bar, [10-12) grey bar. Subplots: a) first 5 sprints (three weeks each) of a 15-week time period, b) last 5 sprints (three weeks each) of a 15-week time period with less tutor guidance.

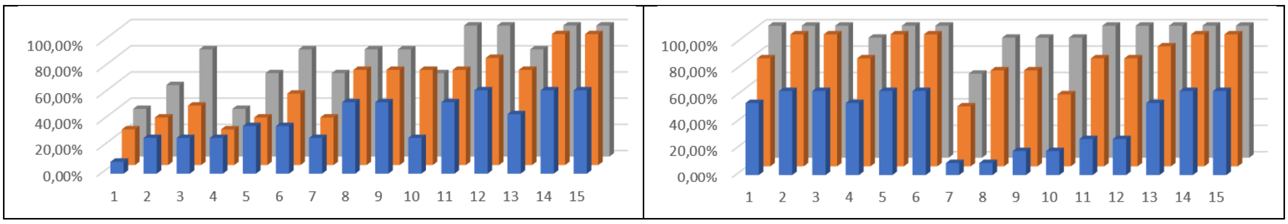

Fig. 5 Students’ weekly performance in guided structured Scratch programming. Age groups: [6-8) blue bar, [8-10) orange bar, [10-12) grey bar. Subplots: a) first 5 sprints (three weeks each) of a 15-week time period, b) last 5 sprints (three weeks each) of a 15-week time period with less tutor guidance.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 9 , September 2022

Reflecting to the above results brings rise to a number of noteworthy observations. Students aged eight to ten and ten to twelve demonstrate almost equally performances in comparison. The students' group aged six to eight years old appears to fall behind by a small margin. This could be contributed to the apparent [17] reading difficulty and understanding what was read on the console of their personal computer, which at these ages lags with respect to older students. Supportive to that is that results do not show inability regarding devising and implementing a working algorithm for the same age group. All groups regardless of age were troubled at the beginning of the second 15-week sprints when tutors were asked to minimally intervene and guidance was purposefully reduced to a minimum. A period of a month to a month and a half, i.e., two to three sprints was necessary for students to revive from the new teaching approach and foremost to realise and believe that they are equally as capable to manage both by themselves and via teamwork. During the last two sprints of the second 15-week sprints cycle all age groups managed to cope fully with the given tasks.

All students were capable (Figure 3) of following 3D instructions in building educational robots. In terms of systems' understanding, all three age groups were able to cope with after the first three sprints period (Figure 4) during the first guided learning cycle. All age groups managed to realise the essence of inputs and outputs to and from a system, in the form of sensors and actuators, respectively. Some difficulty was observed, especially in the case of the younger age group (ages six to eight), when keyboard buttons and mouse clicks were first introduced as inputs to sprints three (3) and four (4) of the second fifteen-week cycle, rather than sensors' recordings to which K-12 students were accustomed with until then.

Regarding structured programming using Scratch 3 (Figure 5), the age group six to eight managed to cope with tasks encompassing serial algorithms, loops and simple two-way branches. Age groups eight to ten and ten to twelve performed astonishingly well and were able to realise and use variables, Boolean variables and secondary functions, with the ten to twelve group performing slightly better than the eight to ten group. All groups were able to break larger problems into smaller ones and set milestones in order to design and programme both partial and overall algorithms with the aid of the tutor's guidance during the first fifteen-week cycle. In the second less guided fifteen week learning cycle, all three groups did struggle with that task and particularly for the six to eight age group guidance was once more deemed necessary lasting throughout the second cycle. For the groups aged eight to ten and ten to twelve, some students did manage to cope by themselves better than others, yet in occasions some guidance was once more deemed necessary to be offered to the entire class.

VI. CONCLUSION AND FUTURE WORK

Industry 4.0 requires software engineering and emergent technologies understanding and usage skills to be developed from as early as the K-12 band [18]. Programmable devices for ages under the K-12 band starting as early as the first classes in primary schools become all that more common to primary school curricula [19] to more and more countries around the globe. Alongside maths and languages students are capable to develop further skills such as analytical thinking and algorithmic thinking. The test case presented in this research article demonstrates that K-12 students are capable of coping, with some guidance from the tutors, with educational robotics both in terms of construction, system understanding and programming. K-12 students are capable of realising the meaning of sensors and actuators thereby understanding the essence of a system and the role of its inputs and outputs. K-12 students also appear capable of devising and implementing structured programming algorithms to produce a solution to a problem using Scratch 3 programming language, which has been especially designed for use by the K-12 student group. More importantly, agile learning and educational robotics enable students to realise that programming affects the outer world and is not limited to the console of their personal computer. This teaching approach helps students to learn while trying to solve real life problems. Learning is conducted in the process of problem solving by attempting, evaluating, adjusting and trying and re-trying again and again until the final goal is achieved. This approach of learning promotes the most valuable "Can Do" attitude.

Further work shall focus in invoking affective computing [20] in class during real time agile teaching. The purpose is to assist the tutor in implementing the agile learning process with a real time notification of the feelings depicted by various K-12 students while working with a particular learning sprint. Information regarding depicted feelings such as anxiety, success, joy, boredom, etc, shall be transferred to the tutors during the class lesson, in real time, allowing them to instantly intervene as they see fit to resolve or excel an ongoing learning situation or even adjust their teaching approach of a particular agile learning sprint.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 9 , September 2022

REFERENCES

[1] H. H. Tillema, H. H. "Stability and change in student teachers' beliefs about teaching", Teachers and Teaching: Theory and Practice, vol. 4, no. 2, 1998, pp. 217-228.
[2] J. A. Manco-Chavez, Y. C. Uribe-Hernandez, R. Buendia-Aparcana, J. J. Vertiz-Osores, S. Alcoser and R. A. Rengifo-Lozano, "Integration of icts and digital skills in times of the pandemic COVID-19", International Journal of Higher Education, vol. 9, no. 9, 2020, pp. 11-20.
[3] F. B. V. Benitti, "Exploring the educational potential of robotics in schools: A systematic review", Computers and Education, vol. 58, no. 3, 2020, pp. 978-988.
[4] R. Higashi, E. Harpstead, J. Solyst, J. Kemper, J. Odili Uchidiuno and J. Hammer, "The design of co-robotic games for computer science education", CHI PLAY 2021 - Extended Abstracts of the 2021 Annual Symposium on Computer-Human Interaction in Play, 2021, pp. 111-116.
[5] M. Paasivaara, V. Heikkilä, C. Lassenius and T. Toivola, "Teaching students scrum using LEGO blocks", 36th International Conference on Software Engineering, ICSE Companion 2014 - Proceedings, 2014, pp. 382-391.
[6] S. Evripidou, K. Georgiou, L Doitsidis, A. A. Amanatiadis, Z. Zinonos and S. Chatzichristofis, "Educational robotics: Platforms, competitions and expected learning outcomes". IEEE Access, 2020, pp. 219534-219562.
[7] G. Tsalmpouris, G. Tsinarakis, N. Gertsakis, S. A. Chatzichristofis and L. Doitsidis, "Hydra: Introducing a low-cost framework for stem education using open tools", Electronics (Switzerland), vol.10, no. 24, 2021, pp. 1-19.
[8] G. Kakaras, E. Goumenakis, E. Glynos, P. Spanoudakis, G. Tsinarakis, N. Tsourveloudis, and L. Doitsidis, "HYDRA 2.0: Towards developing a holistic tool for STEM education", 30th Mediterranean Conference on Control and Automation, MED 2022, 2022, pp. 1099-1104.
[9] A. Sophokleous, P. Christodoulou, L. Doitsidis and S. A. Chatzichristofis, "Computer vision meets educational robotics", Electronics (Switzerland), vol. 10, no. 6, 2021, pp. 1-24.
[10] S. Evripidou, L. Doitsidis, G. Tsinarakis, Z. Zinonos and S. A. Chatzichristofis, "Selecting a robotic platform for education", Digest of Technical Papers - IEEE International Conference on Consumer Electronics, 2022, pp. 1-6.
[11] A. Feijoo-Almonacid and C. H. Rodriguez-Garavito, Hardware-software platform for the development of STEM skills. Revista Iberoamericana De Tecnologias Del Aprendizaje, vol. 1, no. (2, 2022, pp. 170-177.
[12] T. M. Schifferle and N. Kollegger, "Enabling agile rapid product development in K12 classrooms by enhancing an educational exoskeleton", ACM International Conference Proceeding Series, 2021, pp. 1-3.
[13] S. M. Saleh and S. R. Philip, S. R, "Students' evaluation in AGILE after completing software engineering course", 1st International Conference on Advances in Science, Engineering and Robotics Technology 2019, ICASERT-2019, 2019, pp. 1-6.
[14] C. Pahl and I. Fronza, "Teaching software engineering principles in programming and non-programming activities at schools", CEUR Workshop Proceedings, 2531, 2020, pp. 68-69.
[15] F. C. Pinheiro, C. G. von Wangenheim and R. M. Filho, "Teaching software engineering in K-12 education: A systematic mapping study", Informatics in Education, vol. 17, no. 2, 2018, pp. 167-206.
[16] J. Fagerlund, M. Vesisenaho and P. Häkkinen, "Fourth grade students' computational thinking in pair programming with scratch: A holistic case analysis", International Journal of Child-Computer Interaction, vol. 33, no. 2, 2022, pp. 1-19.
[17] V. Yigit-Gencten and M. Gultekin, "Nature-based reading and writing instructions in early childhood education: The giving tree example", Environmental Education Research, vol.28, no. 1, 2022, pp. 95-108.
[18] O. Howson, P. Charlton, F. Iniesto and W. Holmes, "INSPIRE: Fourth industrial revolution teaching in the classroom", SIGCSE 2022 - Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V.2, 1099, 2022, pp. 1099.
[19] A. Jagadeesha, P. Rayavaram, M. Bonam, S. Narain, C. S. Lee, "AlgoScratch - simplifying data structures and algorithms education using block-based programming", Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE, vol. 2, 2022, pp. 625.
[20] H. J. Yun and J. Cho, "Affective domain studies of K-12 computing education: A systematic review from a perspective on affective objectives", Journal of Computers in Education, vol. 9, no. 3, 2022, pp. 477-514.