Re-Imaging Learning Environments
Proceedings of the European Distance and E-Learning Network 2016 Annual Conference
Budapest, 14-17 June, 2016
ISBN 978-615-5511-10-3
ISSN: 2707-2819
doi: https://doi.org/10.38069/edenconf-2016-ac-0007

IF LEARNING TO CODE IS NOT ABOUT CODING, THEN WHAT IT IS ABOUT?

Koen DePryck, Vrije Universiteit Brussel, Jens Vermeersch, Annemie Tytgat, GO! Onderwijs van de Vlaamse Gemeenschap, Belgium

To code or not to be…

The need for programming professionals is still growing. The European Commission estimates there are 700,000 unfilled vacancies for IT practitioners, of which programmers outnumber other IT professionals by 5 to 1. In the Netherlands development jobs account for 51.6% of all IT jobs.

The limited response from the different European education systems is worrisome, although there are signs that awareness is on the rise. Estonia has introduced programming across the curriculum in 2012, Denmark is following suit. So are the UK, Finland and Ireland. In Germany some regions are more on the forefront than others. It should be noted that schools are not the only answer to this challenge – so are for example after-school coding clubs as well as initiatives from the IT industry itself – but they are, obviously, a key stakeholder.

Introducing coding in the curriculum at an early age is a long term investment in bridging the skills gap between the technology demands of the labour market and the availability of people to fill them. The key seems to be moving from mere literacy to active control.

One of the problems all are encountering is the shortage of not only generally ICT literate teachers but programming savvy teachers.

At the same time, the art – or skill – of programming is changing rapidly as languages are evolving and, perhaps even more important, interfaces and environments are becoming more intuitive, away from a write-run-test-edit cycle and towards interactive programming environments assisting the developer while writing code, and not just after writing it. This lowers the threshold for meaningful output substantially without raising the frustration of learners to a level where they are prone to abandoning the project at hand. Even in robotics, a few days of initiation allow students to deliver stunning results. Programming moves towards writing a logical map of the desired functionality.

These changes make programming increasingly accessible to end users. Perhaps not, as the IT industry might require, as seasoned developers capable of e.g. Sophisticated parallel programming but definitely in the sense of high level users capable of designing smart apps handling their own needs for collection and presentation level transfer of information customization and personalization of all the devices they are using (PC, laptop and smartphone and soon extending into an internet of things, wearables and more.)

Also important is a trend towards interactive programming environments dealing with a specific context: business related, science oriented, creative manipulation and more. This development facilitates the inclusion of coding in curriculums designed for very different learner profiles.

Metacognition

Computer programming requires higher order thinking skills, including metacognitive abilities. Many seem to expect that including programming at an early age in the curriculum will improve metacognitive skills. This is important as research observes at the level of university students a lack of such skills in first year programming students.

Blakey and Spence (1990) identify the following basic metacognitve strategies:

- Connecting new information to former knowledge;
- Selecting thinking strategies deliberately; and
- Planning, monitoring and evaluating thinking processes.

Chetty and van der Weshuizen (2014) built on this framework and on work by Jenkins (2001) on expressing feelings and emotions in a classroom to establish a metacognitive learning environment, including small groups, an appropriate classroom arrangement and an atmosphere of openness, discussion and relaxation.

A metacognition strategy card was designed to assist learners in developing and selecting a strategy while programming. Reflexive questionnaires, focusing on weekly results, were added. A plan-of-action sheet assists in planning and monitoring.

But as Chetty and van der Weshuizen (2014) indicate, "merely providing learners with a set of metacognition strategies may be insufficient. Guiding them on how to approach and utilize such strategies may be required."

TACCLE 3

TACCLE 3 (following TACCLE which focussed on developing teacher's competences in developing digital content and TACCLE 2 which focussed on learners using programs and apps to create their own content) is an EU funded strategic partnership (2015-2017) for school education focussing on introducing coding in schools in light of a competitive strategy after 2020. In order to do so, the partnership identified the following needs

- Professionalization and upskilling of teachers ;
- Enhanced digital literacy for teachers and learners; and
- A positive mindset of young children towards coding and STEM.

From a critical perspective and in light of the previous sections, we would like to make the following comments.

Metacognitive strategies must be an integral part of such a project. The pre-service ad in-service professionalization of teachers should therefore not only include the use and mastery of coding environment but also insights in metacognitive strategies and tools to teach them. Learning how to guide students in the use of metacognitive strategies may be essential.

The link with STEM may be less crucial than one might expect. As mentioned earlier, the development of interactive coding environment catering to a specific segment (business, science, arts…) may open up opportunities for coding in very different domains, without necessarily requiring an interest in STEM. While the need for more students to become interested in STEM is need in its own right, strongly connecting that need to coding might not only scare away learners without that explicit interest in STEM but might also hinder the pervasive inclusion of coding skills in a broad array of domains. A STEM mindset can be developed and integrated with programming skills, but that is by no means a requirement or a desired outcome. The reach of ICT is much broader than STEM alone and a background in STEM is no longer a requirement for successful coding.

Conclusion

Introducing coding in the curriculum is really not about coding itself. It is, rather, about introducing a culture of algorithmic thinking, breaking down more complex actions into a sequence of instructions. This computational thinking is itself based on a set of metacognitive strategies in a wide array of domains, applicable beyond coding.

References

1. Blakey, E., & Spence, S. (1990). Developing Metacognition. *ERIC Digest* (ED327218), 5.
2. Chetty, J., & van der Westhuizen, D. (2014). *Implementing Metacognition Skills for Learners Studying Computer Programming.* Paper presented at the Edmedia Conference, Tampere, Finland. Retrieved from https://www.editlib.org/f/147775/
3. Jenkins, T. (2001). *Teaching Programming – A Journey from Teacher To Motivator.* Paper presented at the 2nd Annual LTSN-ICS Conference.