Co-funded by the
Erasmus+ Programme
of the European Union

Accelerating the transition towards Edu 4.0 in HEIs

TEACH 4EDU4

Catalogue of new forms of teaching, learning and assessment in Computer Science in Edu 4.0 and related teachers' skills and competences

The Open University: Rebecca Ferguson, Christothea Herodotou, Francisco Iniesto, Bart Rienties, Julia Sargent

University of l'Aquila: Henry Muccini, Maria Vittoria Isidori **The University of Belgrade:** Jeremic Veljko **Polytechnic University of Catalonia:** Ariadna Llorens, Marc Alier **Tallinn University:** Sirje Virkus **University of Zagreb:** Goran Hajdin, Miran Zlatovic **The University of Žilina:** Dalibor Gonda, Michal Kvet, Peter Marton, Marek Moravcik, Jozef Papán, Jana Uramova.

Milton Keynes, UK, 01 June 2021

Executive Summary

1. This catalogue is the first intellectual output of the TEACH4EDU4 project. The overall project aim is to enable the creation of an environment that supports implementation of new learning and teaching approaches in Computer Science (CS) and related disciplines. It is designed to support Higher Education institutions (HEIs) across Europe as they incorporate innovative practices within their teaching – particularly their CS teaching.
2. The catalogue is based on a systematic review of recent research literature dealing with innovations in the teaching and learning of CS. The focus is on practices that align with Education 4.0, an area that connects new educational practices with the needs of industry. The catalogue also considers the skills and competences that will be required by educators as they engage in innovative practice.
3. Definitions about Education 4.0 vary but usually focus on innovation, novelty, use of technology, and connections with employment and industry (Hussin, 2018; Salmon, 2019). The number 4.0 makes a connection with the view that there have been four industrial revolutions with the current Industry 4.0 increasingly automated, making use of modern smart technologies and the Internet of Things (objects that can exchange information over the Internet). What Education 4.0 means in practice is still being negotiated, so this catalogue uses two definitions.
    a. The first is a well-cited version from Fisk (2017) and further updated by Hussin (2018), which included nine characteristics of **Education 4.0** (i.e., 1) Anytime / Anywhere; 2) Personalised; 3) Choice in how to Learn; 4) Project-based Learning; 5) Hands-on Learning; 6) Data Interpretation; 7) Assessed in New Ways; 8) Student Opinion Counts; 9) Develops Independence).
    b. We propose an updated definition of Education 4.0: "Education 4.0 employs an approach to learning and teaching that emphasises the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment. The approach involves the use of technology and/or pedagogy that is innovative in the context, and therefore requires flexible and creative approaches to its implementation."
4. Aligning the needs of industry with CS programmes requires an overview of the subject area from outside the academy. The TEACH4EDU4 project therefore carried out focus groups that involved employers and employment experts from across Europe. These focus groups identified the skills and capabilities that employers are looking for when they recruit CS graduates.
5. Conversations with the nine industry partners indicated four main emergent themes, namely 1) Gaps in skills or knowledge from university; 2) Strengths CS graduates bring from university to the company; 3) Ideas for innovative ways of addressing these gaps; 4) Areas that companies work with their graduates to develop.
6. According to the industry partners the main gaps that graduates have are a lack of soft skills such as communication and teamwork. Similarly, CS graduates often struggle to apply their detailed theoretical knowledge to different practical contexts. The strengths they supply to the company mainly include knowledge of programming and interacting with clients and customers on a technical level. There were various suggestions in ways that these gaps could be addressed which include company placements, guest speakers and project-based learning. Areas that companies feel that they could work with their graduates

Co-funded by the Erasmus+ Programme of the European Union

to develop include areas such as confidence in their own abilities, knowledge, skills, and competences.

7. In order to identify the skills and competencies required by educators in order to implement innovations in CS education, two approaches were taken. The first was to draw on research carried out by the Technology-Enhanced Learning Programme (Scanlon et al., 2013), which used desk research, interviews with experts, and case studies from around the world to identify successful approaches to developing and embedding educational innovations successfully. The second approach used a Systematic Literature Review (SLR) to identify skills and competencies that were highlighted by authors when describing innovative practices.
8. Pedagogy and technology offer ways of achieving the initial vision of innovative practice. These two elements are closely related, particularly in a technology-rich subject area such as CS. Education 4.0 is likely to involve not only new devices or technical approaches but also new ways of teaching and learning. A recurring theme we identified in this report was the teacher as facilitator or moderator or consultant of learning as opposed to a teacher controlling or being the centre of the learning process. Teachers could achieve that by being flexible (adopt to change), supportive, help students to develop ownership of learning, foster an environment where students take risks and share what they do not know about, and where failure is acceptable.
9. Educators and learners are key actors, but there are also other enablers without whom any innovation cannot last long. These are the managers and department heads who approve funding and infrastructure and the senior leadership teams who develop the policies that block or encourage innovations. Learners, educators and other enablers need to be engaged throughout the life of the innovation but their roles are likely to change as it moves from novelty to business as usual.
10. In order to investigate which innovations are being introduced in the field of CS, a systematic literature review was carried out, focusing on three research questions.
    a. RQ1: Which pedagogic approaches are used to support the teaching of computer science to undergraduate and postgraduate students?
    b. RQ2: Which of these approaches align with Education 4.0?
    c. RQ3: What skills and competences do HE educators require in order to align their computer science teaching with Education 4.0?
11. In terms of RQ1 and RQ2, of the 66 articles selected perhaps surprisingly none of the articles explicitly mentions “Education 4.0”. In part this could be a result of the relatively recent conceptualisation of Education 4.0, and in part this could be due to a lack of adoption of the term Education 4.0 in the specific discipline of CS.
12. Based upon our broad definition of Education 4.0 in total 54 articles (80%) were considered to fit under this definition. Furthermore, all 66 articles included at least one Hussin (2018) Education 4.0 characteristic. The most common Education 4.0 characteristic was 5) Hands-on Learning (73%), followed by 9) Develops Independence (67%), and 4) Project-based Learning (61%). Around half of the articles included the characteristic that 1) Anytime / Anywhere; 7) Assessed in New Ways (35%) and 8) Student Opinion Counts (32%).
13. A cluster analysis indicated a three-cluster solution across the 66 papers, which we label as 1) EDU 4.0 light (n = 18), 2) project-based/hands-on learning (n = 22), and 3) full EDU 4.0 (n = 26).
14. **EDU 4.0 light studies** mostly had relatively low Hussin (2018) total scores, and often did not include project-based activities. EDU 4.0 light studies mostly focussed on 9) Develop Independence (61%), 1) Anytime / Anywhere (44%), 2) Personalised (39%), and 3) Choice

Co-funded by the Erasmus+ Programme of the European Union

in how to learn (39%), but with limited 5) Hands-on learning (17%) to no 4) Project-based learning (0%). For example, Schäfer (2019) introduced the concept of a modern C ++ course for students of electrical engineering and CS based on a flipped classroom and with pleasant Internet of Things (IoT) hardware. The main goal of the new course was to reduce lecture time in favour of practical learning of students through programming.

15. The second cluster that we labelled as **project-based/hands-on learning** had a strong focus on 4) Project-based Learning (86%) and 5) Hands-on Learning (86%), with relatively limited focus on 3) Choice in how to learn (5%), 2) Personalised (5%), and 1) Anytime / Anywhere (18%). For example, Caceffo, Gama, and Azevedo (2018) assessed the benefits of the use of technology and active learning practices (i.e., Problem-Based Learning) in the classroom with 24 students for undergraduates in Brazil to contribute to a more effective and efficient CS learning environment.
16. The third and final cluster, **full EDU 4.0**, was strongly focussed on 5) Hands-on learning (100%), 9) Develops independence (96%), 2) Personalised (85%), 1) Anytime / Anywhere (77%), and 3) Choice in how to learn (77%). The lowest Hussin (2018) characteristic was 8) Student opinion counts (38%), although this was substantially higher than the other two clusters. For example, Pivkina (2016) described the experience of undergraduate students as peer learning assistants (PLAs) in two presentations of an undergraduate CS course in the USA; the impact of the experience was measured by comparing student grades in the offerings of the course with and without PLAs.
17. In terms of RQ3, nearly half of the papers (n = 30) reviewed made an explicit reference to skills and competences Higher Education (HE) educators should have or develop to align their computer science teaching to the broader definition of Education 4.0. Furthermore, the aggregate Hussin (2018) score was substantially higher in articles that mentioned skills and competences of teachers relative to those who did not mention those skills. In other words, those articles who explicitly referred to skills and competences for teachers seemed to be more explicit and innovative in terms of pedagogies and EDU 4.0 elements. An alternative explanation could be that the authors of more innovative pedagogical approaches and full EDU 4.0 modes provided more narratives about how teachers could effectively support these innovative approaches.
18. From an EU perspective, we only found six studies from Europe that were classified as EDU 4.0 light. Five out of six studies aimed to develop independence and anytime/anywhere learning. At the same time, a mix of other Hussin (2018) activities were included in these six studies, although mostly just one or two characteristics. Seven studies from Europe were classified as project-based/hands-on studies, whereby all but one included both 4) Project-based Learning and 5) Hands-on Learning in their designs. Five studies were labelled as full EDU 4.0 studies where nearly all of the Hussin activities were included. Overall, while there are some engaging and diverse practices in CS and Education 4.0 in Europe, it seems that relative to other countries (e.g., USA) more work needs to be done. This is one of the aims the TEACH4EDU project will address.

Co-funded by the
Erasmus+ Programme
of the European Union

CONTENTS

Co-funded by the Erasmus+ Programme of the European Union

1. Introduction

This catalogue is an output of the TEACH4EDU4 project (https://teach4edu4-project.eu/). The overall project aim is to enable the creation of an environment that supports implementation of new learning and teaching approaches in Computer Science (CS) and related disciplines. This catalogue is the first output of the project. It is designed to support HEIs across Europe as they incorporate innovative practices within their teaching – particularly their CS teaching.

The catalogue is based on a systematic review of recent research literature dealing with innovations in the teaching and learning of CS. The focus is on practices that align with Education 4.0, an area that connects new educational practices with the needs of industry. Recent innovative practices from around the world are described in the section on 'Innovative Teaching Methods'. The catalogue also considers the skills and competences that will be required by educators as they engage in innovative practice.

Aligning the needs of industry with university teaching programmes requires an overview of the subject area from outside academia. The project therefore carried out focus groups that involved employers and employment experts from across Europe. These focus groups identified the skills and capabilities that employers are looking for when they recruit CS graduates.

1.1. Computer Science and Education 4.0

CS is a broad subject area that covers many areas and overlaps with many others. This catalogue uses the definition provided by the UK Quality Assurance Agency Subject Benchmark Statement on Computing: 'Computer science provides the necessary knowledge to understand and build computational systems' (Quality Assurance Agency, 2019). The statement goes on to list the main characteristics of CS and notes that, 'Generally, these are expressed in the ability to specify, design and write computer programs.' The breadth of the field means it is able to draw on teaching methods from diverse disciplines, and is also used as a way of preparing students for a wide range of professions.

This catalogue focuses on approaches to teaching and learning that are aligned with Education 4.0. This is a relatively new term – Harkins proposed it in 2008 to describe innovation-producing education (Harkins, 2008) as opposed to knowledge-producing education. Definitions vary but usually focus on innovation, novelty, use of technology, and connections with employment and industry (Hussin, 2018; Jisc, 2019; Salmon, 2019). The number 4.0 makes a connection with the view that there have been four industrial revolutions with the current Industry 4.0 increasingly automated, making use of modern smart technologies and the Internet of Things (objects that can exchange information over the Internet).

There is also a connection with developments in the world wide web: Web 1.0 was read only, Web 2.0 was the read-write web, and Web 3.0 the semantic web (using artificial intelligence and a machine-based understanding of data). Web 4.0 is still on the horizon, but is predicted to bring humans and technology closer together, making use of both augmented and virtual reality.

Note that we also need to consider the shallowness and possible consequences of these new technology evolutions. We are dealing with and trying to fix as a society the second order consequences of the Web 2.0 which gave birth to social networks, cloud computing and big data. The effects of Web 2.0 on society have been complex and diverse, including polarization, exploitation of personal data for behaviour manipulation (Harris, 2016), surveillance capitalism

Co-funded by the Erasmus+ Programme of the European Union

(Zuboff, 2015), and substantial privacy issues (Baker & Hawn, 2021; Korir, Slade, Holmes, & Rienties, 2021; Webb & Doctorow, 2020).

As a technological evolution we should remember that the Web 3.0 (semantic web) was based on the promise of semantic content embedded on the web, plus ontologies to interface domains of knowledge (Berners-Lee, Hendler, & Lassila, 2001). This promise was never realised due to the lack of adoption of standards and real-world use. This failed approach has been replaced by brute force data gathering and indexing, machine learning analysis, the use of Mechanical Turk (Paolacci & Chandler, 2014) to catalogue huge datasets, and the results are proving to be profoundly biased and flawed. The Web 3.0's second order consequences are insinuating on China and other instances of "Black Mirror like" situations. The unresolved issues that Web 2.0 poses, and the technological pivot of the Web 3.0 and its still unforeseen consequences (both positive and negative) is a cautionary tale of the muddy waters we step in when we wander about the opportunities lying in the realm of the 4.0.

Hence, the meaning of what Education 4.0 means in practice is still being negotiated, so this catalogue uses two definitions. The first is a well-cited version from Peter Fisk, a professor of leadership, strategy and innovation in Madrid. Writing on his Gamechangers website (Fisk, 2017), he identified nine trends associated with Education 4.0:

- **Diverse time and place:** Students will be able to learn where and when they choose.
- **Personalised learning:** Study tools will adapt to the capabilities of the student.
- **Free choice:** Students will be able to modify their learning process.
- **Project based:** Students will learn to apply their skills in a variety of situations.
- **Field experience:** Students will have authentic experiences and gain real-world skills.
- **Data interpretation:** Students will learn to interpret and reason with data.
- **Exams will change completely:** Knowledge and skills will be assessed in new ways.
- **Student ownership:** Students will have critical input into their courses.
- **Mentoring will become more important** and students will become more independent.

These characteristics of Education 4.0 were further updated by Hussin (2018), and formed part of the review of relevant literature, as indicated in Table 1. In the catalogue below, they are represented in tables, with (Y)es indicating that the approach to teaching and learning that is described includes that characteristic to a greater or lesser extent. If a particular characteristic was not identified by the coders, this will not be included in the catalogue.

Table 1 Characteristics of Education 4.0 (Hussin, 2018)

Anytime / Anywhere	Y
Personalised	Y
Choice in how to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

These elements identified by Fisk (2017) and further updated by Hussin (2018) are all potentially innovative. However, they refer to trends that were apparent in 2018, rather than the latest developments. They also focus on students rather than on the broader picture of how innovations

Co-funded by the Erasmus+ Programme of the European Union

are developed and embedded. Genuinely innovative programmes such as ‘Sustainable educational innovation through engaged pedagogy and organizational change’ (Frevert et al., 2018), which is described below, appear to be very limited examples of Education 4.0 when classified in this way.

This report therefore proposes a definition of Education 4.0 that draws on ideas and descriptions in a range of literature (Bonfield, Salter, Longmuir, Benson, & Adachi, 2020; Costa, Silva, & Fonseca, 2012; Jisc, 2019; Puncreobutr, 2016; Salmon, 2019; Suhaimi, 2019; Wallner & Wagner, 2016) and that aligns with the aims of TEACH4EDU4.

Education 4.0 employs an approach to learning and teaching that emphasises the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment. The approach involves the use of technology and/or pedagogy that is innovative in the context, and therefore requires flexible and creative approaches to its implementation.

In the tables in the catalogue below, a row at the top indicates whether a new educational practice aligns with this definition.

Education 4.0	Y

2. What employers are looking for

As Education 4.0 emphasises the development of skills and competences necessary in a modern workplace, it was important to find out what these skills are including the opinion of professionals in the sector. For that purpose, a series of focus groups involving employers from across Europe was used to identify the skills required and the current gaps in training.

2.1. Focus groups

Participants for the focus groups have been approached from a closed list provided by the TEACH4EDU project partners. The focus groups have included eleven participants as disclosed in Table 2 from six countries. In total seven focus groups sessions were run: 1 with 3 participants, 2 with 2 and 4 with a single participant. The sample included different types of companies: foundations, small enterprises or start-ups and large international enterprises, as indicated in

Table 3. All of them have links with the project partners as indicated.

Table 2 Focus groups for Teach4Edu sample

Country	Participants
Croatia	4
Slovakia	4
Italy	1
Spain	1
The UK	1
Total	**11**

Co-funded by the Erasmus+ Programme of the European Union

Table 3 Companies included in the focus groups

Company	Country	Company type	Participants
TICM	Croatia	Technology park	2
GoranUX	Croatia	Private design company - UX and marketing design	1
Rodiger d.o.o.	Croatia	IT services company	1
Ringier Axel Springer	Slovakia	Private Company	2
Scheidt&Bachmann Slovensko	Slovakia	Private Company	1
Siemens Healthineers	Slovakia	Private Company	1
Raspberry Pi Foundation	The UK	Private Foundation	1
SE4GreenDeals Erasmus Mundus Master Program	Italy	University	1
Col·legi Oficial d'Enginyeria en Informàtica (COEINF)	Spain	Association	1

Research through focus groups had the approval of The Open University Human Research Ethics Committee and all participants signed a consent form to participate. The language for the focus groups was English, to facilitate a mixture of opinions from different countries and the analysis. There were two researchers from different partners institutions moderating the focus groups, in this case, those roles were covered by the Open University and the Polytechnic University of Catalonia. The semi-structured focus groups were based on several key themes and lasted about 45-60 minutes. In Appendix B the respective questions and structure of the focus groups are detailed. Full transcriptions of the focus groups were produced by the tool Otter (https://otter.ai/) automatically and then reviewed manually for consistency. The transcriptions were added to Nvivo software for their analysis. Thematic analysis as described in the next section was selected as the analysis method.

2.2. What employers identify and want from CS graduates

In total four main themes emerged from the conversations with the nine industry partners, namely 1) Gaps in skills or knowledge from university; 2) Strengths CS graduates bring from university to the company; 3) Ideas for innovative ways of addressing these gaps; 4) Areas that companies work with their graduates to develop.

1) Gaps in skills or knowledge from university

Participants discussed how CS graduates tended to arrive at their company or business with a sound knowledge of technical skills in specific pieces of technology but, often lacked what was described as 'softer skills' (i.e., communication skills, organisation, teamwork). For example, a participant discussed this in relation to the gap in certain types of skills. "From my perspective, our students have good conceptual skills and good enough technical skills, but sometimes not enough, not yet enough social and communication skills." (Focus Group 9). This area was also discussed throughout the other focus groups:

> *I think that entrepreneurship is also something that we should push a little bit more and ability really to be independent and taking decisions. Some of the students are still immature in understanding how to run a meeting, how to organise notes, how to organise*

Co-funded by the Erasmus+ Programme of the European Union

> *tasks, for example. So this is more from, let's say, an organisational point of view*. (Focus Group 9)

> *For me, the communication is an important thing, because maybe not in companies who everyone is computer engineer. So knowing how to communicate at least certain things. It's important, especially when you have to communicate it to people who the only thing they know about computers is that they have one button to start them up. So, communication skills it's important. It's a thing that usually it's totally forgotten* (Focus Group 2).

Participants often talked about generic training that they offered new graduates, such as getting to know the culture of the business, IT processes, etc., but more individualised and personalised professional development. For example, in focus group 6, they discussed the importance of getting to know the culture of the company:

> *Getting the feel of the company's culture. So this is really important, because you can have someone with great technical skills, but whose values or work approach to work is not in line with the company's culture. So that can be a problem. So internalising the values of the company is also a really important part when joining a team (Focus Group 6).*

Graduates were perceived too often to struggle to apply their technical skills from what they had been taught at university to different scenarios or projects. Therefore, the graduate's ability to move from applying their technology skills on a theoretical level into a practical application was seen as a gap by those we interviewed. Indeed, it was seen that perhaps graduates were too focused on particular pieces of technology or software rather than having a broader knowledge in terms of applying the different technologies to projects or contexts.

> *I think this is the biggest gap in in current environment, like the students are mostly really good prepared in the "how". So they know how to build something. And they know like, when they know the when they do they know the how and they know the who but they don't know the why. So understanding the why and asking the right questions. To bring the solution in terms of time use scalability is something that that I would really appreciate if was more of a focus* (Focus Group 7).

Other gaps that were discussed included lacking passion, appropriate attitude or motivation for their new roles and the ability to experiment with the learning that they had gained from university. For example,

> *What we see is that many of them are simply not motivated or willing to step out of those boundaries. And they just go with the or do the minimal. They the minimum they require for gaining a degree or something. (Focus Group 6).*

That being said, this was not the case for all the focus groups. Indeed, in focus group 10 the inverse was discussed in terms of graduates bringing drive and motivation into the company.

> *Typically graduates brings that I've brought into the company, they are young, they want to really show the to the world that they are really skilled and they are ready to go to achieve something, and they have the very strong drive, and they are able to, I would say, change the flow in the company. So they are bringing a lot of drive into the company* (Focus Group 10).

Other gaps included asking critical questions, challenging others, decision making or knowing when to ask for help.

2) Strengths CS graduates bring from university to the company

The skills that CS graduates would bring from university included in-depth and up to date programming or technical knowledge. This might be knowledge of new software or hardware that

Co-funded by the Erasmus+ Programme of the European Union

other employees in the company might not have and so this was seen to address a skills gap. This was illustrated by colleagues in Focus Group 10 and 2:

> *[g]raduates are typically very well prepared regarding the usage of different software technologies. So they are skilled they are able to do software to make the very complicated software and they a lot they know very lot of lot of things regarding how to develop software.*
>
> *I think that you are better prepared than people University for some things, but specific programming. They are really good at programming, for example. And from the university they are not so good with that, at least until they get experience with the technology.*

Other areas that were seen as strengths were the ability to interact with clients and customers on a technical level: “In a technical level, they can interact totally with the clients and customers” (Focus Group 2).

3) Ideas for innovative ways of addressing these gaps

Several ideas were put forward from the companies in terms of how the gaps mentioned above could be addressed. These included allowing the graduates to undertake work placements during their studies in different companies in order to get some experience of how to apply their skills in the workplace. Additionally, an increase in project-based learning opportunities in their university studies would allow the graduates to apply their theoretical knowledge and use problem solving to adapt and apply their skills. Finally, getting colleagues from the companies to deliver or practically apply certain bits of content within the university course was suggested as a way in which the theory-practice gap could be addressed. This was nicely illustrated in Focus Group 9:

> That's why I think it's fundamental to have seminars from a company, I call them time to time, I have seminars from companies time to time. And I think that when they repeat what we tell them from the academic point of view, it becomes more convincing in our like, I don't know, two weeks ago at a company talking about software testing, and what they do in training and in the company on software testing. And they explain all the theory, or basic theory of software testing, and why it's important to have a specification, why it's important to have traceability why it's important to have requirements and all the other things. And this is things that indeed, we tell them in our courses. But I believe that when a same message comes from the company, it strongly reinforces what is the message coming from the academia, because otherwise academia can be seen, like theoretical things that is not used in the company.

Therefore, bringing experiences from the working environment into the university context would help students to apply some of their theoretical knowledge to practise. Furthermore, from the focus groups some suggested to involve CS students and graduates in project-based learning during the curriculum, that would involve problem solving and applying theoretical knowledge in practice. Finally, some participants suggested that it would be useful to provide a stronger link with practice by involving industry partners into curriculum design and implementation. This would allow programmes to keep on par with latest developments in both theoretical and practical industry CS.

4) Areas that companies work with their graduates to develop

The areas mentioned as those in which companies would work with their graduates to develop within the workplace setting would be developing their confidence in their knowledge and abilities.

Co-funded by the
Erasmus+ Programme
of the European Union

This would include independent learning and thinking about way in which they could put new ideas forward.

> *But for example, when they have to plan a project, and I know that this is the responsibility of another above them, but in a certain level, when you ask them for an opinion, it's like suddenly "you are asking me", I don't know, what do you think? No, it's like, maybe what the problem is like a lack of confidence in their knowledge (Focus Group 2)*

> *[Graduates] to understand that they, they can impact the company, they can impact the actual project they work on, no matter how much experience they have, because they might have different views, they might have different opinions, which might not be right. But it could be right. Sometimes it could be better than comparing to somebody who is much, much more experienced. And it's always to have, like, fresh pair of eyes, and somebody who's actually hasn't been biased by things ten more experienced people have been working on. So they have like a fresh a fresh view on things, which I think should be encouraged. (Focus Group 6).*

Mentorship support from other colleagues was also mentioned as a means in which both graduates and employees could learn from one another and provide a supportive development environment.

> *So they should really get like somebody who will mentor them and lead them. I think leadership is playing a crucial role here, in this process of integration of graduates into the company (Focus Group 6).*

> *...when we are talking that which kind of training they receive during the joining of the company, one module is also like you mentioned the mentoring, so, every new guy is the mentor (Focus Group 10)*

More standardised forms of development for CS graduates would also include the knowledge of IT processes that the company uses and inductions into the company culture and ways of working.

In summary, the participants we interviewed saw as the main gaps that graduates have when coming into their company or business from university a lack of soft skills such as communication and teamwork. Similarly, it would seem that they often struggle to apply their detailed theoretical knowledge to different practical contexts. The strengths they supply to the company mainly include knowledge of programming and interacting with clients and customers on a technical level. There were various suggestions in ways that these gaps could be addressed which include company placements, guest speakers and project-based learning. Areas that companies feel that they could work with their graduates to develop include areas such as confidence in their own knowledge/abilities, mentoring support from another colleague.

Gaps Computer Science graduates have when coming into the company from university

- Soft skills – e.g. communication, teamwork, organisation, basic project management
- How to apply their technical skills in different projects or move from the theoretical level into the practical application
- Too focused on particular technologies or software rather than having a broad application or can apply the technology in different contexts or projects
- Ready to experiment with their learning, challenge and ask critical questions – know when to ask for help
- Lacking the passion, attitude or motivation

Co-funded by the
Erasmus+ Programme
of the European Union

Strengths that Computer Science graduates bring from university into the company

- Programming and in-depth technical knowledge
- Some knowledge, competences and skills of new software that others in the company might not have
- Interacting on a technical level with clients and customers

Ideas for innovative ways of addressing these gaps

- Work placements in companies
- Project based learning that involves problem solving, applying theoretical knowledge
- Getting people from the company to deliver or practically apply certain bits of content

Areas that companies work with their graduates to develop

- Developing confidence in their knowledge and abilities – i.e. willing to put new ideas forward
- Mentorship support so that they can learn from other colleagues
- Knowledge, skills and competences of the IT processes that the company uses
- The company culture and ways of working

Co-funded by the Erasmus+ Programme of the European Union

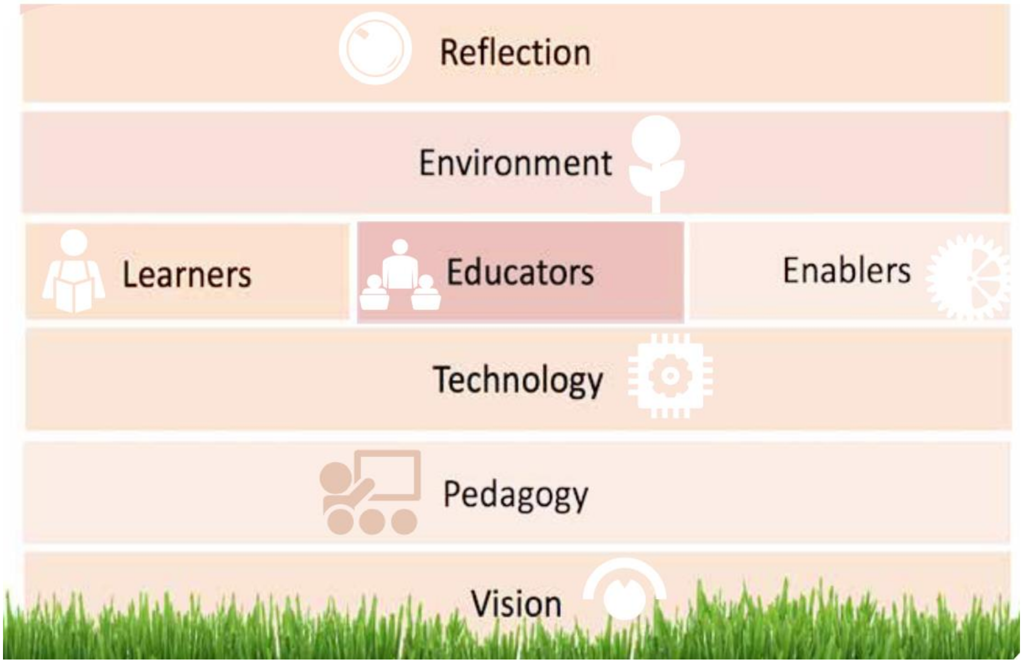
3. Teachers' skills and competencies

In the majority of cases, the literature dealing with innovations in CS pedagogy, and the literature dealing with educator training do not overlap (Bonfield et al., 2020; Garousi, Giray, Tüzün, Catal, & Felderer, 2019; Hussin, 2018; Jisc, 2019). In order to identify the skills and competencies required by educators in order to implement innovations in CS education, two approaches were taken. The first was to draw on research carried out by the Technology-Enhanced Learning Programme (Scanlon et al., 2013), which used desk research, interviews with experts, and case studies from around the world to identify approaches to developing and embedding educational innovations successfully. The second approach used the SLR (see section 4.1) to identify skills and competencies that were highlighted by authors when describing innovative practices.

Moving beyond prototypes

In order for CS educators to make use of innovative approaches, they need to be aware of the building blocks of a successful TEL innovation (Scanlon et al., 2013), as shown in Figure 1.

Figure 1 Building blocks of a successful innovation



Source: Scanlon et al. (2013)

As argued by European Commission/EACEA/Eurydice (2018, p. 16), "[w]hile innovation and digital technologies offer new possibilities for improving teaching and learning, Europe is slow to make the best use of them". The foundation of any successful innovation that uses technology to enhance learning is the vision that underpins it. The focus is initially not on the innovation, but on what it is intended to achieve. The vision describes an achievable and desirable end state. Examples might be: 'Students will learn quantum computing successfully' or 'Students will have authentic experiences that prepare them for future careers'. Starting with a vision gives a firm basis for introducing an innovation. Rather than starting with the technology or pedagogy ('Let's try the flipped classroom approach'), educators need to start by thinking about what they want to achieve

Co-funded by the Erasmus+ Programme of the European Union

(‘Our students need more time for hands-on lab work. It looks as though the flipped classroom approach could help.’)

Pedagogy and technology offer ways of achieving the initial vision of innovative practice. These two elements are closely related, particularly in a technology-rich subject area such as CS. Education 4.0 innovations are likely to involve not only new devices or technical approaches but also new ways of teaching and learning. A recurring theme we identified in this report was the teacher as facilitator or moderator or consultant of learning (Borge, Ong, & Goggins, 2020; Gestwicki & McNely, 2016; Juárez, Aldeco-Pérez, & Velázquez, 2020) as opposed to a teacher controlling or being the centre of the learning process (Corritore & Love, 2020). Teachers could achieve that by being flexible (adopt to change) (Gestwicki & McNely, 2016), supportive, help students to develop ownership of learning (Corritore & Love, 2020), foster an environment where students take risks and share what they do not know about, and where failure is acceptable (Borge et al., 2020). This role was often discussed within a flipped classroom implementation (Corritore & Love, 2020) that could give control to students to study the teaching material at their own pace and contact the teacher to solve problems and discuss their learning. In such conditions, the teacher is monitoring a student's progress and facilitates understanding through discussions (Paschoal, Oliveira, Nakagawa, & Souza, 2019).

Another important consideration is the need to develop pedagogies that are open (Wiley & Hilton Iii, 2018) and inclusive. In particular with the rise of Open Educational Resources (OERs), Massive Open Online Courses (MOOCs), open source programming (Alasbali & Benatallah, 2015), and online learning communities both learners and educators have unprecedented opportunities to learn together on a local, national, and global level (Fisk, 2017; Rienties, Hampel, Scanlon, & Whitelock, 2021).

Just as important when developing an innovation that will last are the different groups of people involved. Educators and learners are key, but there are also enablers without whom any innovation cannot last long (Guan, Mou, & Jiang, 2020; Herodotou, Rienties, et al., 2020; Wakelam, Jefferies, Davey, & Sun, 2019). The managers and department heads who approve funding and infrastructure (Erickson, Hanna, & Walker, 2020; McMahon, 2017). The senior leadership teams who develop the policies that block or encourage innovations (Higher Education Commission, 2016). The technicians responsible for installing and upgrading software, who install, connect, and maintain hardware. The industries and employers who are sources of support or resistance (Herodotou, Naydenova, Boroowa, Gilmour, & Rienties, 2020; Rienties, 2014).

Learners, educators and enablers need to be engaged throughout the life of the innovation but their roles are likely to change as it moves from novelty to business as usual. Innovations that rely on novelty value are unlikely to sustain interest in the long term (Bond, Zawacki-Richter, & Nichols, 2019; Guan et al., 2020). Therefore, training and professional development are essential. As argued by European Commission (2015b, p. 1), “[i]nvestment in human capital is money well spent. Good education and training help promote sustained economic growth, as well as sustainable development”. Furthermore, the need for technical support may increase as the hardware and software age. Additional funding or changes to infrastructure may be required and should be planned for. If employers and industry are involved, contacts will need to be refreshed and renewed over time.

Educators must therefore be aware of the need to interact with different groups, identifying and addressing potential problems, discussing possible improvements and engaging the people who will be crucial to the success of the innovation. These conversations also provide opportunities to

Co-funded by the Erasmus+ Programme of the European Union

discuss the environment in which the proposed innovation will take place. For example, flipped classrooms rely on students having access to technology and the Internet when away from the lab or the lecture theatre (Blau & Shamir-Inbal, 2017; Fisk, 2017; Hew & Lo, 2018; Parejo et al., 2020; Paschoal et al., 2019). Simple issues can block innovative practice if they are not foreseen and dealt with in good time.

The final element of innovation, which is demonstrated in all the studies described below, is reflection (Foster, Siddle, Crowson, & Bonne, 2020; Wasson & Kirschner, 2020; Zawacki-Richter, Marín, Bond, & Gouverneur, 2019). Educators need to build in time to consider whether the innovation has supported progress towards the original vision. In some cases, the vision will need to be updated, or the way of achieving it may need to be changed. Only when all the building blocks of innovation are firmly can an innovation achieve its full potential and go on to be incorporated within standard practice.

Skills and competencies identified in the literature

Educators should be able to

- Encourage student collaboration (Aničić, Divjak, & Arbanas, 2017; Apiola, Lokkila, & Laakso, 2019; Charlton & Avramides, 2016; Troussas, Krouska, & Sgouropoulou, 2020)
- Support the development of students' soft skills such as communication (Apiola et al., 2019; Burrows & Borowczak, 2017; European Commission, 2015a; Mäkiö, Yablochnikov, Colombo, Mäkiö, & Harrison, 2020) and leadership (Alegre, Moreno, Dawson, Tanjong, & Kirshner, 2020; Garousi et al., 2019)
- Be aware of recent developments in learning and teaching (Alasbali & Benatallah, 2015; Corritore & Love, 2020; Frevert et al., 2018)
- Be flexible and supportive (Aničić et al., 2017)
- Be aware of ways of using data to support their teaching. For example, social network analysis can be used to make sense of relations within a group or network (Alsaif, Li, Soh, & Alraddady, 2019; Borge et al., 2020; Charlton & Avramides, 2016; Shi, Min, & Zhang, 2017)
- Develop technical (Garousi et al., 2019; Scatalon, Garcia, & Barbosa, 2020; Tlili, Essalmi, Jemni, & Kinshuk, 2017), teaching and management skills (Burrows & Borowczak, 2019; Corritore & Love, 2020; Gonçalves, von Wangenheim, Hauck, & Zanella, 2018; Liang & Chapa-Martell, 2018)
- Incorporate innovations oriented towards social justice (Winiecki & Salzman, 2019)
- Make use of new, cloud-based environments (Halili, 2019; Liang & Chapa-Martell, 2018; Park & Kim, 2020)
- Train, support and make use of teaching assistants (Dickson, Dragon, & Lee, 2017; Pivkina, 2016)
- Make students aware of ethical issues (Bielefeldt, Polmear, Swan, Knight, & Canney, 2017; Casañ, Alier, & Llorens, 2020; Winiecki & Salzman, 2019)
- Highlight security issues (European Commission, 2015b)
- Manage the time of themselves and their students (Aničić et al., 2017)
- Share experiences and materials
- Vet and recommend third-party resources and curriculum
- Implement and use external resources
- Review, and implement where appropriate, different forms of assessment (Goumopoulos, Nicopolitidis, Gavalas, & Kameas, 2017; Liang & Chapa-Martell, 2018; Medeiros, Ramalho, & Falcão, 2019; Salem, Damaj, Hamandi, & Zantout, 2020)

Co-funded by the Erasmus+ Programme of the European Union

4. Innovations in teaching and learning Computer Science

A range of systematic literature reviews (SLRs) on CS and innovative approaches to teaching and learning have been published in the last five years (e.g., Aničić et al., 2017; Garousi et al., 2019; Guan et al., 2020; Hundhausen, Olivares, & Carter, 2017; Medeiros et al., 2019; Scatalon et al., 2020). For example, Aničić et al. (2017, p. 192) conducted a meta-analysis of 155 papers from 1980-2014, with the main aim of "give insight into the current research on the education and career development of graduates in the field of ICT". A broad range of search terms were used, whereby the findings in terms of curriculum design and delivery indicate a need to adjust the curricula to the needs of industry. As argued by Aničić et al. (2017, p. 194) "the literature indicates a need for innovative approaches in curriculum design and delivery, such as designing competency-based programs that are not restrained by the traditional semester seat-time model, providing flexible curriculum and minimizing the time spent in the classroom, or offering courses on not only how to manage innovation, but also on how to innovate". In terms of teaching methods that could help to encourage graduate employability include learning by doing, learning from mistakes, team-work and collaborative learning. Furthermore, a job-oriented experiment course system, problem- or project-based learning and work-integrated learning to develop a wide range of desired skills, competences and knowledge are encouraged, which link well with some of the Education 4.0 concepts.

Garousi et al. (2019) specifically focussed on how software engineering education was aligned with industrial needs. Using a SLR of 34 papers in period 1995-2018 they identified eight research questions, whereby two are highly relevant to our project (What curriculum models (bodies of knowledge) have been used to design the studies?; What educational recommendations are provided in each study?). Garousi et al. (2019, p. 77) indicated that the "qualitative coding provided four themes for the educational recommendations provided in the papers: (1) Need for more emphasis on soft skills (20 papers), (2) Need for active Infrastructure as Code (IAC) (3 papers), (3) Less emphasis on certain topics (2 papers), and (4) Other recommendations (7 papers)." In order to encourage development of soft-skills, Garousi et al. (2019) encouraged educators to use real-life projects, implement industry-academia collaboration in the design of education, and to anticipate future trends, while also preparing students to deal with them.

In a review of 89 papers on introductory programming in HE in the period 2010-2016, Medeiros et al. (2019) explored how prior skills influence learning to programme, and what challenges both students and teachers experience in learning to programme. A clearer need is identified to understand problem solving, prior knowledge, and relevant to our project, better specific tools and methods for problem formulation and solution expression. In a broader review of Artificial Intelligence (AI) and Education of 425 papers from 2000-2019, Guan et al. (2020) showed that a range of approaches have been adopted to embed technology in HE, although few have stood the test of time.

Based upon 195 empirical papers Scatalon et al. (2020) provided an overview of the practices that have been used to integrate software testing into programming education. The study showed that testing practices in programming assignments involved students to different extents: analysing test results from submission tools, working with instructor-provided tests, using support mechanisms to design tests (e.g. plugins where students insert inputs and expected outputs) and, finally, students writing their own tests. However, very few studies addressed how students learned testing concepts in programming courses.

Co-funded by the
Erasmus+ Programme
of the European Union

While these SLRs provide important and deep insights into how CS, computer programming, and AI have been used in a range of HE contexts, none of these studies specifically focus, mention, or include concepts of Education 4.0. In particular with the Covid19 Pandemic and the rapid shift to online education, it is essential to update our insights about how CS teachers are adopting innovative pedagogies and Education 4.0 approaches. Therefore, as part of IO1 deliverable we conducted an extensive Systematic Literature Review specifically focussed on how Education 4.0 approaches were used by teachers in Europe and across the globe in CS.

4.1. Systematic literature review

In order to investigate which innovations are being introduced in the field of CS, a systematic literature review was carried out, focusing on three research questions.

RQ1: Which pedagogic approaches are used to support the teaching of computer science to undergraduate and postgraduate students?

RQ2: Which of these approaches align with Education 4.0?

RQ3: What skills and competences do HE educators require in order to align their computer science teaching with Education 4.0?

Four research databases were searched: ScienceDirect, Wiley Online Library, Web of Science, and Scopus. These were chosen because of their ranking as academic research databases, and good coverage of relevant studies relevant for the review.

Initial inclusion criteria

- The item is included in one of the four research databases.
- The item is peer reviewed.
- The item is written in English.
- The item is accessible openly or through a university library.
- The item meets the search criteria.

The exact search terms for each database are set out in Appendix C. In summary, papers had to be published in English during the five-year period 2016–2020, keywords had to include Computer Science; undergraduate and/or postgraduate; as well as education, teaching and/or pedagogy. These search terms identified 231 publications.

Exclusion criteria

Publications identified using the search criteria were excluded if any of the following exclusion criteria applied:

- The focus is on primary and/or secondary education
- The focus is on a subject other than Computer Science
- The focus is on learners (e.g., their gender or expectations) rather than teaching

4.1.1. Education 4.0

For RQ2, as highlighted in on page 6 there are a range of conceptualisations of Education 4.0 (Fisk, 2017; Halili, 2019; Jisc, 2019; Puncreobutr, 2016; Salmon, 2019; Suhaimi, 2019; Wallner & Wagner, 2016). In this project we defined Education 4.0 in the following manner: “Education 4.0 employs an approach to learning and teaching that emphasises the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment." Building on the work of Fisk (2017) we adopted the nine key characteristics identified by Hussin (2018) of Education 4.0, which include the characteristic that 1) learning can be taken place anytime anywhere; 2) learning will be personalized to individual students; 3) students have a choice in determining how they want to learn; 4) students will be exposed to more project-based learning; 5) students will be exposed to more hands-on learning through field experience (e.g., internships, mentoring projects, collaborative projects); 6) students will be exposed to data interpretation in which they are required to apply their theoretical knowledge to numbers and use their reasoning skills to make inferences based on logic and trends from given sets of data; 7) students will be assessed differently and the conventional platforms to assess students may become irrelevant or insufficient; 8) students' opinion will be considered in designing and updating the curriculum; 9) students will become more independent in their own learning. For the analysis we used both the individual scores as well as the aggregate score. If a study did not indicate any of the Hussin (2018) concepts, we removed the study from further analysis.

4.1.2. Skills of teachers to teach in Education 4.0

For RQ3 coders indicated whether (or not) any specific skills of teachers to support the teaching of CS to students was mentioned. If yes, coders could use a follow-up open text box to add any description and conceptualisation of the teachers skills, afterwards we recoded and aggregated the skills.

4.2. Coding process

Applying the inclusion and exclusion criteria as indicated in Appendix C, 231 papers were identified. In Phase 0, Author Rebecca Ferguson manually screened the abstracts to check whether the respective papers should be included or excluded based upon the above criteria. Subsequently, 75 papers were excluded.

In Phase 1, following a one hour online training and discussion of the online coding scheme of four variables, 156 papers were read in-depth by 18 members of the TEACH4EDU4 project and based upon three inclusion criteria (i.e., 1) is it an "innovative" application in a CS course; 2) Does it use technology or pedagogy in an innovative way; 3) Is the innovation evaluated, if so how?). By including experts from CS and educational technology from six EU countries we aimed to develop an inclusive multi-disciplinary team of coders to analyse the literature. On average the members coded 8.26 papers (range: 3-11), whereby 68 papers were included for subsequent analysis. All papers were annotated and uploaded in Google drive for a second round of coding.

In Phase 2, 17 members of the TEACH4EDU4 project participated in a follow-up one hour online training and discussion of the online coding scheme of 20 variables. Coders were randomly allocated a new set of papers to code in comparison to their initial coding in Phase 1. The 17 members coded on average 4.25 papers (range: 2-10) based upon the coding scheme developed from the above research questions. Afterwards, the first coders from Phase 1 checked the codings from the second coders in Phase 2, discussed any differences (i.e., 17 times (1%)), and agreed on the final coding. A random sample of 15 papers was double coded and indicated reliable coding. Removing any paper which did not meet our definition of Education 4.0 or received a 0 score on Hussin (2018), we ended up with 66 papers.

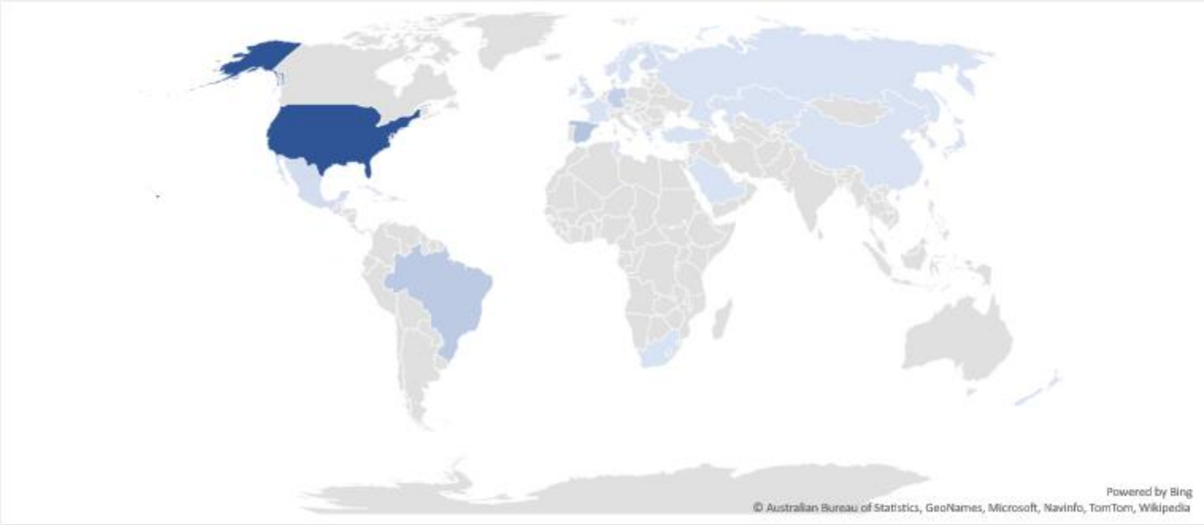
Co-funded by the Erasmus+ Programme of the European Union

4.3. Data analyses

The vast majority of studies included referred to undergraduate CS students (79%), followed by a mix of undergraduate and post-graduate students. Five studies did not explicitly mention the specific student population, and one included teachers only. 36% of studies were from the USA, followed by Spain (9%), Brazil (8%), and Germany (6%). As indicated in Figure 2 using the GLOBE geo-cultural regions classification by House, Hanges, Javidan, Dorfman, and Gupta (2004), 47% of studies were conducted by Anglo-Saxon countries, followed by Latin American countries and Latin European (each 12%), Eastern European (8%), Scandinavian and German countries (each 6%), Confucian Asian (5%) and Middle Eastern countries (3%). No studies were identified from African or Southern Asian countries.

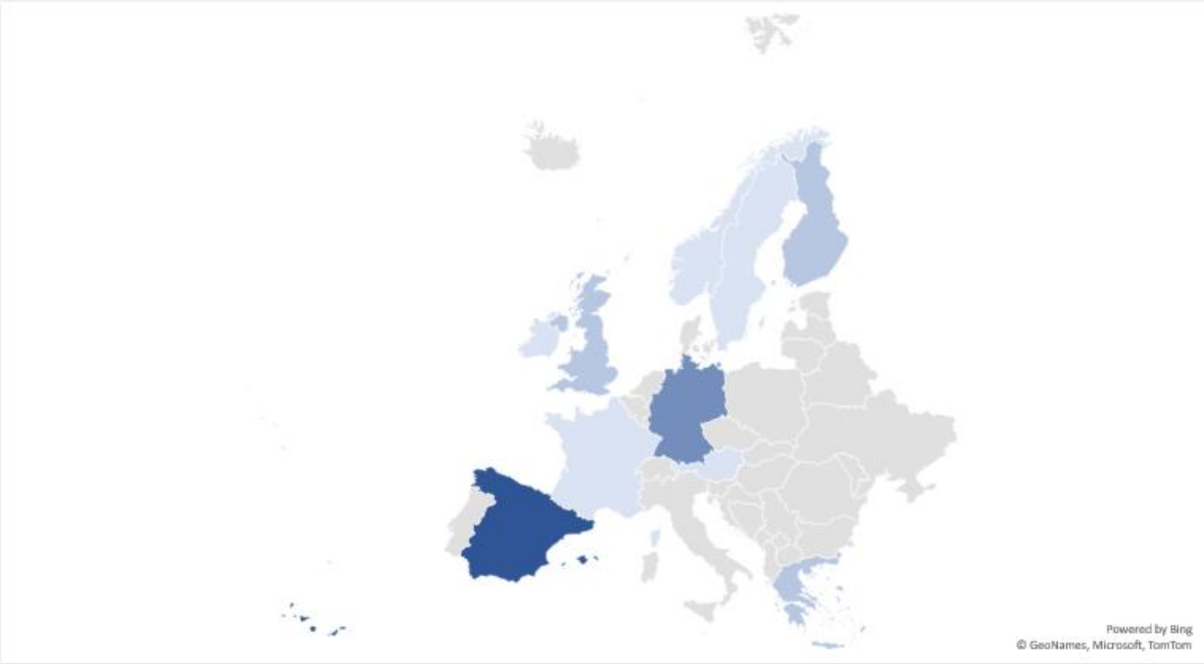
Given the focus of Teach 4.0, in total 18 (27%) studies were identified from the EU27, as indicated in Figure 3, mainly from Spain (6), Germany (4), Finland (2), Greece (2), Austria (1), France (1), Ireland (1), as well as from Norway (1) and the UK (2). Obviously, this does not mean that in other EU countries no research on Education 4.0 in CS is conducted, and the findings might be different if other search strings are used. Using ANOVA analyses, no significant differences were found on our key variables and GLOBE. Furthermore, no significant differences were found on our key variables and EU27, indicating no substantial differences in practices in CS based upon national/geo-cultural regions.

Figure 2 Geographical location of studies (based upon frequency of included studies)



Co-funded by the Erasmus+ Programme of the European Union

Figure 3 Location of studies in Europe (based upon frequency of included studies)

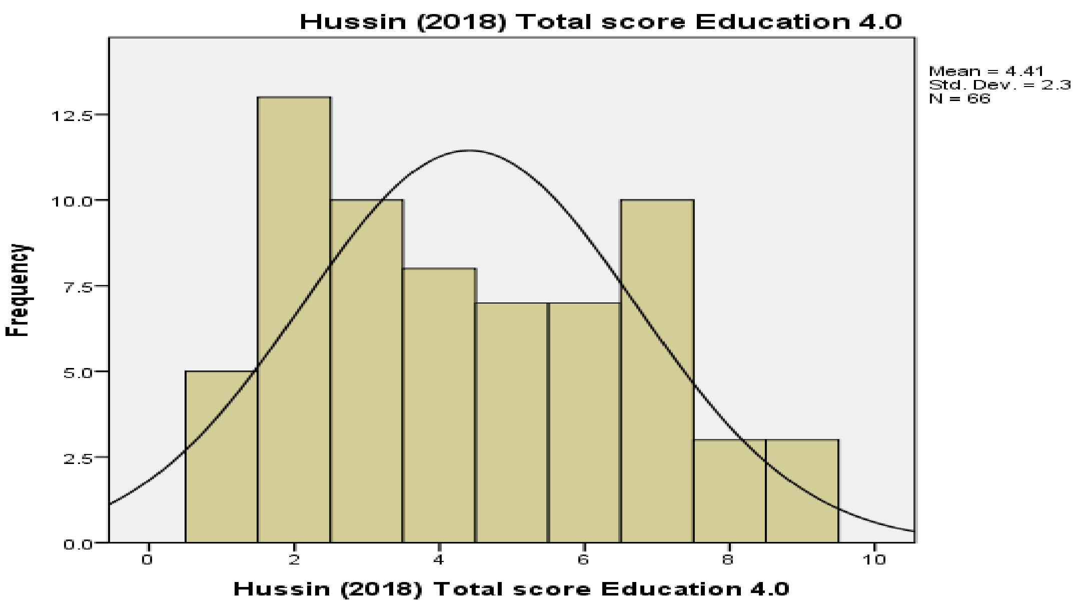


5. Innovative teaching methods

In terms of RQ1 and RQ2, of the 66 articles selected perhaps surprisingly none of the articles explicitly mentions "Education 4.0". In part this could be a result of the relatively recent conceptualisation of Education 4.0, and in part this could be due to the lack of adoption of the term Education 4.0 in the specific discipline of CS. Based upon our broad definition of Education 4.0 (i.e., "Education 4.0 employs an approach to learning and teaching that emphasises the development of skills and competences necessary in a modern workplace using up-to-date technology. The skills and competences developed may relate directly to the technology, or they may be the softer skills (such as team-working and creativity) that are needed to work effectively in such an environment)") in total 54 articles (80%) were considered to fit under this definition. Furthermore, 66 articles included at least one Hussin (2018) Education 4.0 characteristic.

As indicated in Figure 4, on average the 66 articles included 4.41 out of nine Education 4.0 characteristics of Hussin (2018), with a substantial variation (SD = 2.30). There seemed to be two peaks in Figure 4, whereby 35% of articles only had 2-3 Education 4.0 characteristics, while another peak at 7 Education 4.0 characteristics was present.

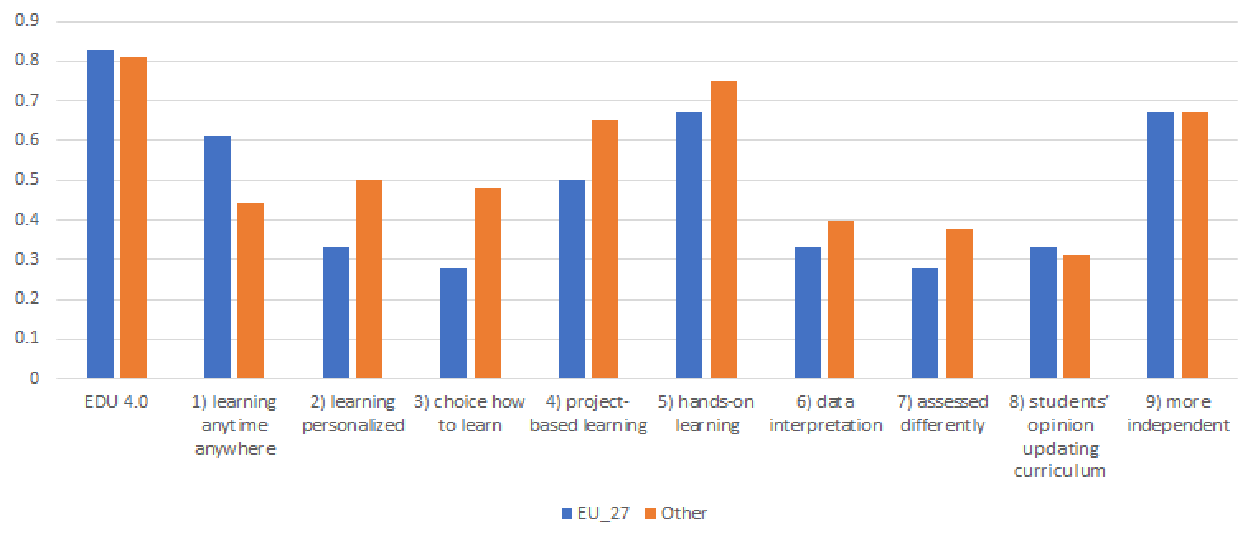
Figure 4 Histogram of Education 4.0 (Hussin, 2018)



As indicated in Figure 5, the most common Education 4.0 characteristic was "5) students will be exposed to more hands-on learning through field experience" (73%), followed by "9) students will become more independent in their own learning" (67%), "4) students will be exposed to more project-based learning" (61%). Around half of the articles included the characteristic that "1) learning can be taken place anytime anywhere", while around a third of articles included "7) students will be assessed differently and the conventional platforms to assess students may become irrelevant or insufficient" (35%) and "8) students' opinion will be considered in designing and updating the curriculum" (32%). Follow-up ANOVA analysis showed no significant differences in the Hussin (2018) characteristics when comparing EU27 versus other countries.

Co-funded by the Erasmus+ Programme of the European Union

Figure 5 Education 4.0 of EU and other countries



We found a moderate strong correlation (rho = .429, p < .01) between our EDU 4.0 definition and the aggregate Hussin (2018), with the strongest correlation on the individual Hussin (2018) characteristic of 5) “students will be exposed to more hands-on learning through field experience”. The individual Hussin (2018) characteristics were not all directly and significantly correlated. For example, Hussin 1 was positively correlated with Hussin 6, Hussin 7 and Hussin 8, but not to the other characteristics.

Table 4 Pattern structure of Factor analysis Hussin (2018)

	1	2
Hussin 2) learning will be personalized to individual students	0.769	
Hussin 1) learning can be taken place anytime anywhere	0.644	
Hussin 6) students will be exposed to data interpretation in which they are required to apply their theoretical knowledge to numbers and use their reasoning skills to make inferences based on logic and trends from given sets of data	0.618	
Hussin 9) students will become more independent in their own learning	0.617	
Hussin 3) students have a choice in determining how they want to learn	0.589	
Hussin 8) students' opinion will be considered in designing and updating the curriculum		
Hussin 4) students will be exposed to more project-based learning		0.883
Hussin 5) students will be exposed to more hands-on learning through field experience (e.g., internships, mentoring projects, collaborative projects)		0.836
Hussin 7) students will be assessed differently and the conventional platforms to assess students may become irrelevant or insufficient		0.455

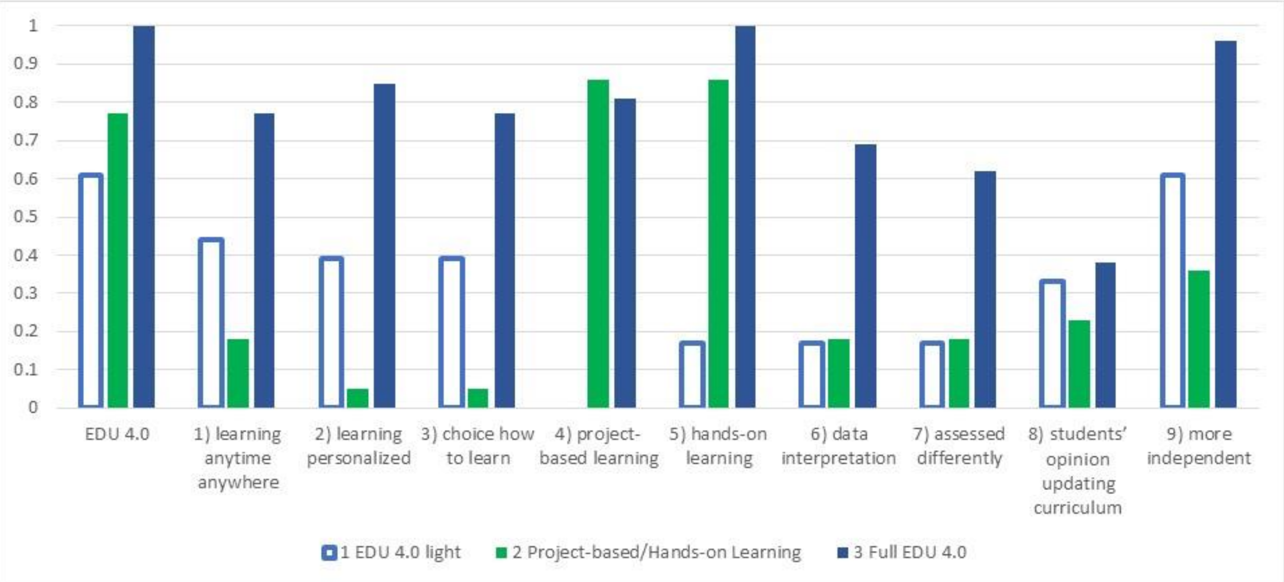
Extraction Method: Principal Component Analysis.
Rotation Method: Oblimin with Kaiser Normalization.

Co-funded by the Erasmus+ Programme of the European Union

Therefore, an exploratory factor analysis (principal component analysis) with direct oblimin rotation conducted on the data collected immediately indicated the existence of two factors, with item loads of .45 and more. The first component had an eigenvalue of 2.62 (corresponding to 29% of the explained variance), the second component had an eigenvalue of 1.45 (corresponding to 16% of the explained variance). As indicated in Table 4, Hussin 2, Hussin 1, Hussin 6, Hussin 9, and Hussin 3 items tapping the loaded on the first factor, which we will label as “intention self-learning”. Hussin 4, Hussin 5 and Hussin 7 items loaded on the second factor, which we will label as “intention project-based/hands-on learning”. Hussin 8 did not load on any factor. Respective Cronbach Alphas for these two factors were .68 and .62, indicating reasonable reliability.

Finally, a follow-up analysis using K-means cluster techniques indicated a three-cluster solution across the 66 papers. As illustrated in Figure 6, there seemed to be 3 clusters of papers, which we label as 1) *EDU 4.0 light* (n = 18), 2) *project-based/hands-on learning* (n = 22), and 3) *full EDU 4.0* (n = 26). With the notable exception of Hussin 8) “students opinion will be considered in designing and updating the curriculum”, using ANOVAs all the Hussin characteristics were significantly different between the three clusters with large effect sizes. In other words, there appeared to be three distinct innovative pedagogical practices present in published work on CS in the last four years. No significant differences were found based upon the country of study, or GLOBE, although relatively fewer full EDU 4.0 papers (28%) were published in the EU27 relative to other countries, whereas 44% of papers were clustered under full EDU 4.0.

Figure 6 Cluster analysis of EDU 4.0 (3 cluster solution)

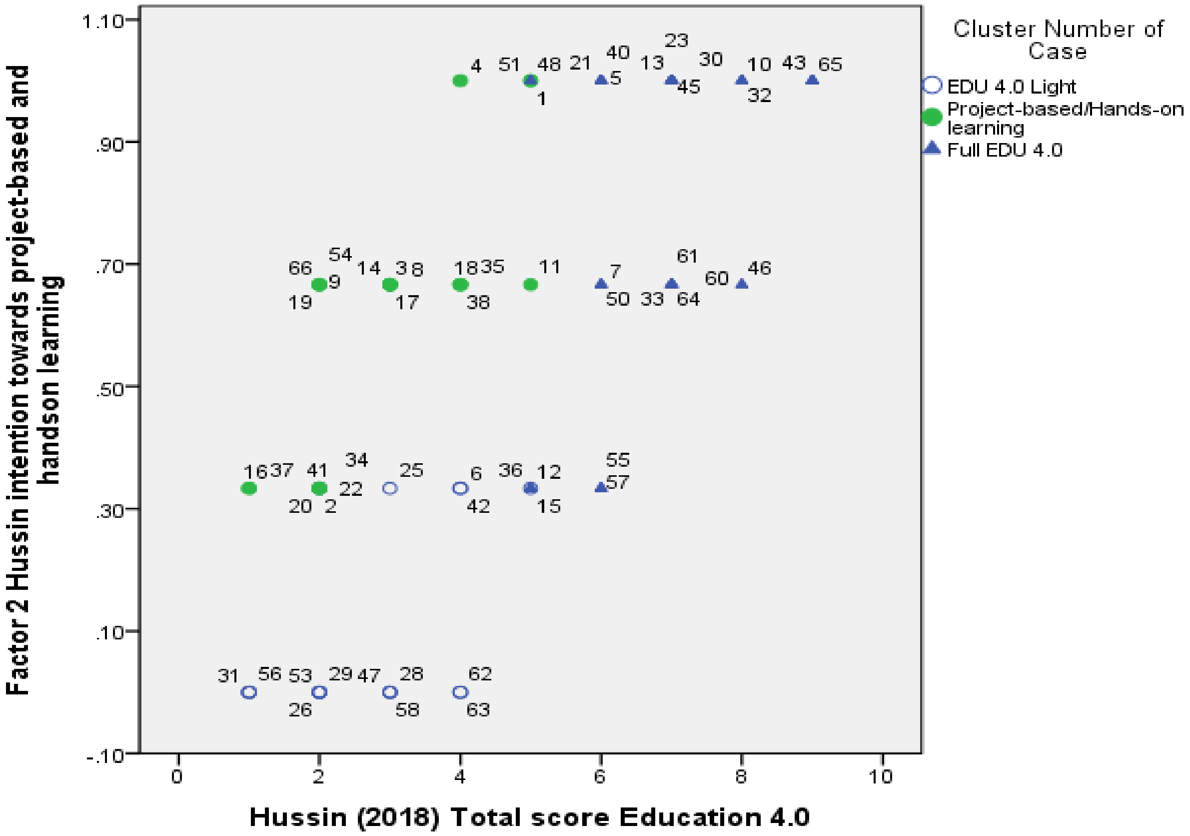


As indicated in Figure 7 EDU 4.0 light studies (blue circles) mostly had relatively low Hussin (2018) total scores, and often did not include project-based activities. Therefore, most of these studies in Figure 7 were positioned in the bottom left. In contrast, while some project-based/hands-on learning studies (green circles) had relatively low Hussin (2018) scores, in particular for personalised learning and choice how to learn, they had a strong focus on project-based and hands-on learning. Therefore, many of these studies are positioned in the middle to top-left quadrant of Figure 7. Finally, studies which were classified as full EDU 4.0 studies (blue triangles) were mostly positioned in the middle and right of Figure 7, indicating these studies used more and even all EDU 4.0 characteristics in their designs. Note that the numbers in Figure 7 refer to the

Co-funded by the Erasmus+ Programme of the European Union

studies discussed below, whereby initially the papers were ordered alphabetically based upon the last name of the first author.

Figure 7 Scatterplot of cluster analysis results of 66 papers



5.1. EDU 4.0 light

As indicated in Figure 6 and Figure 7, in EDU 4.0 light studies teachers mostly focussed on more independent learning (61%), learning anytime anywhere (44%), personalised learning (39%), and choice over how to learn (39%), but with limited hands-on learning (17%) and no project-based (0%). A detailed description of each study can be found in Appendix A.1, with the overview of how each study used Education 4.0 in Table 5.

Table 5 EDU 4.0 Light studies

Authors	1) learning anytime anywhere	2) learning personalised	3) choice how to learn	4) project-based learning	5) hands-on learning	6) data inter-pretation	7) assessed differently	8) students' opinion updating curriculum	9) more independent	Country
Apiola et al. (2019)	Y		Y		Y				Y	Finland
Burrows and Borowczak (2019)	Y	Y		Y				Y	Y	USA
Degener, Haak, Gold-Veerkamp, and Abke (2019)	Y								Y	Germany
Dickson et al. (2017)		Y	Y		Y					USA

Dondio and Shaheen (2019)								Y	Y	Ireland
Fisher, Rader, and Camp (2016)	Y	Y						Y		USA
Frevert et al. (2018)								Y	Y	USA
Giacaman and De Ruvo (2018)			Y							New Zealand
Hosseini, Hartt, and Mostafapour (2019)							Y		Y	USA, Wales, Canada
Parejo et al. (2020)	Y						Y			Spain
Park and Kim (2020)	Y	Y					Y		Y	Korea
Pilkington (2017)	Y					Y			Y	South Africa
Scatalon et al. (2020)		Y								Brazil
Schäfer (2019)	Y								Y	Germany
Shi et al. (2017)			Y							China
W. Silva, Steinmacher, and Conte (2019)			Y					Y	Y	Brazil
Tyler and Abdrakhmanova (2016)		Y	Y			Y		Y		Kazakhstan

For example, Degener et al. (2019) integrated LEGO MindStorms EV3 robots within lessons teaching the programming language ANSI-C. The intention was to make practical programming lessons more tangible and closer to the future field of work for CS and engineering students. Although the EV3 contributed to learning success and fun during the lessons, students were not able to program the EV3 outside laboratory opening hours and practical lessons. To solve this problem, a simulation was provided to make the programming task time- and location-independent. In another example of EDU 4.0 light, Schäfer (2019) introduced the concept of a modern C ++ course for students of electrical engineering and CS based on a flipped classroom and with pleasant Internet of Things (IoT) hardware. The main goal of the new course was to reduce lecture time in favour of practical learning of students through programming. Schäfer (2019) used a flipped classroom (inverted in his terminology) to adapt the pace of teaching to the individual needs of the students and to enable students to study anytime and anywhere. In his conception of the course, Schäfer (2019) recommended replacing theoretical lectures with discussion meetings between teachers and students, and most of the time defines students' practical work on a programming project.

5.2. Project-based/hands-on learning

The second cluster that we labelled as project-based/hands-on learning had a strong focus on Hussin (2018) project-based (86%) and hands-on learning (86%), with relatively limited focus on choice how to learn (5%), personalised learning (5%), and learning anytime anywhere (18%). A detailed description of each study can be found in Appendix A.2, with the overview of how each study used Education 4.0 in Table 6.

.

Co-funded by the
Erasmus+ Programme
of the European Union

Table 6 Project-based/hands-on learning studies

Authors	1) learning anytime anywhere	2) learning personalised	3) choice how to learn	4) project-based learning	5) hands-on learning	6) data inter-pretation	7) assessed differently	8) students' opinion updating curriculum	9) more independent	Country
Aghaee and Keller (2016)	Y			Y	Y		Y		Y	Sweden
Alasbali and Benatallah (2015)				Y					Y	Global
Alegre et al. (2020)			Y	Y	Y					USA
Alomari, Ramasamy, Kiper, and Potvin (2020)	Y			Y	Y		Y			USA
Berikan and Özdemir (2020)				Y	Y	Y				Turkey
Bielefeldt et al. (2017)				Y	Y					USA
Borowczak and Burrows (2017)				Y	Y	Y		Y	Y	USA
Burrows and Borowczak (2017)					Y		Y	Y		USA
Bushmeleva and Baklashova (2017)					Y					Russia
Caceffo et al. (2018)				Y	Y				Y	Brazil
Carrascal, del Barrio, and Botella (2021)				Y	Y	Y		Y		Spain
Casañ et al. (2020)				Y	Y					Spain
Chamberlin, Hussey, Klimkowski, Moody, and Morrell (2017)	Y				Y					USA
Cobos and Roger (2019)				Y				Y		Spain
Fagerholm et al. (2018)				Y	Y				Y	Finland
Juárez et al. (2020)		Y		Y	Y				Y	Mexico
Lewis and Lacher (2020)				Y						USA
Liang and Chapa-Martell (2018)	Y			Y	Y				Y	Japan
A. Llorens, Berbegal-Mirabent, and Llinas-Audet (2016)				Y	Y					Spain
Mäkiö et al. (2020)				Y	Y				Y	UK
Santos et al. (2018).				Y	Y	Y	Y	Y		Austria, Czech Republic, Slovak Republic, UK
Seyam and McCrickard (2016)				Y	Y					USA

For example, Caceffo et al. (2018) assessed the benefits of the use of technology and active learning practices (i.e., Project-Based Learning and Peer Instruction) in the classroom with 25 students for undergraduates to contribute to a more effective and efficient learning environment. Alomari et al. (2020) used an innovative platform to improve the knowledge of 51 CS and Software

Co-funded by the Erasmus+ Programme of the European Union

Engineering students about software testing by providing a set of learning objects and tutorials categorised by difficulty level. This evolved into a collaborative learning environment that included social networking features such as the ability to award virtual points for student social interaction about testing (Alomari et al., 2020). While these studies show strong practical hands-on learning cases for the students, often studies in this cluster do not have the option to choose the way to learn or personalise their learning experience.

5.3. Full EDU 4.0

The third and final cluster which we labelled as the full EDU 4.0 version was strongly focussed on hands-on learning (100%), becoming more independent (96%), personalised learning (85%), learning anytime anywhere (77%) and choice how to learn (77%). The lowest Hussin (2018) characteristic was including students' opinion when updating the curriculum (38%), although this was substantially higher than the other two clusters. Table 7 provides an overview of how each study used Education 4.0, whereby a detailed description of each study can be found in Appendix A.3.

Table 7 Full EDU 4.0 studies

Authors	1) learning anytime anywhere	2) learning personalised	3) choice how to learn	4) project-based learning	5) hands-on learning	6) data inter-pretation	7) assessed differently	8) students' opinion updating curriculum	9) more independent	Country
Alsaif et al. (2019)	Y		Y	Y	Y		Y		Y	Saudi Arabia
Behnke, Kos, and Bennett (2016)	Y		Y	Y	Y	Y			Y	USA
Borge et al. (2020)	Y	Y		Y	Y	Y	Y	Y	Y	USA
Broisin, Venant, and Vidal (2017)	Y	Y	Y		Y				Y	France
Buffardi and Valdivia (2018)			Y	Y	Y	Y	Y	Y	Y	USA
Charlton and Avramides (2016)		Y	Y	Y	Y		Y		Y	UK
Corritore and Love (2020)	Y		Y	Y	Y		Y	Y	Y	USA
Gestwicki and McNely (2016)		Y	Y	Y	Y		Y	Y	Y	USA
Gonçalves et al. (2018).	Y	Y	Y	Y	Y	Y	Y		Y	Brazil
Goumopoulos et al. (2017)	Y	Y	Y	Y	Y	Y			Y	Greece
Knobelsdorf, Frede, Böhne, and Kreitz (2017)	Y	Y			Y	Y			Y	Germany
Munkvold (2017)		Y	Y	Y	Y		Y		Y	Norway
Paschoal et al. (2019)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Brazil
Pawelczak (2017)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Germany
Peng (2015)	Y	Y	Y	Y	Y	Y	Y			USA
Peteranetz, Flanigan, Shell, and Soh (2017)	Y	Y	Y	Y	Y	Y		Y	Y	USA
Pivkina (2016)		Y		Y	Y		Y		Y	Mexico
Ruiz, Serral Asensio, and Snoeck (2020)	Y	Y		Y	Y	Y	Y		Y	Cuba
Salem et al. (2020)	Y	Y		Y	Y	Y			Y	Lebanon

Co-funded by the Erasmus+ Programme of the European Union

Seyam, McCrickard, Niu, Esakia, and Kim (2016)	Y	Y	Y		Y	Y			Y	USA, Korea
P. A. Silva, Polo, and Crosby (2017)		Y	Y		Y	Y		Y	Y	USA
Tanaka, Ferreira da Silva, and Casanova (2019)	Y	Y		Y	Y	Y	Y		Y	USA
Tlili et al. (2017)	Y	Y	Y	Y	Y	Y			Y	Tunisia
Troussas et al. (2020)	Y	Y	Y		Y		Y	Y	Y	Greece
Winiecki and Salzman (2019)	Y	Y	Y	Y	Y	Y			Y	USA
Wood et al. (2018)	Y	Y	Y	Y	Y	Y	Y	Y	Y	USA

For example, Pivkina (2016) described the experience of using a undergraduate student as peer learning assistant (PLA), supporting 80 students in in three different undergraduate computer science courses with 20 students in each at New Mexico State University; PLAs were holding office hours, helping with the labs and tutorials and facilitating student group work in class, therefore were practicing tutoring their peer students. The impact of the experience was measured by comparing student interactions between PLAs and regular university teaching assistants, showing a preference of students to interact with their PLAs. While that research offers a case for practical learning it does not offer any choice of learning anytime or anywhere, limiting the experience to the interaction with the PLAs.

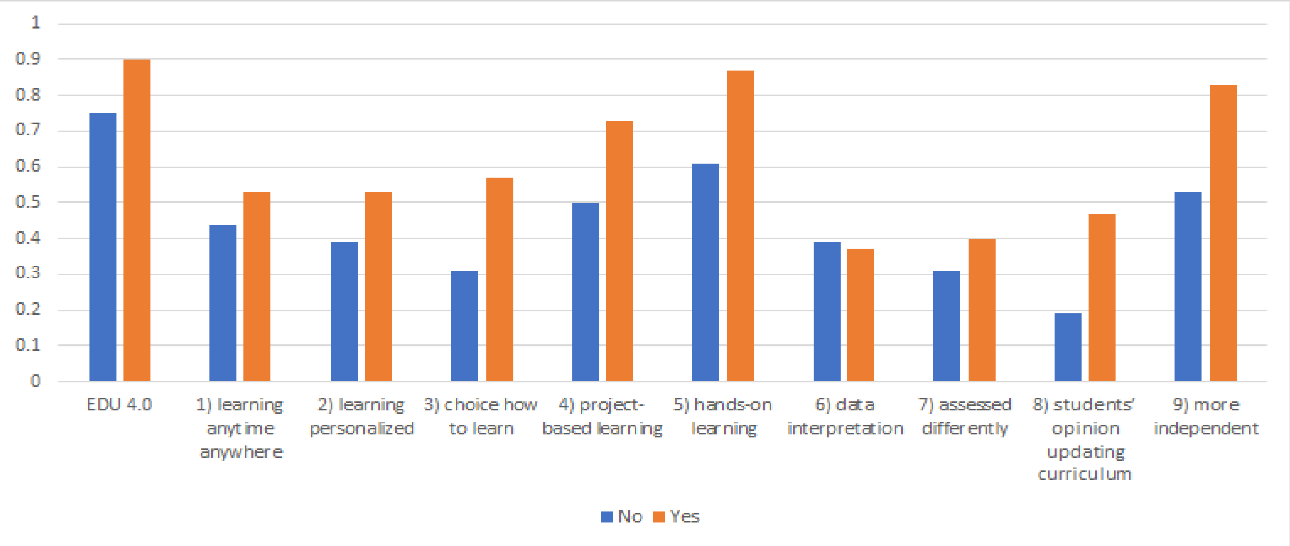
A similar case can be found in Seyam et al. (2016) who evaluated whether Pair Programming (an agile software development practice which supports time work in software development companies) would help 53 students during five sessions in a mobile development course for undergraduates in Virginia Tech to better understand mobile programming. For the evaluation of the experience, observations and questionnaires were used to show a rich experience where programming for mobile devices cannot be seen as merely writing code. This study shows a practical hands-on learning case for the students in a real-life software development environment, but students do not have the option to choose the way to learn or personalise their learning experience.

Co-funded by the Erasmus+ Programme of the European Union

6. Skills for teachers in Computer Science to deliver EDU 4.0

Nearly half of the papers (n = 30) reviewed made an explicit reference to skills and competences HE educators should have or develop to align their computer science teaching to the broader definition of Education 4.0. As indicated in Figure 8, papers that referred to skills and competences of teachers on average had higher scores on nearly all the Hussin (2018) characteristics and EDU 4.0, with the notable exception of 6) data interpretation. A follow-up ANOVA analysis indicated significant differences between articles that did and did not mention skills of teachers on 3) choice how to learn ($F = 4.758$, $p < .05$); 5) hands-on learning ($F = 5.689$, $p < .05$); 8) including students' opinion in updating curriculum ($F = 5.922$, $p < .05$); and 9) more independent learning ($F = 7.422$, $p < .01$).

Figure 8 Mentioning of skills and competences of teachers and Education 4.0



Furthermore, the aggregate Hussin (2018) score was substantially higher in articles that mentioned skills and competences of teachers ($M = 5.30$, $SD = 2.20$, $F = 9.304$, $p < .01$) relative to those who did not mention those skills ($M = 3.67$, $SD = 2.14$). Although no significant differences across the three clusters were found between articles that did or did not mention skills and competences of teachers, 53% of articles who did were part of the full EDU 4.0 cluster, while only 28% of articles who did not were part of that cluster. In other words, those articles who explicitly referred to skills and competences for teachers seemed to be more explicit and innovative in terms of pedagogies and EDU 4.0 elements. An alternative explanation could be that the authors of more innovative pedagogical approaches and full EDU 4.0 modes provided more narratives about how teachers could effectively support these innovative approaches.

The articles that made an explicit reference to skills and competences of HE educators discussed the issue of educators' skills in relation to the implementation and assessment of an innovative learning intervention which was the main focus of the article. A reference or discussion about skills and competences was often presented as an implication of the proposed study rather than being examined as the starting point of a given article. This could be explained by the fact that innovative teaching approaches or interventions are more likely to require teachers to develop new skills and competences, and thus such a discussion was seen as closely relevant. This observation could explain some of the insights of the quantitative analysis, in particular the observed higher scores on Hussin (2018) characteristics and EDU 4.0 in papers where skills and competences of teachers are discussed.

In terms of the educators' skills discussed in these papers, some of them could be seen as generic such as the creation of student-centred environments, while others as more concrete such as the

Co-funded by the Erasmus+ Programme of the European Union

use of specific mobile games in teaching. A recurring theme we identified in this report was the teacher as facilitator or moderator or consultant of learning (Borge et al., 2020; Gestwicki & McNely, 2016; Juárez et al., 2020) as opposed to a teacher controlling or being the centre of the learning process (Corritore & Love, 2020). Teachers could achieve that by being flexible (adopt to change) (Gestwicki & McNely, 2016), supportive, help students to develop ownership of learning (Corritore & Love, 2020), foster an environment where students take risks and share what they do not know about, and where failure is acceptable (Borge et al., 2020).

This facilitative role was often discussed within a flipped classroom implementation (Corritore & Love, 2020) that could give control to students to study the teaching material at their own pace and contact the teacher to solve problems and discuss their learning. In such conditions, the teacher is monitoring a student's progress and facilitates understanding through discussions (Paschoal et al., 2019). A teacher as facilitator was also seen as the person strengthening communication, ethics, leadership, security, and software skills (Juárez et al., 2020). These conditions point to teachers as the agents in charge of developing student-centred learning environments (Troussas et al., 2020).

Teachers' skills and competences were also discussed in relation to the development of more specific expertise including the use of social network analysis techniques to understand social relationships when students are part of an online network or community (Borge et al., 2020), the use of a peer learning assistance approach, that is, having peers to hold office hours, help with labs and facilitate student group work, as they were shown to better support learning than teaching assistants (Pivkina, 2016), and the use of specific educational games (Buffardi & Valdivia, 2018; Troussas et al., 2020) and remote laboratories (Broisin et al., 2017) that could support CS education. In terms of game-based approaches to computer science, teachers should have skills to provide tailored and personalised feedback (Troussas et al., 2020) and assign students to game roles within a course management system (Buffardi & Valdivia, 2018).

Active learning practices require considerable time for preparation compared to the traditional lecture-based class (Caceffo et al., 2018) and this may be overwhelming for teachers. Therefore, support should be provided through, for example, teaching assistants, fellow teachers, or the reuse of existing activities to help teachers gradually develop the proposed skills and competences.

Co-funded by the Erasmus+ Programme of the European Union

7. European perspectives on EDU4.0

As indicated in Table 8, there were six studies from Europe that were classified as EDU 4.0 light. Five out of six studies aimed to develop independence and anytime/anywhere learning. At the same time, a mix of other Hussin (2018) activities were included in these six studies, although mostly just one or two characteristics. Seven studies from Europe were classified as project-based/hands-on studies, whereby all but one included both Hussin 4 and Hussin 5 in their designs. Perhaps surprisingly four out of seven studies in this category were from a single country, Spain. Finally, as indicated in Table 8 five studies were labelled as full EDU 4.0 studies where nearly all of the Hussin activities were included. Two studies from Germany and Greece were also included in this cluster. As previously highlighted in

Figure 5, except for Hussin 1 (anytime/anywhere) most of the European studies scored relatively lower on the Hussin (2018) activities. Nonetheless, as highlighted in Table 8 there are several studies that do lead the way in terms of innovative approaches to CS (e.g., Aghaee & Keller, 2016; Broisin et al., 2017; Carrascal et al., 2021; Goumopoulos et al., 2017; Knobelsdorf et al., 2017; Pawelczak, 2017; Santos et al., 2018; Troussas et al., 2020).

Table 8 European perspectives on EDU 4.0

Authors	Cluster	E4	H1	H2	H3	H4	H5	H6	H7	H8	H9	EU country
(Apiola et al., 2019)	Light	Y	Y		Y		Y				Y	Finland
(Degener et al., 2019)	Light	Y	Y								Y	Germany
(Dondio & Shaheen, 2019)	Light									Y	Y	Ireland
(Parejo et al., 2020)	Light	Y	Y						Y			Spain
(Schäfer, 2019)	Light	Y	Y								Y	Germany
(Urquiza-Fuentes, 2020)	Light		Y	Y				Y			Y	Spain
(Aghaee & Keller, 2016)	Project	Y	Y			Y	Y		Y		Y	Sweden
(Carrascal et al., 2021)	Project	Y				Y	Y	Y		Y		Spain
(Casañ et al., 2020)	Project	Y				Y	Y					Spain
(Cobos & Roger, 2019)	Project	Y				Y				Y		Spain
(Fagerholm et al., 2018)	Project					Y	Y				Y	Finland
(Ariadna Llorens, Berbegal-Mirabent, & Llinàs-Audet, 2017)	Project	Y				Y	Y					Spain
(Santos et al., 2018)	Project	Y				Y	Y	Y	Y	Y		Austria
(Broisin et al., 2017)	Full	Y	Y	Y	Y		Y				Y	France
(Goumopoulos et al., 2017)	Full	Y	Y	Y	Y	Y	Y	Y			Y	Greece
(Knobelsdorf et al., 2017)	Full	Y	Y	Y			Y	Y			Y	Germany
(Pawelczak, 2017)	Full	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Germany
(Troussas et al., 2020)	Full	Y	Y	Y	Y		Y		Y	Y	Y	Greece

Co-funded by the Erasmus+ Programme of the European Union

8. Discussion and moving forwards

This catalogue is the first intellectual output of the TEACH4EDU4 project. The catalogue is based on a systematic review of recent research literature dealing with innovations in the teaching and learning of Computer Science (CS). The focus is on practices that align with Education 4.0, an area that connects new educational practices with the needs of industry. The catalogue also considers the skills and competences that will be required by educators as they engage in innovative practice.

Aligning the needs of industry with CS programmes requires an overview of the subject area from outside the academy. The TEACH4EDU4 project therefore carried out focus groups that involved employers and employment experts from across Europe. These focus groups identified the skills and capabilities that employers are looking for when they recruit CS graduates. According to the industry partners the main gaps that graduates have are a lack of soft skills such as communication and teamwork. Similarly, CS graduates often struggle to apply their detailed theoretical knowledge to different practical contexts.

In order to identify the skills and competencies required by educators in order to implement innovations in CS education, a Systematic Literature Review (SLR) was conducted amongst 66 articles in 2016-2020. Perhaps surprisingly none of the articles explicitly mentions “Education 4.0”. In part this could be a result of the relatively recent conceptualisation of Education 4.0, and in part this could be due to a lack of adoption of the term Education 4.0 in the specific discipline of CS.

Based upon our broad definition of Education 4.0 in total 54 articles (80%) were considered to fit under this definition. Furthermore, all 66 articles included at least one Hussin (2018) Education 4.0 characteristic. The most common Education 4.0 characteristic was 5) Hands-on Learning (73%), followed by 9) Develops Independence (67%), and 4) Project-based Learning (61%). Around half of the articles included the characteristic that 1) Anytime / Anywhere; 7) Assessed in New Ways (35%) and 8) Student Opinion Counts (32%).

A cluster analysis indicated a three-cluster solution across the 66 papers, which we label as 1) EDU 4.0 light (n = 18), 2) project-based/hands-on learning (n = 22), and 3) full EDU 4.0 (n = 26). EDU 4.0 light studies mostly had relatively low total Hussin (2018) scores, and often did not include project-based activities. EDU 4.0 light studies mostly focussed on developing independence, anytime / anywhere, personalised, and choice in how to learn. As illustrated by the descriptions of some of these studies, substantial technological and pedagogical innovations were introduced in CS courses, although mostly focused on just one or two Education 4.0 characteristics. This could be linked to teachers willing to take some innovations forward based upon a particular problem perceived in a course, but “updating” parts of the pedagogy rather than fully redesigning a CS course (Aničić et al., 2017; Mangaroska & Giannakos, 2019; Rienties et al., 2012).

The second cluster that we labelled as project-based/hands-on learning had a strong focus on project-based learning and hands-on learning. These studies mainly used collaborative and project-based learning approaches with some interesting innovations, such as where students were considered as prospective entrepreneurs (Fagerholm et al., 2018). In these studies there was a strong focus on hands-on and project-based learning, allowing CS graduates to develop strong programming and soft skills, often working in teams. However, due to the nature of project-based learning there was relatively low flexibility in terms of anytime/anywhere, personalisation, and choice of study.

Co-funded by the Erasmus+ Programme of the European Union

The third and final cluster, Full EDU 4.0, was strongly focussed on hands-on learning, developing independence, personalisation, anytime/anywhere study, and choice in how to learn. The lowest Education 4.0 characteristic was student opinion, although this was substantially higher than the other two clusters. Several innovative and integrated perspectives were used including flipped classrooms (Pawelczak, 2017), game-based learning (Troussas et al., 2020) and online lab work (Broisin et al., 2017), indicated how CS teachers might help students to develop strong project, programming, and team skills.

Based on the research articles reviewed in this catalogue, we can conclude that Education 4.0 is a new concept in teaching computer science courses and has not yet been utilised by teachers. This study indicated that although this field is at its early beginnings, some basic trends can be noted and conceptualised. In a way it was surprising to identify three clear clusters in terms of design of CS courses. While in some learning design research there is evidence of common design practices (Mangaroska & Giannakos, 2019; Rienties et al., 2012; Rienties & Toetenel, 2016) when comparing different disciplines, these preliminary findings seem to suggest three broad flavours of design in European CS.

Future research should be carried out to identify and propose corresponding learning designs that would include Education 4.0 characteristics and thus transform the university computer science courses. In addition, it is essential to conduct a wider review within Europe to determine whether these three clusters are unique to Europe, or whether similar/different clusters in CS can be defined across the CS field. Finally, it is essential that more research is conducted to investigate which skills CS teachers might need to develop, implement and evaluate these Education 4.0 courses, and whether or not these courses actually deliver in terms of student expectations and those of industry.

Co-funded by the Erasmus+ Programme of the European Union

9. References

Aghaee, N., & Keller, C. (2016). ICT-supported peer interaction among learners in Bachelor's and Master's thesis courses. *Computers & Education, 94*, 276-297. doi: 10.1016/j.compedu.2015.11.006

Alasbali, N., & Benatallah, B. (2015). *Open source as an innovative approach in computer science education A systematic review of advantages and challenges*. Paper presented at the 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE).

Alegre, F., Moreno, J., Dawson, T., Tanjong, E. E., & Kirshner, D. H. (2020). *Computational Thinking for STEM Teacher Leadership Training at Louisiana State University*. Paper presented at the 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT).

Alomari, H. W., Ramasamy, V., Kiper, J. D., & Potvin, G. (2020). A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education. *Heliyon, 6*(5), e03917. doi: https://doi.org/10.1016/j.heliyon.2020.e03917

Alsaif, S., Li, A. S., Soh, B., & Alraddady, S. (2019). The Efficacy of Facebook in Teaching and Learning: Studied via Content Analysis of Web Log Data. *Procedia Computer Science, 161*, 493-501. doi: https://doi.org/10.1016/j.procs.2019.11.149

Aničić, K. P., Divjak, B., & Arbanas, K. (2017). Preparing ICT Graduates for Real-World Challenges: Results of a Meta-Analysis. *IEEE Transactions on Education, 60*(3), 191-197. doi: 10.1109/TE.2016.2633959

Apiola, M., Lokkila, E., & Laakso, M.-J. (2019). Digital learning approaches in an intermediate-level computer science course. *The International Journal of Information and Learning Technology, 36*(5), 467-484. doi: 10.1108/IJILT-06-2018-0079

Baker, R. S., & Hawn, A. (2021). Algorithmic Bias in Education. *https://doi.org/10.35542/osf.io/pbmvz*.

Behnke, K. A., Kos, B. A., & Bennett, J. K. (2016). *Computer Science Principles: Impacting Student Motivation & Learning Within and Beyond the Classroom*. Paper presented at the Proceedings of the 2016 ACM Conference on International Computing Education Research, Melbourne, VIC, Australia. https://doi.org/10.1145/2960310.2960336

Berikan, B., & Özdemir, S. (2020). Investigating "Problem-Solving With Datasets" as an Implementation of Computational Thinking: A Literature Review. *Journal of Educational Computing Research, 58*(2), 502-534. doi: 10.1177/0735633119845694

Berners-Lee, T. I. M., Hendler, J., & Lassila, O. R. A. (2001). The semantic web. *Scientific American, 284*(5), 34-43.

Bielefeldt, A. R., Polmear, M., Swan, C., Knight, D., & Canney, N. (2017). *An overview of the microethics and macroethics education of computing students in the United States*. Paper presented at the 2017 IEEE Frontiers in Education Conference (FIE).

Blau, I., & Shamir-Inbal, T. (2017). Re-designed flipped learning model in an academic course: The role of co-creation and co-regulation. *Computers & Education, 115*, 69-81. doi: 10.1016/j.compedu.2017.07.014

Bond, M., Zawacki-Richter, O., & Nichols, M. (2019). Revisiting five decades of educational technology research: A content and authorship analysis of the British Journal of Educational Technology. *British Journal of Educational Technology, 50*(1), 12-63. doi: doi:10.1111/bjet.12730

Bonfield, C. A., Salter, M., Longmuir, A., Benson, M., & Adachi, C. (2020). Transformation or evolution?: Education 4.0, teaching and learning in the digital age. *Higher Education Pedagogies, 5*(1), 223-246. doi: 10.1080/23752696.2020.1816847

Borge, M., Ong, Y. S., & Goggins, S. (2020). A sociocultural approach to using social networking sites as learning tools. *Educational Technology Research and Development, 68*(3), 1089-1120. doi: 10.1007/s11423-019-09721-z

Borowczak, M., & Burrows, A. C. (2017). *Interactive Web Notebooks Using the Cloud to Enable CS in K-16+ Classrooms and PDs*. Paper presented at the 2017 ASEE Annual Conference & Exposition.

Co-funded by the
Erasmus+ Programme
of the European Union

Broisin, J., Venant, R., & Vidal, P. (2017). Lab4CE: a remote laboratory for computer education. *International Journal of Artificial Intelligence in Education, 27*(1), 154-180.
Buffardi, K., & Valdivia, P. (2018). *Bug Hide-and-Seek: An Educational Game for Investigating Verification Accuracy in Software Tests*. Paper presented at the 2018 IEEE Frontiers in Education Conference (FIE).
Burrows, A. C., & Borowczak, M. (2017). *Hardening freshman engineering student soft skills*. Paper presented at the Session W1A First Year Engineering Experience (FYEE) Conference.
Burrows, A. C., & Borowczak, M. (2019). Computer science and engineering: utilizing action research and lesson study. *Educational Action Research, 27*(4), 631-646. doi: 10.1080/09650792.2019.1566082
Bushmeleva, N. A., & Baklashova, T. A. (2017). Methodological teaching system of mathematical foundations of formal languages as a means of fundamentalization of education. *EURASIA Journal of Mathematics, Science and Technology Education, 13*(8), 5141-5155.
Caceffo, R., Gama, G., & Azevedo, R. (2018). *Exploring Active Learning Approaches to Computer Science Classes*. Paper presented at the Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA. https://doi.org/10.11almo45/3159450.3159585
Carrascal, G., del Barrio, A. A., & Botella, G. (2021). First experiences of teaching quantum computing. *The Journal of Supercomputing, 77*(3), 2770-2799. doi: 10.1007/s11227-020-03376-x
Casañ, M. J., Alier, M., & Llorens, A. (2020). Teaching Ethics and Sustainability to Informatics Engineering Students, An Almost 30 Years' Experience. *Sustainability, 12*(14), 5499.
Chamberlin, J., Hussey, J., Klimkowski, B., Moody, W., & Morrell, C. (2017). *The Impact of Virtualized Technology on Undergraduate Computer Networking Education*. Paper presented at the Proceedings of the 18th Annual Conference on Information Technology Education, Rochester, New York, USA. https://doi.org/10.1145/3125659.3125693
Charlton, P., & Avramides, K. (2016). Knowledge Construction in Computer Science and Engineering when Learning Through Making. *IEEE Transactions on Learning Technologies, 9*(4), 379-390. doi: 10.1109/TLT.2016.2627567
Cobos, M., & Roger, S. (2019). SART3D: A MATLAB toolbox for spatial audio and signal processing education. *Computer Applications in Engineering Education, 27*(4), 971-985. doi: 10.1002/cae.7
Corritore, C. L., & Love, B. (2020). Redesigning an Introductory Programming Course to Facilitate Effective Student Learning: A Case Study. *Journal of Information Technology Education: Innovations in Practice, 19*, 91-135.
Costa, G. J. M. d., Silva, N., & Fonseca, T. (2012). Moral reasoning in knowledge authoring: An e-learning 4.0 analysis! In S. Abramovich (Ed.), *Computers in Education* (pp. 135-154): Nova Science Publishers.
Degener, P., Haak, V., Gold-Veerkamp, C., & Abke, J. (2019). *Towards the Vision of an LMS Integrated, Browser-Based Simulation to Program LEGO MindStorms EV3s in ANSI-C*. Paper presented at the 2019 IEEE Global Engineering Education Conference (EDUCON).
Dickson, P. E., Dragon, T., & Lee, A. (2017). *Using Undergraduate Teaching Assistants in Small Classes*. Paper presented at the Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, Seattle, Washington, USA. https://doi.org/10.1145/3017680.3017725
Dondio, P., & Shaheen, S. (2019). *Is StackOverflow an Effective Complement to Gaining Practical Knowledge Compared to Traditional Computer Science Learning?* Paper presented at the Proceedings of the 2019 11th International Conference on Education Technology and Computers, Amsterdam, Netherlands.
Erickson, M., Hanna, P., & Walker, C. (2020). The UK higher education senior management survey: a statactivist response to managerialist governance. *Studies in Higher Education*, 1-18. doi: 10.1080/03075079.2020.1712693
European Commission. (2015a). Erasmus: Facts, figures and trends *The European Union support for student and staff exchanges and university cooperation in 2013–14* (pp. 1-40). Luxembourg.

European Commission. (2015b). Joint Report of the Council and the Commission on the implementation of the strategic framework for European cooperation in education and training *Official Journal of the European Union* (pp. 1-11). Luxembourg.

European Commission/EACEA/Eurydice. (2018). Teaching Careers in Europe: Access, Progression and Support. Luxembourg: Eurydice Report,.

Fagerholm, F., Hellas, A., Luukkainen, M., Kyllönen, K., Yaman, S., & Mäenpää, H. (2018). Designing and implementing an environment for software start-up education: Patterns and anti-patterns. *Journal of Systems and Software, 146*, 1-13. doi: 10.1016/j.jss.2018.08.060

Fisher, W., Rader, C., & Camp, T. (2016). *Online programming tutors or paper study guides?* Paper presented at the 2016 IEEE Frontiers in Education Conference (FIE).

Fisk, P. (2017). Education 4.0 … the future of learning will be dramatically different, in school and throughout life., 20 January 2021, from https://www.thegeniusworks.com/2017/01/future-education-young-everyone-taught-together/

Foster, E., Siddle, R., Crowson, P., & Bonne, P. (2020). It's All About the Intervention: Reflections on Building Staff Capacity for Using Learning Analytics to Support Student Success. In D. Ifenthaler & D. Gibson (Eds.), *Adoption of Data Analytics in Higher Education Learning and Teaching* (pp. 241-256). Cham: Springer International Publishing.

Frevert, T., Rorrer, A., Davis, D. J., Latulipe, C., Maher, M. L., Cukic, B., . . . Rogelberg, S. (2018). *Sustainable Educational Innovation Through Engaged Pedagogy and Organizational Change*. Paper presented at the 2018 IEEE Frontiers in Education Conference (FIE).

Garousi, V., Giray, G., Tüzün, E., Catal, C., & Felderer, M. (2019). Aligning software engineering education with industrial needs: A meta-analysis. *Journal of Systems and Software, 156*, 65-83. doi: 10.1016/j.jss.2019.06.044

Gestwicki, P., & McNely, B. (2016). Interdisciplinary Projects in the Academic Studio. *ACM Trans. Comput. Educ., 16*(2), Article 8. doi: 10.1145/2732157

Giacaman, N., & De Ruvo, G. (2018). Bridging Theory and Practice in Programming Lectures With Active Classroom Programmer. *IEEE Transactions on Education, 61*(3), 177-186. doi: 10.1109/TE.2018.2819969

Gonçalves, R. Q., von Wangenheim, C. A. G., Hauck, J. C. R., & Zanella, A. (2018). An Instructional Feedback Technique for Teaching Project Management Tools Aligned With PMBOK. *IEEE Transactions on Education, 61*(2), 143-150. doi: 10.1109/TE.2017.2774766

Goumopoulos, C., Nicopolitidis, P., Gavalas, D., & Kameas, A. (2017). A distance learning curriculum on pervasive computing. *International Journal of Continuing Engineering Education and Life Long Learning, 27*(1-2), 122-146. doi: 10.1504/ijceell.2017.080996

Guan, C., Mou, J., & Jiang, Z. (2020). Artificial intelligence innovation in education: A twenty-year data-driven historical analysis. *International Journal of Innovation Studies, 4*(4), 134-147. doi: https://doi.org/10.1016/j.ijis.2020.09.001

Halili, S. H. (2019). Technological advancements in education 4.0. *The Online Journal of Distance Education and E-Learning, 7*(1), 63-69.

Harkins, A. M. (2008). Leapfrog principles and practices: Core components of education 3.0 and 4.0. *Futures Research Quarterly, 24*(1), 19-31.

Harris, T. (2016). How technology Hijacks people's minds—From a magician and Google's design ethicist. *Medium Magazine*.

Herodotou, C., Naydenova, G., Boroowa, A., Gilmour, A., & Rienties, B. (2020). How Can Predictive Learning Analytics and Motivational Interventions Increase Student Retention and Enhance Administrative Support in Distance Education? *Journal of Learning Analytics, 7*(2), 72-83. doi: 10.18608/jla.2020.72.4

Herodotou, C., Rienties, B., Hlosta, M., Boroowa, A., Mangafa, C., & Zdrahal, Z. (2020). The scalable implementation of predictive learning analytics at a distance learning university: Insights from a longitudinal case study. *The Internet and Higher Education, 45*, 100725. doi: 10.1016/j.iheduc.2020.100725

Hew, K. F., & Lo, C. K. (2018). Flipped classroom improves student learning in health professions education: a meta-analysis. *BMC Medical Education, 18*(1), 38. doi: 10.1186/s12909-018-1144-z

Higher Education Commission. (2016). From bricks to clicks. In X. Shacklock (Ed.), (pp. 1-76). London: Higher Education Commission.

Hosseini, H., Hartt, M., & Mostafapour, M. (2019). Learning IS Child's Play: Game-Based Learning in Computer Science Education. *ACM Trans. Comput. Educ., 19*(3), Article 22. doi: 10.1145/3282844

House, R. J., Hanges, P. J., Javidan, M., Dorfman, P. W., & Gupta, V. (2004). *Culture, leadership and organizations: The GLOBE study of 62 Societies*. Thousand Oaks, CA: Sage Publications Inc.

Hundhausen, C. D., Olivares, D. M., & Carter, A. S. (2017). IDE-Based Learning Analytics for Computing Education: A Process Model, Critical Review, and Research Agenda. *ACM Trans. Comput. Educ., 17*(3), Article 11. doi: 10.1145/3105759

Hussin, A. A. (2018). Education 4.0 made simple: Ideas for teaching. *International Journal of Education and Literacy Studies, 6*(3), 92-98.

Jisc. (2019). Education 4.0 – transforming the future of education (through advanced technology). Retrieved 20 January 2021, from https://www.youtube.com/watch?v=aVWHp8FsV1w

Juárez, E., Aldeco-Pérez, R., & Velázquez, J. M. (2020). Academic approach to transform organisations: one engineer at a time. *IET Software, 14*(2), 106-114.

Knobelsdorf, M., Frede, C., Böhne, S., & Kreitz, C. (2017). *Theorem Provers as a Learning Tool in Theory of Computation*. Paper presented at the Proceedings of the 2017 ACM Conference on International Computing Education Research, Tacoma, Washington, USA.

Korir, M., Slade, S., Holmes, W., & Rienties, B. (2021). Eliciting students' preferences for the use of their data for learning analytics: A crowdsourcing approach. In B. Rienties, R. Hampel, E. Scanlon, & D. Whitelock (Eds.), *Open World Learning: Research, Innovation and the Challenges of High-Quality Education* (pp. XX-XX). London: Routledge.

Lewis, M. C., & Lacher, L. L. (2020). *Teaching Modern Multithreading in CS2 with Actors*. Paper presented at the 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).

Liang, Z., & Chapa-Martell, M. A. (2018). *A Top-Down Approach to Teaching Web Development in the Cloud*. Paper presented at the 2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE).

Llorens, A., Berbegal-Mirabent, J., & Llinas-Audet, X. (2016). Aligning professional skills and active learning methods: an application for information and communications technology engineering. *European Journal of Engineering Education,* 1-14.

Llorens, A., Berbegal-Mirabent, J., & Llinàs-Audet, X. (2017). Aligning professional skills and active learning methods: an application for information and communications technology engineering. *European Journal of Engineering Education, 42*(4), 382-395. doi: 10.1080/03043797.2016.1189880

Mäkiö, E., Yablochnikov, E., Colombo, A. W., Mäkiö, J., & Harrison, C. (2020). *Applying Task-centric Holistic Teaching Approach in Education of Industrial Cyber Physical Systems*. Paper presented at the 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS).

Mangaroska, K., & Giannakos, M. N. (2019). Learning analytics for learning design: A systematic literature review of analytics-driven design to enhance learning. *IEEE Transactions on Learning Technologies, 12*(4), 516-534. doi: 10.1109/TLT.2018.2868673

McMahon, G. T. (2017). The Leadership Case for Investing in Continuing Professional Development. *Academic Medicine, 92*(8), 1075-1077. doi: 10.1097/acm.0000000000001619

Medeiros, R. P., Ramalho, G. L., & Falcão, T. P. (2019). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. *IEEE Transactions on Education, 62*(2), 77-90. doi: 10.1109/TE.2018.2864133

Co-funded by the
Erasmus+ Programme
of the European Union

Munkvold, R. I. (2017). *Game lab: a practical learning approach for game development*. Paper presented at the European Conference on Games Based Learning.

Paolacci, G., & Chandler, J. (2014). Inside the Turk:Understanding Mechanical Turk as a Participant Pool. *Current Directions in Psychological Science, 23*(3), 184-188. doi: 10.1177/0963721414531598

Parejo, J. A., Troya, J., Segura, S., del-Río-Ortega, A., Gámez-Díaz, A., & Márquez-Chamorro, A. E. (2020). Flipping Laboratory Sessions: An Experience in Computer Science. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, 15*(3), 183-191. doi: 10.1109/RITA.2020.3008132

Park, H., & Kim, Y. (2020). CLIK: Cloud-based Linux kernel practice environment and judgment system. *Computer Applications in Engineering Education, 28*(5), 1137-1153. doi: https://doi.org/10.1002/cae.22289

Paschoal, L. N., Oliveira, B. R. N., Nakagawa, E. Y., & Souza, S. R. S. (2019). *Can we use the Flipped Classroom Model to teach Black-box Testing to Computer Students?* Paper presented at the Proceedings of the XVIII Brazilian Symposium on Software Quality, Fortaleza, Brazil. https://doi.org/10.1145/3364641.3364659

Pawelczak, D. (2017). *Comparison of traditional lecture and flipped classroom for teaching programming.* Paper presented at the Proceedings of the 3rd International Conference on Higher Education Advances.

Peng, C. (2015). *Introductory Game Development Course: A Mix of Programming and Art*. Paper presented at the 2015 International Conference on Computational Science and Computational Intelligence (CSCI).

Peteranetz, M. S., Flanigan, A. E., Shell, D. F., & Soh, L. (2017). Computational Creativity Exercises: An Avenue for Promoting Learning in Computer Science. *IEEE Transactions on Education, 60*(4), 305-313. doi: 10.1109/TE.2017.2705152

Pilkington, C. (2017). *Questioning the Value of Vodcasts in a Distance Learning Theoretical Computer Science Course*. Cham.

Pivkina, I. (2016). *Peer learning assistants in undergraduate computer science courses*. Paper presented at the 2016 IEEE Frontiers in Education Conference (FIE).

Puncreobutr, V. (2016). Education 4.0: New challenge of learning. *St. Theresa Journal of Humanities and Social Sciences, 2*(2).

Quality Assurance Agency. (2019). Subject Benchmark Statement on Computing. Gloucester, UK: Quality Assurance Agency.

Rienties, B. (2014). Understanding academics' resistance towards (online) student evaluation. *Assessment & Evaluation in Higher Education, 39*(8), 987-1001. doi: 10.1080/02602938.2014.880777

Rienties, B., Hampel, R., Scanlon, E., & Whitelock, D. (2021). *Open World Learning: Research, Innovation and the Challenges of High-Quality Education*. London: Routledge.

Rienties, B., Kaper, W., Struyven, K., Tempelaar, D. T., Van Gastel, L., Vrancken, S., . . . Virgailaite-Meckauskaite, E. (2012). A review of the role of Information Communication Technology and course design in transitional education practices. *Interactive Learning Environments, 20*(6), 563-581. doi: 10.1080/10494820.2010.542757

Rienties, B., & Toetenel, L. (2016). The impact of learning design on student behaviour, satisfaction and performance: a cross-institutional comparison across 151 modules. *Computers in Human Behavior, 60*, 333-341. doi: 10.1016/j.chb.2016.02.074

Ruiz, J., Serral Asensio, E., & Snoeck, M. (2020). Learning UI Functional Design Principles Through Simulation With Feedback. *IEEE Transactions on Learning Technologies, 13*(4), 833-846. doi: 10.1109/TLT.2020.3028596

Salem, F. A., Damaj, I. W., Hamandi, L., & Zantout, R. N. (2020). Effective Assessment of Computer Science Capstone Projects and Student Outcomes. *iJEP, 10*(2), 72-93.

Salmon, G. (2019). May the Fourth Be with you: Creating Education 4.0. *Journal of Learning for Development, 6*(2).

Co-funded by the
Erasmus+ Programme
of the European Union

Santos, B. S., Dischler, J.-M., Adzhiev, V., Anderson, E. F., Ferko, A., Fryazinov, O., . . . Zara, J. (2018). Distinctive Approaches to Computer Graphics Education. *Computer Graphics Forum, 37*(1), 403-412. doi: https://doi.org/10.1111/cgf.13305

Scanlon, E., Sharples, M., Fenton-O'Creevy, M., Fleck, J., Cooban, C., Ferguson, R., . . . Waterhouse, P. (2013). Beyond prototypes: Enabling innovation in technology-enhanced learning. Milton Keynes: Open University.

Scatalon, L. P., Garcia, R. E., & Barbosa, E. F. (2020). *Teaching Practices of Software Testing in Programming Education*. Paper presented at the 2020 IEEE Frontiers in Education Conference (FIE).

Schäfer, U. (2019). *Teaching Modern C++ with Flipped Classroom and Enjoyable IoT Hardware*. Paper presented at the 2019 IEEE Global Engineering Education Conference (EDUCON).

Seyam, M., & McCrickard, D. S. (2016). *Teaching Mobile Development with Pair Programming*. Paper presented at the Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA. https://doi.org/10.1145/2839509.2844644

Seyam, M., McCrickard, D. S., Niu, S., Esakia, A., & Kim, W. (2016). *Teaching mobile application development through lectures, interactive tutorials, and Pair Programming*. Paper presented at the 2016 IEEE Frontiers in Education Conference (FIE).

Shi, N., Min, Z., & Zhang, P. (2017). Effects of visualizing roles of variables with animation and IDE in novice program construction. *Telematics and Informatics, 34*(5), 743-754. doi: https://doi.org/10.1016/j.tele.2017.02.005

Silva, P. A., Polo, B. J., & Crosby, M. E. (2017). Adapting the Studio Based Learning Methodology to Computer Science Education. In S. B. Fee, A. M. Holland-Minkley, & T. E. Lombardi (Eds.), *New Directions for Computing Education: Embedding Computing Across Disciplines* (pp. 119-142). Cham: Springer International Publishing.

Silva, W., Steinmacher, I., & Conte, T. (2019). Students' and Instructors' Perceptions of Five Different Active Learning Strategies Used to Teach Software Modeling. *IEEE Access, 7*, 184063-184077. doi: 10.1109/ACCESS.2019.2929507

Suhaimi, S. (2019). Education 4.0: The Impact of Computer Architecture and Organization Course on Students' Computer Anxiety and Computer Self-Efficacy. *International Journal of Advanced Trends in Computer Science and Engineering, 8*, 3022-3025. doi: 10.30534/ijatcse/2019/57862019

Tanaka, R., Ferreira da Silva, R., & Casanova, H. (2019). *Teaching Parallel and Distributed Computing Concepts in Simulation with WRENCH*. Paper presented at the 2019 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC).

Tlili, A., Essalmi, F., Jemni, M., & Kinshuk. (2017). *Towards Applying Keller's ARCS Model and Learning by doing strategy in Classroom Courses*, Singapore.

Troussas, C., Krouska, A., & Sgouropoulou, C. (2020). Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education. *Computers & Education, 144*, 103698. doi: 10.1016/j.compedu.2019.103698

Tyler, B., & Abdrakhmanova, M. (2016). *Flipping the CS1 and CS2 classrooms in Central Asia*. Paper presented at the 2016 IEEE Frontiers in Education Conference (FIE).

Urquiza-Fuentes, J. (2020). Increasing Students' Responsibility and Learning Outcomes Using Partial Flipped Classroom in a Language Processors Course. *IEEE Access, 8*, 211211-211223. doi: 10.1109/ACCESS.2020.3039628

Wakelam, E., Jefferies, A., Davey, N., & Sun, Y. (2019). The potential for student performance prediction in small cohorts with minimal available attributes. *British Journal of Educational Technology, 0*(0). doi: 10.1111/bjet.12836

Wallner, T., & Wagner, G. (2016). *Academic Education 4.0.* Paper presented at the International Conference on Education and New Developments.

Wasson, B., & Kirschner, P. A. (2020). Learning Design: European Approaches. *TechTrends*, 1-13.

Co-funded by the Erasmus+ Programme of the European Union

Webb, M., & Doctorow, C. (2020). 8 THE GATHERING STORM *Coding Democracy: How Hackers Are Disrupting Power, Surveillance, and Authoritarianism* (pp. 139-173): MIT Press.

Wiley, D., & Hilton Iii, J. L. (2018). Defining OER-Enabled Pedagogy. *International Review of Research in Open and Distributed Learning, 19*(4). doi: https://doi.org/10.19173/irrodl.v19i4.3601

Winiecki, D., & Salzman, N. (2019). *Teaching Professional Morality & Ethics to Undergraduate Computer Science Students through Cognitive Apprenticeships & Case Studies: Experiences in CS-HU 130 'Foundational Values'*. Paper presented at the 2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT).

Wood, Z. J., Clements, J., Peterson, Z., Janzen, D., Smith, H., Haungs, M., . . . DeBruhl, B. (2018). *Mixed Approaches to CS0: Exploring Topic and Pedagogy Variance after Six Years of CS0*. Paper presented at the Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA. https://doi.org/10.1145/3159450.3159592

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education, 16*(1), 39. doi: 10.1186/s41239-019-0171-0

Zuboff, S. (2015). Big other: surveillance capitalism and the prospects of an information civilization. *Journal of Information Technology, 30*(1), 75-89. doi: 10.1057/jit.2015.5

Co-funded by the
Erasmus+ Programme
of the European Union

10. Appendix

10.1 Appendix A1 Detailed descriptions of included EDU 4.0 light studies.

06 Digital learning approaches in an intermediate-level Computer Science course

(Apiola et al., 2019)

Education 4.0	Y
Anytime / Anywhere	Y
Choice in How to Learn	Y
Hands-on Learning	Y
Develops Independence	Y

An intermediate-level CS course in Finland was transformed to use the collaborative educational tool ViLLE. This tool includes automatic assessment, immediate feedback, support for a range of exercise types and teaching modes such as pair programming. This transformation included development of digital learning materials and tasks, as well as redesigning the tutorials and learning sessions of the course. The course was originally offered in blended learning mode, with on-site instruction accompanied by extensive electronic practices and an electronic exam. This exam provides students with an environment where they can test, debug and fix their answers before submitting, as they would in an authentic coding environment.

15 Computer Science and Engineering: utilizing action research and lesson study

(Burrows & Borowczak, 2019)

Personalised	Y
Choice in How to Learn	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Student Opinion Counts	Y

This paper explores one group's use of action research and lesson study in three US university-level CS courses. It offers an overview of three areas including action research, lesson study, and engineering soft skills. The research group identified a problem: undergraduate Engineering student soft skill understanding. The group collaborated to enhance participant engagement with this problem by utilizing one lesson focused on soft skills over three research lesson iterations in three distinct university semesters. Undergraduate CS and Engineering students were interested and engaged in soft-skill learning and, with explicit instruction and guidance, were able to understand and apply soft skills in their study and potentially in their future positions.

24 Towards the vision of an LMS integrated, browser-based simulation to program LEGO MindStorms EV3s in ANSI-C

(Degener et al., 2019)

LEGO MindStorms EV3 robots were integrated within lessons teaching the programming language ANSI-C. The intention was to make practical programming lessons more tangible and closer to the future field of work for Engineering students. Although the EV3 contributed to learning success and fun during the lessons, students were not able to program the EV3 outside laboratory opening

Co-funded by the Erasmus+ Programme of the European Union

hours and practical lessons. To solve this problem, a simulation was provided to make the programming task time-and location-independent. Open-source tools were used throughout.

Education 4.0	Y
Anytime / Anywhere	Y
Develops Independence	Y

25 Using undergraduate teaching assistants in small classes

(Dickson et al., 2017**)**

Personalised	Y
Choice in How to Learn	Y
Hands-on Learning	Y

Undergraduate teaching assistants (UTAs) were used in a small college, small class environment. The paper considers the benefits and challenges of using and training UTAs. They helped in traditional lecture-style classes by providing educators with better feedback on student understanding, and by providing extra support for students. UTAs were also used to create a hybrid class environment with more hands-on work and interactive exercises made possible by UTAs offering additional one-on-one instruction and tutoring. When a problem cannot be solved with the help of a UTA, an educator can step in. As a result, introductory-level students see UTAs learning alongside them, which makes the level of knowledge held by UTAs seem more attainable. At the same time the UTAs gain deeper insight by finding out how to solve a difficult problem.

26 Is StackOverflow an effective complement to gaining practical knowledge compared to traditional Computer Science learning?

(Dondio & Shaheen, 2019**)**

Student Opinion Counts	Y
Develops Independence	Y

The paper proposes a method to help CS lecturers to include in their teaching content from StackOverflow, the popular Q&A website about computer programming accessed by both students and professional developers. The site offers one of the largest repositories of problem-solving and practical content related to programming. Incorporating these resources within courses is a way of providing students with authentic practice that will help prepare them for their future careers. The study showed StackOverflow content was at least as effective as traditional teaching materials and had a greater impact on students' short-term learning.

28 Online programming tutors or paper study guides?

(Fisher et al., 2016**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Student Opinion Counts	Y

An undergraduate data structures course is challenging to teach due to the vast number of complex topics that must be covered. Instructional tools, such as an online programming tutor, can reinforce topics students typically find difficult. Prior research has shown that using programming

Co-funded by the Erasmus+ Programme of the European Union

tutors can have a positive impact on student learning in an introductory CS course. The authors trialled use of an online tutor in a second-year programming course but found that simply providing an optional tool had limited value.

29 Sustainable educational innovation through engaged pedagogy and organizational change

(Frevert et al., 2018**)**

Education 4.0	Y
Student Opinion Counts	Y
Develops Independence	Y

This is a midway report on a five-year change initiative designed to revolutionise CS education at the college level through both pedagogical and organisational change, is built upon a foundation of educational innovation through engaged teaching practices. Pedagogical change is strategically embedded through a three-stage model of faculty adoption, redesigned student course evaluations, and realigned values in the reappointment, promotion, and tenure process. Classrooms that were redesigned to foster interactive, engaged teaching and learning by adding movable, reconfigurable chairs and tables as well as multiple viewing screens, dry erase surfaces, and ample electric receptacles for students' laptops and tablets. The programme refocused student learning on connection to their peers, their purpose and their profession, enabling them to leave with the identity of a computing professional. Pedagogic innovation included engaged pedagogy, peer learning, meaningful learning activities and the establishment of ongoing partnerships with industry.

31 Bridging theory and practice in programming lectures with Active Classroom Programmer

(Giacaman & De Ruvo, 2018**)**

Choice in How to Learn	Y

This paper introduces Active Classroom Programmer (ACP), a software tool that encourages a high degree of engagement. ACP allows students to program alongside instructors as new concepts are introduced. The instructor delivers a lecture, the students then work on an exercise before the lecture resumes. However, students are not in the lab and are not expected to have a computer in front of them. The focus is on the instructor guiding students through a programming strategy. As the students are not studying independently, they can draw on the instructor's expert knowledge. Once the lecture is complete all the tags and snapshots made (either during the lesson or prepared by the instructor outside of lessons) remain available for students, providing them with a rich collection of practical exercises they can review.

34 Learning is child's play: game-based learning in Computer Science education

(Hosseini et al., 2019**)**

Education 4.0	Y
Assessed in New Ways	Y
Develops Independence	Y

The authors used game design in CS courses to improve students' perception of learning, engagement and teamwork. They introduced modular game-based learning in two topics that were

Co-funded by the Erasmus+ Programme of the European Union

similar in terms of difficulty and compared this approach to a more traditional one without game-based learning. They found the nature of the course and the disciplinary culture affect how students perceive, and subsequently respond to, different pedagogical methods.

41 Flipping laboratory sessions: an experience in Computer Science

(Parejo et al., 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Assessed in New Ways	Y

This paper reports on the experience of flipping a course on software architecture and integration, that formed part of a Software Engineering degree. In addition, the gamified platform Kahoot was used for interactive tests at the beginning of the laboratory sessions. Based on the answers to these quick quizzes, the lab instructor decided which concepts to clarify. Students had, on average, 24 more minutes per session to solve in-class exercises when using the flipped-classroom approach. More than 70% of students considered the quantity, duration and didactic content of the videos (very) appropriate; and 90% of students preferred this approach for laboratory sessions.

42 CLIK: cloud-based Linux kernel practice environment and judgment system

(Park & Kim, 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Assessed in New Ways	Y
Develops Independence	Y

Assignments on kernel programming are essential parts of operating system courses taught to CS students to provide them a deep understanding of real-world OSs. However, students are routinely flustered by the daunting task of building a practice environment from scratch and instructors are pressed for time while validating student work that requires several kernel installations and reboots. This paper proposes CLIK as a solution. This is a cloud-based Linux kernel practice environment that supports automatic judgment. It provides students with an individual and easy-to-use kernel practice environment and instructors with a fast and easy evaluation of student work with live feedback. The cloud-based system provides a high level of accessibility, enabling students to work on their assignments anywhere they want.

47 Questioning the value of vodcasts in a distance learning theoretical Computer Science course

(Pilkington, 2017**)**

Education 4.0	Y
Anytime / Anywhere	Y
Choice in How to Learn	Y
Develops Independence	Y

A series of vodcasts was produced and used as supplementary material in a distance learning theoretical CS course. The expected benefits were (1) flexibility, with learners able to control the flow of the vodcast, pausing and reviewing where necessary and (2) accessibility of learning

Co-funded by the Erasmus+ Programme of the European Union

content outside the lecture theatre. However, vodcasts may also assign students a passive role with low levels of engagement and interaction and require discipline and self-regulated learning skills. This study found no measurable increase in student performance associated with the use of vodcasts in this context, but also found low rates of use.

52 Teaching practices of software testing in programming education

(Scatalon et al., 2020)

Personalised	Y

An overview of the practices that have been used to integrate software testing into programming education. Information was collected from 195 papers reporting empirical studies: (i) course; (ii) programming language; (iii) testing concepts presented to students; (iv) testing practices adopted by students in programming assignments and (v) support tools. Very few studies addressed how students learn testing concepts in programming courses. The study showed testing practices in programming assignments involve students to different extents: analysing test results from submission tools, working with instructor-provided tests, using support mechanisms to design tests (e.g. plugins where students insert inputs and expected outputs) and, finally, students writing their own tests.

53 Teaching Modern C++ with flipped classroom and enjoyable IoT hardware

(Schäfer, 2019)

Education 4.0	Y
Anytime / Anywhere	Y
Develops Independence	Y

A novel approach to teaching modern C++ to Electrical Engineering and CS students. The approach uses flipped classroom and the practical exercises are centred on an Internet of Things device. The main motivations of this innovation are to: (1) guide students to a lifelong learning mode, (2) increase the time available to practise software development in presence of a trainer, (3) motivate programming practice by using attractive IoT hardware, (4) focus on modern C++.

56 Effects of visualizing roles of variables with animation and IDE in novice program construction

(Shi et al., 2017)

Education 4.0	Y
Choice in How to Learn	Y

Programming is one of the core competencies for every CS graduate. However, there is a high rate of failure in introductory programming courses. The roles of variables (for example, fixed value, stepper, and gatherer) are an aspect of programming knowledge that can be explicitly taught to students. This study evaluates whether the role-based approach with visualization tools is an effective way of teaching novice programmers to construct programs in the C language. The PlanAni animation system was used to demonstrate the roles of variables, showing how the successive values of variables are related to each other and to other variables. Although there was some improvement in the ways students engaged with programming, marks were similar between the two groups, and students were not satisfied with their program construction capability.

Co-funded by the Erasmus+ Programme of the European Union

58 Students' and instructors' perceptions of five different active learning strategies used to teach software modeling

(W. Silva et al., 2019)

Education 4.0	Y
Choice in How to Learn	Y
Student Opinion Counts	Y
Develops Independence	Y

The purpose of this paper was to understand how active learning strategies influence the teaching and learning process of Unified Modelling Language (UML) diagrams. Instructors were introduced to active learning, and selected teaching strategies from a catalogued web portal. The study identified 15 factors that influence active learning. In particular, students need prior knowledge about the content; instructors need to take care when setting the level of difficulty; instructors should be aware of how long it takes to apply the strategies; and instructors must carefully develop the materials needed to implement the strategies.

62 Flipping the CS1 and CS2 classrooms in Central Asia

(Tyler & Abdrakhmanova, 2016)

Personalised	Y
Choice in How to Learn	Y
Data Interpretation	Y
Student Opinion Counts	Y

The paper reports on the experience of transitioning to a flipped classroom model for first-year undergraduate programming courses in a newly-formed, Western-style university in Central Asia. Before the transition, courses were taught in a more traditional lecture-lab setting. Now, all lesson content is contained in online video lectures, which are embedded in WordPress course pages. In-class time is spent almost exclusively on students working through programming exercises in pairs, getting help from instructors and teaching assistants as needed. The move to a flipped classroom allowed effective coverage of material in more depth in lectures, enabled students to focus more on hands-on activities during class time and enabled educators to deliver course content in a consistent manner from semester to semester. The shift was associated a slight, though not significant, improvement in exam scores and very positive student feedback.

63 Increasing students' responsibility and learning outcomes using partial flipped classroom in a Language Processors course

(Urquiza-Fuentes, 2020)

Anytime / Anywhere	Y
Personalised	Y
Data Interpretation	Y
Develops Independence	Y

This paper presents a comparative study between traditional lectures and a partial application of the flipped classroom approach in a CS degree on a course dealing with Language Processors. Students in the flipped classroom group took more responsibility for their learning process and

Co-funded by the Erasmus+ Programme of the European Union

achieved significantly better learning outcomes either during or at the end of the learning process. Students' satisfaction was also very positive.

Co-funded by the Erasmus+ Programme of the European Union

10.2 Appendix A2 Detailed descriptions of included project-based/hands-on studies.

01 ICT-supported peer interaction among learners in Bachelor's and Master's thesis courses

(Aghaee & Keller, 2016)

Education 4.0	Y
Anytime / Anywhere	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Develops Independence	Y

This study monitors how an ICT-based Support System facilitates peer interaction in thesis production at undergraduate and postgraduate levels in Sweden. The system facilitates peer interaction in three ways: peer reviews, active participation, and final defence. The peer portal facilitates online peer interaction, motivating learners to support their peers by providing and receiving feedback on their thesis manuscripts before reaching the final phase of thesis work. This process enables both authors and reviewers to learn and to improve their theses. The learners perceived the peer interaction useful to enhance the quality of the thesis outcomes. However, several factors affect the quality of peer interaction in different phases of the thesis process (e.g. quality of thesis manuscripts, supervisors' control and grading, clear instructions and guidelines, learners' understanding of the peer interaction and why it takes place, previous training, and learners' motivation to perform peer reviews). The study developed a set of strategic suggestions from both pedagogical and technical aspects to enhance the peer interaction in the thesis process.

02 Open source as an innovative approach in computer science education: a systematic review of advantages and challenges

(Alasbali & Benatallah, 2015)

Education 4.0	N
Project-based Learning	Y
Develops Independence	Y

This SLR focuses on the advantages and challenges of using open source materials in CS education. The review highlights an innovative approach of asking CS students to become active members of existing open source communities as part of their curriculum. The acquisition of a wide range of skills, increasing student motivation, support for contextual learning and student-centred courses as well as the availability of a wealth of data to inform and support decision making by educators are the main advantages identified by the review. The paper focuses on the engagement of students in using open source software, running 'tasks with real-world relevance, providing an environment that is most likely to keep students highly motivated and interested'. Rather than simply using open source software, students are required to become members of an open source community.

03 Computational thinking for STEM teacher leadership training at Louisiana State University

(Alegre et al., 2020)

Co-funded by the Erasmus+ Programme of the European Union

Education 4.0	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y

This short paper reports on a new course in CS Teaching Methods, based on materials from a previous program that trained high school teachers in computational thinking and programming. The course targeted undergraduate CS majors who were considering a career in teaching or who were interested in the pedagogy of CS. The study identified a disconnect between the pedagogical practices promoted for teaching Computing at high schools and those practised at college level. Students expressed interest in the potential of using more student-centred instruction, not only for high school instruction, but also for their own college courses. An area of disconnect also emerged in the programming formats. All the students were comfortable with Java but unfamiliar with popular block-based programming platforms, such as Scratch.

04 A User Interface (UI) and User eXperience (UX) evaluation framework for cyberlearning environments in computer science and software engineering education

(Alomari et al., 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y

This article focuses on usability and utility evaluations for cyberlearning environments. Multiple user studies are presented that can be used to assess the usefulness of a cyberlearning environment used in CS and Software Engineering (SE) courses through testing its usability and measuring its utility using user interface and user experience evaluations. SEP-CyLE (Software Engineering and Programming Cyberlearning Environment), an external web-based learning tool that helps instructors integrate software development concepts into their programming and software engineering courses, was selected for this research. The main goal was to improve the testing knowledge of 51 CS/SE students by providing a set of learning objects (LOs) and tutorials, which were sequentially categorized by difficulty level. This has evolved into a collaborative learning environment that includes social networking features such as the ability to award virtual points for student social interaction about testing. Educators may choose to use these as a part of students' grades or for the motivation that collaborative learning and gamification can provide. Based on these assessments, they propose an evaluation framework to evaluate cyberlearning environments. The evaluation techniques used are cognitive walkthroughs with a think-aloud protocol and a heuristic evaluation survey. They used a network-based analysis to find the statistically significant correlated responses in the heuristic evaluation survey with regard to the students' perceptions of using SEP-CyLE.

08 Investigating 'problem-solving with datasets' as an implementation of computational thinking: a literature review

(Berikan & Özdemir, 2020**)**

Education 4.0	Y
Project-based Learning	Y

Hands-on Learning	Y
Data Interpretation	Y

This study reviews relevant literature on problem-solving with datasets (PSWD), which can be used to introduce and improve computational thinking. 54 publications were analysed via content analysis. It draws attention to the connections between this approach and higher-order thinking skills such as evidence-based reasoning, critical thinking, analytical thinking, and abstract thinking. The approach also emphasises the connection with soft skills such as having the ability to formulate new or creative questions, as well as being able to determine the role and flow of data in a system. The most frequently suggested or used teaching strategies (in terms of instructional methods, instructional tools, and grade level) for PSWD in the literature were explored.

09 An overview of the microethics and macroethics education of Computing students in the United States

(Bielefeldt et al., 2017**)**

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y

This study considers the integration of ethical and societal issues into CS education. 'Microethics' refers to individual responsibilities, and 'macroethics' to the broader responsibilities of the profession to society. Survey responses from faculty who teach computing students (n=188) were compared to other engineering disciplines (n=1161). The most common topics taught by the computing respondents were identified. The paper identifies teaching methods currently used to teach ethics in both Computing and Engineering courses and raises the need to improve courses in terms of the integration of ethics into teaching. The methods used include: examples of professional scenarios, project-based learning, videos, reflection, debates, and role-plays.

11 Interactive Web Notebooks using the Cloud to enable CS in K-16+ Classrooms and PDs

(Borowczak & Burrows, 2017**)**

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Student Opinion Counts	Y
Develops Independence	Y

Cloud-based implementations of interactive code notebooks and multi-agent simulators enable instructors to incorporate CS into their existing science, technology, engineering and mathematics (STEM) courses and professional development. Three case studies are reported. Although these are at different educational levels, the learning considered is that of the adult instructor rather than the young learners. The case studies originated during a 16-day professional development, during which instructors were exposed to the Python programming language and NetLogo. All of the programming, notes, and lesson materials were hosted using interactive notebooks and open source repositories. The interactive notebooks used open-source Jupyter notebooks hosted using JupyterHub – a related open source project. NetLogo activities and code were either hosted on NetLogo's own site, the modelling commons, or using GitHub, a freely available online code repository site.

Co-funded by the Erasmus+ Programme of the European Union

14 Hardening freshman engineering student soft skills

(Burrows & Borowczak, 2017)

Hands-on Learning	Y
Assessed in New Ways	Y
Student Opinion Counts	Y

Burrows and Borowczak (2017) take the view that 'students need explicit instruction on what engineering researchers and practitioners implicitly understand and use daily' and their research used a game-based approach to investigate what CS students know about soft skills. The focus was particularly on the nontechnical/soft skills identified by the Accreditation Board of Engineering and Technology (ABET) in 2014. They conclude: 'Administration and faculty must focus on explicit soft skill implementation – in and out of the classroom – and then assess those soft skills in the teamwork, communication, and management categories.'

16 Methodological teaching system of mathematical foundations of formal languages as a means of fundamentalization of education

(Bushmeleva & Baklashova, 2017)

Hands-on Learning	Y

This paper proposes a system for studying the mathematical foundations of formal languages as an element in the training of specialists in the field of CS. This involves lectures, practical classes, and independent work. Attention is paid both to the theoretical assimilation of the basic concepts of the course and to the acquisition, development and consolidation of competencies and practical skills. Lectures are conducted using multimedia and interactive technologies. There are also group discussions and individual student consultations. Teaching materials include electronic and paper textbooks, an electronic collection of typical assignments, presentations, and intelligence maps. Results obtained during practical work are discussed individually and in interactive conferences. The most interesting variants of solving problems are considered and analysed together; projects are demonstrated in the form of presentations. Mind mapping tools are used to activate the thinking of students, develop memory, the ability to structure information, effectively take notes, and annotate lecture material.

17 Exploring active learning approaches to Computer Science classes

(Caceffo et al., 2018)

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y
Develops Independence	Y

Three teaching practices were implemented and compared: lecture-based learning, problem-based learning, and peer instruction. Results indicated that a paradigm shift from traditional teaching was not only expected by students and instructor; it was well received, and had a positive influence on students' learning and motivation. The changes also brought an unwelcome overhead for the instructors, as additional time and effort are required to implement such practices.

Co-funded by the Erasmus+ Programme of the European Union

18 First experiences of teaching quantum computing

(Carrascal et al., 2021**)**

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Student Opinion Counts	Y

This paper proposes an approach for teaching quantum computing which includes (1) using classical object-oriented programming to program a basic quantum simulator; (2) a quantum circuit test with a graphical interface; (3) the programming of real quantum computers with a programming language; and (4) a deep exploration of known quantum algorithms. The approach emphasises practical training, and is aimed at students who do not necessarily have a deep theoretical grounding in Physics or Mathematics, but who do have a knowledge of programming classical computers. After a mandatory introduction that covers the basics, the course makes use of problem-based learning.

19 Teaching ethics and sustainability to Informatics Engineering students, an almost 30 years' experience

(Casañ et al., 2020**)**

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y

This paper presents 29 years of teaching courses on social, environmental, and ethical issues to students of Informatics Engineering. It provides an overview of the different teaching strategies used, with emphasis on the use of collaborative learning to facilitate critical thinking and debate. Strategies include case study sessions and active methodologies. Collaborative approaches include the jigsaw method, think-pair-share, group investigation, and role-playing debates in online forums. Use of wikis to support collaboration has given way to use of Google Drive.

20 The impact of virtualized technology on undergraduate computer networking education

(Chamberlin et al., 2017**)**

Education 2.0	Y
Anytime / Anywhere	Y
Hands-on Learning	Y

A network engineering class is taught to two separate groups of students; one group using physical labs for evaluations and lab work, and the other using virtual networking software. Students are divided into two groups; one group using physical labs for evaluations and lab work, and the other group using virtual networking software. Students assigned to the virtual group built their networks in Cisco Packet Tracer 7, which is available from Cisco Networking Academy as part of their free Packet Tracer 101 course. Enrolment and completion of this introductory course was an early homework assignment for all students. No significant difference was found between the groups in either student performance or perceived confidence in the course material.

Co-funded by the Erasmus+ Programme of the European Union

22 A MATLAB toolbox for spatial audio and signal processing education

(Cobos & Roger, 2019)

Education 4.0	Y
Project-based Learning	Y
Student Opinion Counts	Y

While signal processing is the science behind many technologies used in everyday life, students often feel the practical use of signal processing subjects is obscured due to their intensive theoretical perspective. This paper presents an open-source MATLAB toolbox that facilitates spatial audio and signal processing education by enabling students to focus on the implementation and testing of core algorithms and to experience their own results using real-time processing. The time required for other nonrelevant aspects, such as hardware and driver set-up or user interaction, is significantly reduced. The tool is intended to help students establish links between theoretical concepts and practice in an appealing application scenario.

27 Designing and implementing an environment for software start-up education: patterns and anti-patterns

(Fagerholm et al., 2018)

Project-based Learning	Y
Hands-on Learning	Y
Develops Independence	Y

The authors consider students as prospective entrepreneurs, as well as potential employees in modern, start-up-like intrapreneurship environments within established companies. As the abilities needed in start-ups are not among those traditionally taught in universities, new knowledge and skills are required to prepare students for these volatile environments. This paper reports on experiences gained during seven years of teaching start-up knowledge and skills. The authors developed the Software Factory, an educational environment for experiential, project-based learning. They describe ways of designing, implementing and operating physical environments, curricula and teaching materials, as well as planning interventions that may be required for project-based start-up education.

35 Academic approach to transform organisations: one engineer at a time

(Juárez et al., 2020)

Education 4.0	Y
Personalised	Y
Project-based Learning	Y
Hands-on Learning	Y
Develops Independence	Y

This paper presents the design and implementation of a learning experience in undergraduate education with focus on development of nine competencies related to software projects. These competencies are related to solution delivery, quality assurance, teamwork, project management, communication, ethics, and leadership. The initiative started with a learning by-doing approach simulating a software development team of about five to nine team members to produce a software solution for a real client. The method has been developed over time, and the challenge for students is now ‘to consolidate a highly qualified and internationally competitive Information Technology

Co-funded by the Erasmus+ Programme of the European Union

department capable of managing and developing complex software development projects'. For the IT department to be considered consolidated, all the projects selected by the department should delight stakeholders and must be completed within time, scope, and cost.

37 Teaching modern multithreading in CS2 with actors

(Lewis & Lacher, 2020**)**

Project-based Learning	Y

Teaching topics in parallel and distributed computing in a hands-on manner is challenging. The authors outline their use of the actor model, which is very popular in industry and runs some of the most important software today. Their approach includes a semester-long project that involves the use of these concepts to help solidify student understanding and present student feedback on the project and approach. The primary work for students in this course comes from two projects that they develop individually throughout the semester. Both produce networked, multi-player games that give students significant creative leeway. One of these is a text-based, multi-user game modelled on the multi-user dungeons (MUDs) that were predecessors of games such as World of Warcraft. Students are encouraged to spend some time playing one of the few remaining active MUDs, in order to become familiar with the style of play. They must then develop their game by working towards eight checkpoints, a challenging process that is designed to help them understand the actor model.

38 A top-down approach to teaching web development in the cloud

(Liang & Chapa-Martell, 2018**)**

Education 4.0	Y
Anytime / Anywhere	Y
Project-based Learning	Y
Hands-on Learning	Y
Develops Independence	Y

The authors propose a top-down approach to teaching app development. This has three phases: preparation, web technology tutorial, and capstone project. In the preparation phase, the educator selects a sample web application, extracts the technological components of this application, and select core technical topics for tutorials. The sample application creates a social context for the course as a whole. The tutorial phase consists of a series of lectures and hands-on tasks that help students learn web technologies used for implementing the sample application, with a focus on how disparate technologies can be integrated and related to each other. In the capstone project phase, students work in teams to implement their own web applications. This involves them go through a full cycle of web development, starting from conceptualising and planning, moving on to implementation, usability testing, and then creating documentation. Students can be supported in this phase by being taught the basics of relevant approaches to project management, including Agile approaches such as Scrum. The capstone phase allows students to learn from trial and error and see the deeper consequences of their actions. It also requires students to work together to accomplish a shared goal and provides opportunities to develop skills in interpersonal communication and project management.

Co-funded by the Erasmus+ Programme of the European Union

66 Aligning professional skills and active learning methods: an application for information and communications technology engineering

(Ariadna Llorens et al., 2017**)**

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y

This study proposes a model for training Engineering professionals based on active learning methods that are expected to facilitate the acquisition of the professional skills most highly valued in the information and communications technology (ICT) market. Conclusions are based on a Delphi study which brings together the opinions of ten experts from different fields of ICT. The paper identifies seven teaching methods and links them with relevant skills that they are considered to enhance the most: lectures (commitment to learning), case studies (ability to find information), problem-solving exercises (decisiveness), problem-based learning (capacity for teamwork, and decisiveness), project-based learning (ability to find information, and innovation), cooperative learning (capacity for teamwork) and learning contracts (commitment to learning.

39 Work in progress: Task-centric holistic teaching approach to teaching programming with Java

(Mäkiö et al., 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Project-based Learning	Y
Hands-on Learning	Y
Develops Independence	Y

This paper examines the skill gap between graduates and industry expectations, and existing didactic approaches in CS education. It goes on to explain how a Java programming course has been taught using a task-centric holistic agile teaching approach (T-CHAT) to enhance both technical skills and transferable skills in students. It addresses improvement of methodical, social and personal competencies of students along with the development of disciplinary knowledge and skills. T-CHAT integrates five pedagogical approaches: 1) perceptional teaching (moving from observations to explanatory models), 2) project-based learning, 3) problem-based learning, 4) research-based learning, and 5) face-to-face teaching The aim is to make learning more efficient by varying these pedagogical methods in an agile manner, according to the changing needs of students. To ensure efficient guidance and support, there are weekly formative assessments of student homework and student feedback is given immediately after lectures.

51 Distinctive approaches to Computer Graphics education

(Santos et al., 2018**)**

Education 4.0	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y

Co-funded by the Erasmus+ Programme of the European Union

This paper describes educational experiments that tackled problems in Computer Graphics education related to (1) arts and how they can be introduced in technically oriented courses, (2) whether or not a visualisation course in game development curricula would be advantageous and (3) the role of research-oriented activities in undergraduate Computer Graphics. The paper reports four very different initiatives. In the first students are acquainted early on with artistic ways of creation in order to apply this acquired knowledge in various computer graphics courses. In the second, eye-tracking is used to generate data from games. In the third, students carry out a research and development oriented collaborative research project. In the fourth, undergraduates are brought together across Europe in a research seminar programme with 'top experts'. This seminar has multiple interesting elements, including a storytelling workshop.

54 Teaching mobile application development through lectures, interactive tutorials, and pair programming

(Seyam & McCrickard, 2016**)**

Education 4.0	Y
Project-based Learning	Y
Hands-on Learning	Y

This paper probes unique challenges for Pair Programming (PP) when used in mobile software design classes with 53 students, focusing on five mobile design topics discussed in five distinct sessions: dealing with interface and data management, using camera, handling multi-device connectivity, using sensors and collecting GPS data, and using microphones and speakers. The findings indicate that when using PP in a mobile context, there were several issues related to mobile UI/UX, sensors, and multi-device connectivity which introduced unique challenges to students in mobile development classes. PP seemed to provide a reasonable approach to handle such challenges, though with changes put in place.

Co-funded by the Erasmus+ Programme of the European Union

10.3 Appendix A3 Detailed descriptions of included Full EDU 4.0 studies

05 The efficacy of Facebook in teaching and learning: studied via content analysis of web log data

(Alsaif et al., 2019**)**

Education 4.0	Y
Anytime / Anywhere	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Develops Independence	Y

This study investigated the effectiveness of Facebook use in a CS course in Saudi Arabia. A Facebook group was used to facilitate collaborative learning on a course that included an extended group project. Students were assigned to groups of four to five. They used Facebook posts and live chat to discuss their project with other group members or the whole class. Students engaged effectively as on a daily base they created course materials, asked questions, replied to other posts, gave advice and feedback to each other and interacted with the lecturer's posts. They also collaborated on group project tasks, using posts, announcements, comments, feedback and questions. They participated in creating most of the course content in the Facebook group by uploading explanatory videos, photos, PDFs, Word files, diagrams and PowerPoint files. The results showed that the students' positive attitude has been enhanced when utilising Facebook and Web 2.0 tools in their learning activities.

10 A sociocultural approach to using social networking sites as learning tools

(Borge et al., 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

A sociocultural activity design (SCAD) model is proposed and tested in an undergraduate HCI course. Teams of students, altogether 38 undergraduate students, worked on a design challenge each week as a means of applying what they had learned. They were also asked to post on social-networking platform Yammer once a week for 12 weeks to discuss course topics. Discussion-based social networking tools were used to support the development of an online community of learners. Findings suggest that use of the SCAD model facilitated processes associated with a community of learners, as students took over responsibility for the discussions over time, maintained strong connections with multiple peers, engaged in meaningful conversations about course content, and increased the sophistication of cognitive activity over time, even after

Co-funded by the Erasmus+ Programme of the European Union

instructor faded from the environment. However, findings also suggest more support is needed for online argumentation practices.

12 Lab4CE: a remote laboratory for computer education

(Broisin et al., 2017)

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Hands-on Learning	Y
Develops Independence	Y

A remote laboratory, the Lab4CE environment, was used to create a distributed, modular and flexible online learning environment to integrate a set of scaffolding tools and services intended for instructors and learners. Visualisation tools were used to create online and remote laboratories so students could engage in practical activities online. Collaboration tools and awareness artifacts intended for learners promote engagement in remote practical activities. The environment supports human tutoring through the synchronous communication tool, but also through the opportunity offered to users to share a practical session; help can thus be offered by peers, and/or requested if a tutor happens to be online. An exploratory study conducted with 139 undergraduate students enrolled in the first year of a CS degree suggests a positive effect of the framework on learners' engagement when they come to practice system administration, and reveals a significant positive correlation between students' activity within the system and students' learning achievement.

13 An educational game for investigating verification accuracy in software tests

(Buffardi & Valdivia, 2018)

Education 4.0	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

This paper describes a pedagogical technique for introducing unit testing in Software Engineering courses. The Bug Hide-and-Seek educational game reinforces testing principles by requiring students to develop some correct solutions as well as some other solutions that intentionally contain bugs. While developing the correct and buggy solutions, students also write corresponding tests that should identify whether each solution contains bugs or exhibits acceptable behaviour. The first goal of the game is to hide a clever bug that will trick other students' tests into passing the implementation, despite the hidden bug. The second goal is to write thorough tests that can accurately tell correct from incorrect software behaviour A pilot study was conducted with 87 students for preliminary analysis of two variations (between-subject and within-subject) of the Bug Hide-and-Seek game, while comparing their trade-offs.

Co-funded by the Erasmus+ Programme of the European Union

21 Knowledge construction in Computer Science and Engineering when learning through making

(Charlton & Avramides, 2016)

Education 4.0	Y
Personalised	Y
Assessed in New Ways	Y
Hands-on Learning	Y
Develops Independence	Y

This study examines learning by making with students exploring STEM using a constructionist approach with a particular focus on CS and Engineering. The use of the Internet of Things as a technology-enhanced learning tool created the learning conditions. These were (a) collaborative: no individual had the knowledge to complete the project alone, (b) problem-based: no off-the-shelf solution was used, and (c) multidisciplinary: the learning context pushed the boundaries across subjects. The start of the collaboration involved thinking about smart city projects. In groups they brainstormed ideas to investigate after attending a mini-workshop. They attended a final two-day hack-event where their ideas were prototyped and finally presented at the London Festival of Education.

23 Redesigning an introductory programming course to facilitate effective student learning: a case study

(Corritore & Love, 2020)

Education 4.0	Y
Anytime / Anywhere	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

A new course was built with the intention of improving student performance, encouraging students to internalise relevant concepts and skills, and increasing ownership of learning. Student Ownership of Learning is not new – the authors traced it back to Dewey in 1916 – but its use in this context, together with a flipped pedagogy is innovative. The course incorporated students' questions and ideas, refining and adjusting these over time, leading to new insights. Students created goals, evaluated movement towards them, and applied learning in order to meet them.

30 Interdisciplinary projects in the academic studio

(Gestwicki & McNely, 2016)

Education 4.0	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Student Opinion Counts	Y

Co-funded by the Erasmus+ Programme of the European Union

Develops Independence	Y

The academic studio is a model for university courses that brings students, faculty, and community partners together to engage in product-oriented and authentically academic inquiry. This model was developed for students to build disciplinary knowledge and skills; and to develop multidisciplinary collaboration skills including communication, coordination, estimation, and empathy. It was also designed to connect students, faculty, and the university to the wider community; and to address interdisciplinary research questions. The academic studio incorporates agile software development techniques with situated learning theories, project-based pedagogy, a multidisciplinary, self-organizing, cross-functional team of students, working in collaboration with one or more faculty mentors and community partners to investigate an academic question. Students and mentors work together in a dedicated place and time, following incremental and iterative processes, to create a product that embodies their inquiry.

32 An instructional feedback technique for teaching project management tools aligned with PMBOK

(Gonçalves et al., 2018**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Develops Independence	Y

This article presents the design and evaluation of an instructional feedback technique, designed to enhance the teaching of project management tools in higher Computing education. It provides immediate and automatic feedback messages to students as they interact with the functionalities of a project management tool. Messages include examples and suggestions about how to improve the project charter and the project plan developed using the tool. The evaluation found the instructional feedback technique was generally welcomed by students.

33 A distance learning curriculum on pervasive computing

(Goumopoulos et al., 2017**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Develops Independence	Y

This paper reports the authors' experience in organising, managing and teaching a pervasive computing curriculum using distance learning. They address distance education challenges through advanced educational material, intelligent tutoring systems, and virtual laboratories. Students engage in small-scale projects and implement both software and hardware prototypes.

Co-funded by the Erasmus+ Programme of the European Union

Tools such as Lego Mindstorms, Phidgets and Arduino are used to enable students to build simple prototypes. They also carry out laboratory exercises remotely, with the results visualised in real time using webcams.

36 Theorem provers as a learning tool in theory of computation

(Knobelsdorf et al., 2017)

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Hands-on Learning	Y
Data Interpretation	Y
Develops Independence	Y

Theorem provers are used to create and check formal proofs. The authors investigate whether an interactive theorem prover could be used as a learning tool to support students to create formal proofs as well as formally reason within a proof. The intention is to scaffold students' learning and investigate relevant questions related to formal proving practices and the development of domain-specific competencies relating to the theory of computation. The tool Coq was chosen because it provides immediate, individual feedback. This contrasts with the common practice of creating proofs with pen and paper and receiving feedback days or weeks later. Working with Coq and creating proofs in a way akin to programming-debugging is an approach that is close to computing culture and has the potential to be motivating and engaging.

40 Game Lab: a practical learning approach for game development

(Munkvold, 2017)

Education 4.0	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Develops Independence	Y

The Game Lab course puts students into a situation where they have to act as a game company, taking on realistic industry challenges, working with actual customers, and having to answer to an executive committee on a weekly basis to report on their progress. To get good grades, students must prove they are active in the team through clear tasks and responsibility descriptions, notes on working hours, and weekly follow-ups. The social element of working together, the freedom to work on projects that are perceived as interesting and relevant, seeing the final product, being able to specialise, constructive feedback, and industry relevance are all identified as elements of the course that motivate students.

43 Can we use the flipped classroom model to teach black-box testing to computer students?

(Paschoal et al., 2019)

Education 4.0	Y
Anytime / Anywhere	Y

Co-funded by the Erasmus+ Programme of the European Union

Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

Software Engineering courses often cover software testing in a theoretical way, with little time to put the methods into practice. In other cases, students are actively engaged in the classroom, solving problems, developing real projects, or dealing with real cases. The flipped classroom approach can be used to support this model. This paper analyses student learning gains when the flipped model is used and considers the workload implications. Students in the flipped classroom group acquired more knowledge than those taught in a traditional way, and also spent more time on their learning.

43 Comparison of traditional lecture and flipped classroom for teaching programming

(Pawelczak, 2017**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

The authors used a flipped classroom approach on an elective advanced programming course. They compared the new course design and its effects on learning, teaching, and course preparation with the former traditional lecture. They found students seemed more motivated when they could work with the course material at times of their choosing. The students reported they enjoyed the flipped classroom approach very much. Examination results and workload do not differ between approach. The educator found students were better prepared in the flipped classroom and discussions could be established on a higher level. However, the effort involved in setting up the flipped classroom was very high and course materials has to be updated frequently as programming languages evolve.

44 Introductory game development course: a mix of programming and art

(Peng, 2015**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y

Co-funded by the Erasmus+ Programme of the European Union

Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y

The paper describes an introductory course on video game development, which is designed to teach students all aspects in the game production cycle in the undergraduate CS curriculum. 36 students took part in this course. Learners studied theoretical as well as applied aspects of extensible application architectures, discuss existing and emerging scripting languages, explore and utilise current applications, and create extensions to these applications by programming in scripting languages. They work on four projects: design and implement a 2D game; create, rig and animate a 3D model; and, finally, create a 3D game.

46 Computational creativity exercises: an avenue for promoting learning in Computer Science

(Peteranetz et al., 2017**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Student Opinion Counts	Y
Develops Independence	Y

Complex problems in CS and Engineering, require thinking that is systematic, flexible, computational and creative, a combination referred to as computational creativity. This approach views computational thinking and creative thinking as separate but compatible cognitive tools that expand the ways knowledge and skills can be applied to problem-solving situations. This paper introduces a series of Computational Creativity Exercises that were used to introduce computational creativity to students. These exercises challenge students to apply abstract computational thinking principles – such as algorithmic design and abstraction – to contexts other than CS. The exercises also require students to capture and report aspects of the problem-solving process, engage multiple senses, use imaginative thought, and interact with a variety of peers. Examples of the exercises are available online http://cse.unl.edu/agents/ic2think/software.php

48 Peer learning assistants in undergraduate Computer Science courses

(Pivkina, 2016**)**

Education 4.0	Y
Personalised	Y
Project-based Learning	Y
Hands-on Learning	Y
Assessed in New Ways	Y
Develops Independence	Y

The paper describes the experience of using undergraduate students as peer learning assistants (PLAs) in different undergraduate CS courses which also have graduate teaching assistants. Depending on the course, peer learning assistants carry out some of the following tasks: holding

Co-funded by the Erasmus+ Programme of the European Union

office hours, helping with labs and tutorials, and facilitating student group work in class. The use of PLAs resulted in statistically significant increase in student course grades and decrease in the number of non-passing grades. Student experiences with PLAs were very positive; they preferred to seek help from the PLAs rather than from the teaching assistants or instructor. PLAs report that the experience of PLAs was beneficial to them.

49 Learning UI functional design principles through simulation with feedback

(Ruiz et al., 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Develops Independence	Y

Teaching user interface (UI) design is challenging and there is little teaching support. This paper investigates the benefits of a feedback-enriched simulation environment called FENIkS for learning fundamental UI design principles. FENIkS is a model-driven educational environment that makes use of simulation as learning support by generating a UI and the underlying application. The generated application and UI contain feedback that shows whether the prototype generated is compliant with key design principles and why the principles are considered well applied or not. The study showed an improvement in learning of UI design principles when using this approach.

50 Effective assessment of Computer Science capstone projects and student outcomes

(Salem et al., 2020**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Student Opinion Counts	Y
Develops Independence	Y

A capstone project requires creativity, critical thinking, and advanced problem-solving skills. They also enable students to prove their abilities, demonstrate their attained skills, and carry out a significant project in their field. This paper presents an assessment framework for capstone courses that allows for sound evaluations of the performance of students and project qualities; as well as assessing student outcomes. The framework includes criteria, indicators, extensive analytic rubrics, and an aggregate statistical formulation. It is aligned with the ABET (previously the accreditation board for engineering and technology) student outcomes for CS programmes, which include soft skills such as teamwork and taking responsibility.

Co-funded by the Erasmus+ Programme of the European Union

55 Teaching mobile application development through lectures, interactive tutorials, and pair programming

(Seyam et al., 2016)

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Hands-on Learning	Y
Data Interpretation	Y
Student Opinion Counts	Y
Develops Independence	Y

This paper examines three teaching approaches employed on a mobile software development course: lectures, interactive tutorials, and Pair Programming. Lectures were used to introduce topics and explore underlying theories of development. The lectures included time for questions, but otherwise did not have an active learning component. Interactive tutorials presented applied development approaches, then explored their use in hands-on demos. Pair Programming was used as a role-based approach to learning new programming concepts. Results showed that repetition of topics is important for mastery of the topics. Foundational theories seem well suited for lectures, while programming concepts work better in active learning situations. Additional learning took place through office hours, online question forums, and individual and group online exploration. The findings suggest specific approaches to teaching challenging and unique mobile software development topics as well as a general approach to identifying ways to distribute learning objectives across lectures, interactive tutorials, and Pair Programming sessions.

57 Adapting the studio based learning methodology to computer science education

(P. A. Silva et al., 2017)

Education 4.0	Y
Personalised	Y
Choice in How to Learn	Y
Hands-on Learning	Y
Data Interpretation	Y
Student Opinion Counts	Y
Develops Independence	Y

Studio-based learning was used as a pedagogical approach in an online introductory CS course. In this approach, students are given non-trivial problems for which they have to design and implement computational solutions. Because these problems can be solved in different ways, students have to consider alternate solutions, choose the best, and justify their choice. They articulate their solutions and justifications to others for peer review, feedback, and discussion. Peers and the course instructor provide comments and criticisms. Students are given the opportunity to respond to this feedback and modify their solutions appropriately. A benefit of this approach was the confidence the students gained from giving and providing quality feedback to their fellow classmates.

59 Teaching parallel and distributed computing concepts in simulation with WRENCH

(Tanaka et al., 2019)

Co-funded by the Erasmus+ Programme of the European Union

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Develops Independence	Y

Teaching topics related to high performance computing and parallel and distributed computing in a hands-on manner is challenging, especially at introductory undergraduate levels. Securing access to a platform on which students can learn via hands-on activities is not always possible. Any platform provided imposes constraints on which learning objectives can be achieved. To address these challenges, the authors developed a set of pedagogic activities that can be integrated within university courses. The activities use simulations so that students can gain hands-on experience. The simulations (which use the WRENCH simulation framework) used in these activities provide both metrics and visualisations of executions through which students can empirically verify their answers to relevant questions. Students can also use the simulations to explore complex design spaces and acquire knowledge independently, possibly with instructor-provided scaffolding. Capstone activities consist of case-studies in which students apply what they have learned in previous activities to solve real-world problems.

60 Towards applying Keller's ARCS model and learning by doing strategy in classroom courses

(Tlili et al., 2017)

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Develops Independence	Y

This paper presents a pilot educational games development course, which involves a learning-by-doing strategy. It is also based on Keller's attention / relevance / confidence / satisfaction (ARCS) motivational model. This model emphasises the importance of grabbing students' attention, highlighting the relevance of what they are learning, making them feel confident they can succeed and ensuring they are satisfied with the experience. This approach improved student motivation, kept them active, and helped them gain the needed technical skills to develop educational games.

61 Collaboration and fuzzy-modeled personalization for mobile game-based learning in higher education

(Troussas et al., 2020)

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y

Co-funded by the Erasmus+ Programme of the European Union

Hands-on Learning	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

Mobile game-based learning can foster student motivation, increasing engagement in the educational process. This paper investigates how mobile learning and game-based learning can be used in HE settings. The authors designed and implemented Quiz Time!, an intelligent mobile game-based learning application for assessing and advancing learners' knowledge in the programming language C#. The application employs an assessing knowledge module for testing the knowledge of learners, a recommendation module for proposing personalised collaboration, a dynamic fuzzy logic-based advice generator for tailored assistance to learners' profile and misconceptions, and a cognitive learner modeler that supports the other modules. The study concluded that incorporating personalisation and collaboration in mobile game-based learning can help students increase their knowledge level.

64 Teaching professional morality & ethics to undergraduate Computer Science students through cognitive apprenticeships & case studies: experiences in CS-HU 130 'Foundational Values'

(Winiecki & Salzman, 2019**)**

Education 4.0	Y
Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Develops Independence	Y

The CS department at Boise State University and its industry partners have committed to addressing systemic bias in the field, across the undergraduate curriculum and into student internships in local industries. The first CS course required of all students is 'Foundational Values', which exposes students to case studies that document breaches in inclusion, diversity and social justice in CS education, professional practice, and in the products of CS work. Students are provided with a problem-solving rubric and cognitive apprenticeship-type support in using that rubric to systematically analyse the problem(s) and offer systemic solutions to them. Other courses in the curriculum are being developed and updated to include modules which connect what has historically been considered to be 'purely technical' content with social science content supporting inclusion, diversity and justice. The aim is to help produce a generation of computer scientists who understand the value in collaborating with social science experts and in this process becoming knowledgeable about the micro- and macro-societal effects of their actions in computing. Students are supported to develop analytical skills and habits of mind that can be put to use in improving their learning and working environments.

65 Mixed approaches to CS0: exploring topic and pedagogy variance after six years of CS0

(Wood et al., 2018**)**

Education 4.0	Y

Co-funded by the Erasmus+ Programme of the European Union

Anytime / Anywhere	Y
Personalised	Y
Choice in How to Learn	Y
Project-based Learning	Y
Hands-on Learning	Y
Data Interpretation	Y
Assessed in New Ways	Y
Student Opinion Counts	Y
Develops Independence	Y

Computing majors at CalPoly select from a variety of CS0 courses to start their academic year. The objective is to attract and retain undergraduates who have no prior experience in CS by using authentic problems that demonstrate the relevance and highlight the role of computers in solving real-world problems. The courses offer a variety of thematic flavours that connect with a student's pre-existing interests (such as music or art), but all share the common goal of introducing students to the basics of programming, teamwork, and college-level study. The courses vary considerably in topic matter (including robotics, gaming, music, computational art, mobile apps, and security) and in pedagogical approach (for example, principles of design, project-based student driven learning, and traditional topic-based programming modules). The introduction of this range of courses has increased students' commitment to their major and their success in follow-on classes. Evaluations suggest that the existence and goal of the course matter more than the specific content, with all subtopics and pedagogical approaches performing well.

Co-funded by the
Erasmus+ Programme
of the European Union

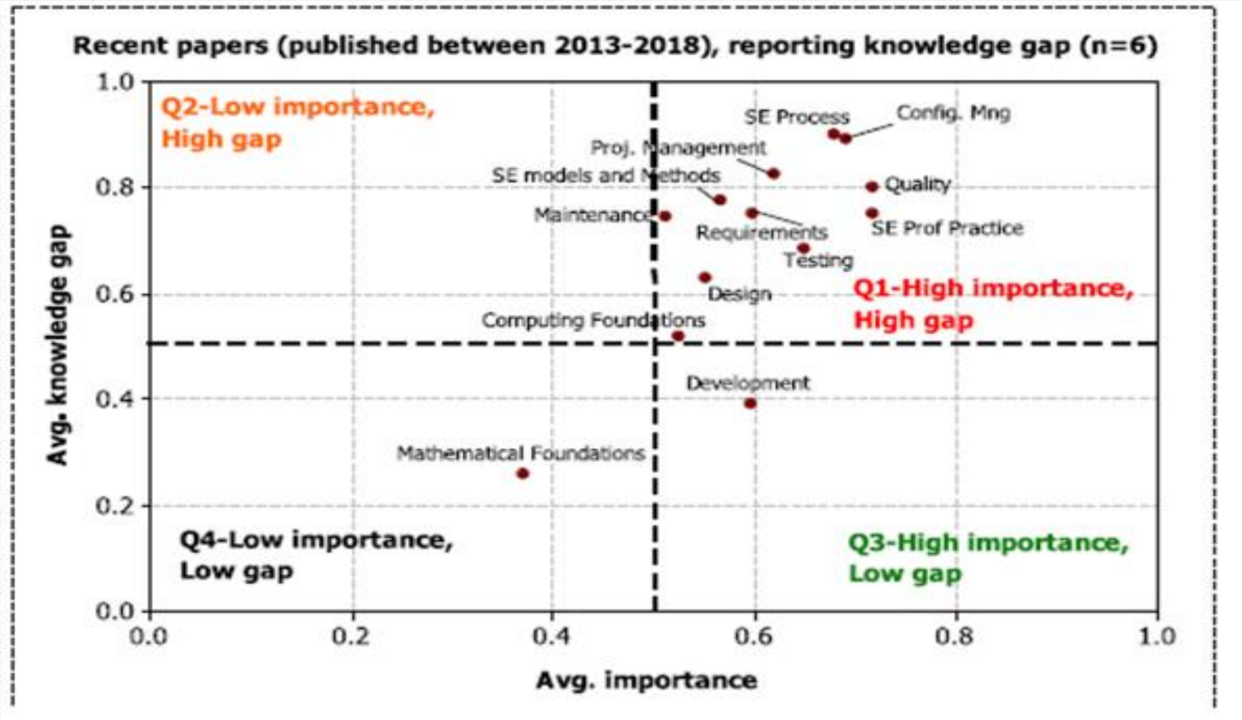
10.4 Appendix B Methodology used for focus groups

The methodology used during the focus groups included several sections to facilitate the discussion between participants as follows:

1. Tell us a little about your company, and what your specific role is in this company (10 min) (ice breaker activity)
   - Introduce yourself
   - How large is your unit \ company?
   - How many CS graduates have you recruited in the last 2 years?
   - What is your definition of Computer Science? What is and what is not
2. CS graduates' knowledge, skills and competences when starting at your company (10 min)
   - What are the expectations you have when recruiting new graduate students in Computer Science?
   - What is the key knowledge, skills and competences the company would benefit from that students are Missing? \ Bringing?
   - What are the key problems CS graduates face when integrating into the company?
   - To summarise: What are your overall impressions of these CS graduates?
3. How knowledge, skills and competences are affecting the company day to day work (10 min)
   - What is the training they receive when joining the company?
   - What are the roles and responsibilities they take from the starting?
   - How do they interact with clients and customers?
4. How knowledge, skills and competences could be acquired before and integrated within appropriate teaching methods (10 min)
   - How the knowledge, skills and competences could be acquired before?
   - Which innovative methods could be included?
5. In the following visualisation of a review of key skills required in software engineering (10 min):
   - Do you agree with this visualisation, or do you think some key skills are missing?

Co-funded by the Erasmus+ Programme of the European Union

Figure 9 From ([Garousi et al., 2019](#)) used for the focus groups



Co-funded by the Erasmus+ Programme of the European Union

10.5 Appendix C: Search terms for literature review

Search criteria: ScienceDirect (34 results)

- “computer science” AND education AND teaching AND pedagogy AND ("undergraduate" OR "postgraduate")
- Title, abstract, or keywords must include “computer science”
- 2016-2020

Search criteria: Wiley Online Library (10 results)

2016-2020: 10 results for"“computer+science”" in Keywords and ""education" OR "teaching"" anywhere and ""undergraduate" OR "postgraduate""

Search criteria: Scopus (110 document results)

AUTHKEY (computer+science) AND education AND teaching AND pedagogy AND ("undergraduate" OR "postgraduate") AND (LIMIT-TO (PUBYEAR , 2020) OR LIMIT-TO (PUBYEAR , 2019) OR LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016)) AND (LIMIT-TO (LANGUAGE , "English"))

Search criteria: Web of Science (90 results)

You searched for: (AK=computer science AND AB=(Education OR Teaching) AND AB=(Undergraduate or Postgraduate)) AND LANGUAGE: (English)

Timespan: 2016-2020.

Co-funded by the
Erasmus+ Programme
of the European Union