*sustainability* MDPI

*Article*

# Sustainable Approaches for Accelerated Learning

**Richard Glassey** *,† and **Olle Bälter** †

Department of Media Technology and Interaction Design, KTH Royal Institute of Technology,
SE-100 44 Stockholm, Sweden; ob1@kth.se
* Correspondence: glassey@kth.se
† These authors contributed equally to this work.

**Abstract:** Sustainable education does not yet have a widely accepted definition in the literature. In this work, we start from the Sustainable Development Goal of Quality Education for All (SDG4) and interpret sustainable education as increasing the quality of learning whilst conserving the resources required to produce and deliver it. From this interpretation, we argue that one path towards realising sustainable education is through the identification of teaching practices that satisfy these conditions of increased quality whilst conserving resources. We present an overview of four case studies, where the conditions for sustainable education are demonstrated through the effective use of people, processes and technologies. Each case represents an intervention that was made to improve the quality of education within an intensive three-month project, which trained immigrants to be employable in the IT industry as junior software developers. Whilst the interventions are independent and unique, they are connected by the themes of quality improvement and resource conservation. In isolation, each specific case produced improvements for both teachers and students; however, it is by combining such approaches that we can start to realise the path towards sustainable education that will help lead to a better quality of education for all. The findings of this work suggest that quality education does not come at the cost of increased resource demands; rather, approaches exist that can be considered to satisfy the conditions for sustainable education.

**Keywords:** sustainability; education; digitalisation; accelerated learning

check for updates

**Citation:** Glassey, R.; Bälter, O. Sustainable Approaches for Accelerated Learning. *Sustainability* **2021**, *13*, 11994. https://doi.org/10.3390/su132111994

Academic Editors: Åke Grönlund and Olga Viberg

Received: 15 September 2021
Accepted: 24 October 2021
Published: 29 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## 1. Introduction

As one of the Sustainable Development Goals (SDGs), Quality Education for all (SDG4) was not on a good path, even before the COVID-19 pandemic. The pandemic made this worse when school closures kept 90% of all students out of school, with the most negative impact being among the poorest [1]. The poorest of the population are also the ones with least access to online education [1], but still online education remains one hope for achieving SDG4 [2,3]. To overcome the lack of access to online education, investments in infrastructure are necessary; a part of this is the development of online learning material. This is challenging in a field that is slow to change while costs are rising in an alarming way [4]. Simply generating online learning material is no panacea, as quality takes time and the resources to do so are often stretched or lacking altogether. The Raccoon Gang (https://raccoongang.com/blog/how-much-does-it-cost-create-online-course/, accessed on 1 October 2021) estimated that it takes between 90 h to 240 h to produce one hour of ready online-learning content, so cost awareness is essential. To improve this situation, we need to better understand what sustainable education is as a concept, and what the necessary conditions for sustainable education might be.

First, there is no universally agreed upon definition for sustainable education. Whilst sustainability has become a widely used term, the specific relationship to education has not yet been firmly established. Second, the conditions for sustainable education are not necessarily radically new approaches to teaching; rather, they might exist already in the practice of teachers or in the ideas of pedagogical researchers. The aim of this article is to bring together examples of approaches to teaching that may help contribute towards a better understanding of sustainable education. In this work, we provide a review of four case studies of how we have reduced costs for learning material development and learning management, while striving to enhance the learning outcomes. Individually, these case studies are interesting examples of teaching practice; however, by taking a wider perspective over all of them, we can start to approach the research question of what sustainable education practices should be like.

The context for these cases is within an intensive three-month education period for newly arrived immigrants to Sweden who are struggling to integrate into the local employment market. As such, quality of education (learning outcomes) and the resources spent (time/effort) are critical aspects of concern. However, the SDA project was rapidly pulled together to address the emerging societal issues that were occurring in Sweden in response to the European migrant crisis. In its earliest iterations, it created the educational program by composing and compressing existing courses that teachers were willing to contribute. Over time and successive iterations, the freedom to experiment within the project was exploited to test pedagogical ideas in a practical environment in order to maintain the quality of learning whilst conserving the resources as much as possible, all without compromising the experience of the participants nor reducing the chance of finding meaningful employment. As such, this context provides a good source to investigate what practices might be good examples of approaches to sustainable education.

The article is structured as follows: first, we define our interpretation of sustainable education and the relevant pedagogical ideas interspersed throughout the SDA project; then, we present the four individual cases where major interventions were made in the project in order to promote sustainable learning; finally, we conclude with a discussion on what we believe these cases tell us about sustainable education and how to work towards achieving it more broadly in global education.

## 2. Background

### *2.1. Sustainable Education*

As we have failed to find a widely accepted definition of sustainable education, we will approach this issue by addressing the four aims in the sustainability goal of Quality Education for all [1]:

1. Help countries in mobilizing resources and implementing innovative and context-appropriate solutions to provide education remotely, leveraging high-tech, low-tech and no-tech approaches;
2. Seek equitable solutions and universal access;
3. Ensure coordinated responses and avoid overlapping efforts;
4. Facilitate the return of students to school when they reopen to avoid an upsurge in dropout rates.

In our context, we interpret these aims as improving the following:

1. Quality of remote accessible learning material;
2. Minimally needed resources;
3. Reduced resource waste by reusing learning material and simplifying the creation of learning material;
4. Quality of learning material and teaching such that the physical return to school is merely one of the equally effective options.

These aims can be further refined to *increasing the quality of learning while conserving resources*, which is the interpretation of *sustainable education* that we will use in this paper.

### *2.2. Question-Based Learning*

The Open Learning Initiative (OLI), is an open educational resource project initiated at Carnegie Mellon University (CMU) in 2002. This project creates and shares innovative learning material with several hundred colleges and universities and more than 1000 instructors (Greg Bunyea, Learning Engineer Open Learning Initiative. Personal communication.). One of the innovative parts is using basic Bayesian hierarchical models to predict student mastery [5]. Others include question-based learning, where questions with feedback are littered over the learning material in order to stimulate the students. This type of active learning is six times more efficient than reading and watching videos [6]. Used repeatedly, and refined with the data that each course instance brings, this method can be very effective. In a randomised case-control study, learning time was reduced by 50%, with maintained learning outcomes compared to the parallel traditional course offering [7]. We have partly repeated this study on new developed material and managed to reduce the learning time by 25%, with maintained learning outcomes [8].

### 2.3. Peer Teaching

One remedy to the resource problem is to use peer teaching (PT) [9], that is, transfer the responsibility of teaching to the students. This technique is also referred to as learning by teaching [10], peer teaching [11], and peer tutoring [12]. PT has been used for a long time [9]. One argument comes from Vygotsky, who concluded that students learn as much from their peers as they do from their teachers [13]. A review study praises the potential for both the student teacher and the student learner [9]. PT can contribute significantly to socio-psychological needs among students: active learning, increased cooperation, motivation, self-confidence and self-esteem. When PT was introduced into introductory computer science classes, it led to dramatically reduced dropout and higher grades [14].

### 2.4. Accelerated Learning

The OLI methodology mentioned above is one example of *accelerated learning* (also referred to as intensive learning). Accelerated learning is planned to be faster is some way than traditional learning by either reducing the number of study hours per week, or the number of weeks, or both, while still aiming for at least the same learning outcomes [15]. While accelerated learning can be criticized for being shallow, there are components that are common for successful accelerated learning programs for adults [16]:

1. Adult students can build the new learning on the experiences that they already have, which makes it easier/faster to learn.
2. The learning environment is adapted to the learners' context, such as classroom locations close to the students' workplaces, or today, online, which reduces the overhead in terms of commute.
3. There is a focus on solving real-life problems, building on the learners' experience.

Accelerated learning is one approach to life-long learning, or *reskilling*, the practice of retraining people to take on jobs in new sectors as old ones become obsolete, due to technological development [17], or by (forced) migration as we did in the Software Development Academy (SDA). The three-month programming course SDA was developed as a response to the 2015 crisis to try to handle the large number of migrants from the Syrian war. The purpose was to quickly educate migrants for the IT job market, where there is a substantial shortage of skilled personnel [8,18]. The course material and methods developed for the SDA are used in several of the examples in this paper.

### 2.5. Agile Practices in Education

Efficient adult education requires agile adaptation to the learners, as their varying experience is an advantage for more learning, but these learners might also have knowledge gaps in a way that mainstream college students do not. These challenges can be met with student evaluation of both their learning and the teaching [19], reflection on learning [20], and assessment with immediate feedback [7,21–23]. Agile methods also have a link to education for sustainable development, as a review study concludes that agile methods in education "creates a learning environment favorable for the creation of responsible and sustainable citizens while improving the performance, satisfaction and motivation of both faculty and students" [24].

### *2.6. Learnersourcing*

*Learnersourcing*, a form of crowdsourcing in which learners collectively create content for other learners while performing a learning experience themselves [25], is one way to reduce the workload for teachers creating learning material, without burdening the learners. There are several systems that provide support for learnersourcing, such as PeerWise (https://peerwise.cs.auckland.ac.nz/, accessed on 1 October 2021) and Ripple (http://ripplelearning.org/, accessed on 1 October 2021). PeerWise is a system that supports learners' creation of multiple-choice questions that other learners can answer, rate and comment [26]. It has been used by over 1500 academic institutions and is widely researched [27]. Students at courses using PeerWise answer significantly more questions in the system than is demanded of them [28]. We successfully used PeerWise to create OLI-style questions with feedback [29].

## **3. Context of SDA**

The context for this work is set within intensive education given over three months with the aim to deliver trained IT professionals, who are then coached towards finding full-time employment within the local industry. The Software Development Academy ran for nine iterations from 2017 until 2021. Each iteration recruited 35 students, with 5 teachers, 10 teaching assistants and 2 administrative staff members.

### *3.1. Crisis and the Creation of SDA*

The European migration crisis hit a critical peak in 2015 when an estimated 1.3 million people came to various European countries hoping to claim asylum [30]. Different countries reacted in different ways, particularly with respect to their openness to newcomers. By 2016, this had become a societal issue in Sweden, and academics at KTH began to discuss how best to contribute towards practical solutions. Whilst lectures in Sweden are public events and any citizen is allowed to join, offering this seemed like it would achieve little practical difference and just create disruption for ongoing teaching. A small group of academics suggested another method by creating focused IT training to be delivered on campus but in parallel with regular teaching. The topic of IT was also advantageous, as Sweden was concurrently facing a shortage in skilled IT professionals. Thus, the Software Development Academy was created.

To balance the critical competencies—training skilled IT professionals and finding a secure job for the newly trained participants—KTH teamed up with a recruitment agency in the local area that was also focusing on this particular problem, having had previous success in helping newcomers who were trained nurses and truck mechanics find employment in these domains in Sweden. The final source of information came from engaging with the local IT industry as a partner and gaining a sense of their needs to better tune the education for potential employers. The target was to deliver junior Java developers with either mobile or web development specialisms. The partners and overall process for SDA is shown in Figure 1. In later iterations, Lund University was incorporated as a second site for delivering education to a different geographic region from KTH in Stockholm.

### *3.2. Education, Environment and Employment*

Figure 2 shows the sequence of modules that made up the education. Each iteration began with the programming foundations (PF) module. This course was adapted from the same course that all KTH computer science students receive and is aimed at those with little experience in programming. Towards the end of the PF module, an individual project (IP) started so that students could practice their growing skill set. This module then ran in parallel with the software engineering (SE) module, which covered the more theoretical and team aspects of development. Participants then returned to more advanced programming in the programming in depth (PiD) module, which approximated the typical content of an algorithms and data structures course. The enterprise technologies (ET) module followed with an intensive week of learning about new and important technologies used in modern software development. This served as a ramp to the advanced module, which either focused on mobile application development (e.g., with the Android SDK) or full stack web development (e.g., covering both frontend and backend development with React and Node.js). The education concluded with a competitive group project (with awards for different categories, such as technical excellence, most collaborative team, and most disruptive idea) focused on the needs of a simulated angel investor who wishes to invest in a tech startup in a given domain (e.g., mobile apps, FinTech, and location-aware games). Students followed the SCRUM agile methodology over four weeks before concluding with graduation. Throughout the education, the recruitment agency worked on coaching participants to meet industry expectations and arranged networking events for participants to meet recruiters.

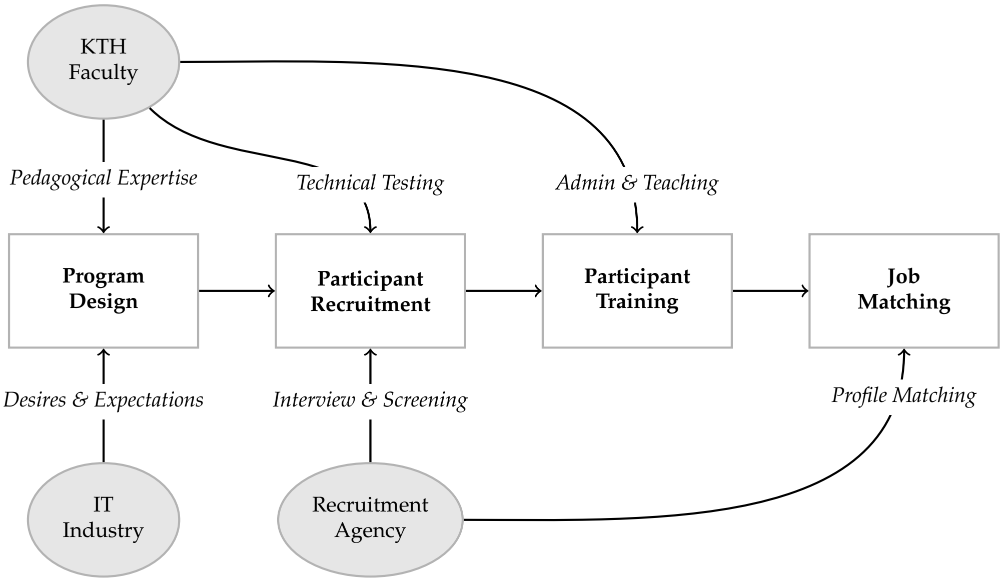


**Figure 1.** High-level overview of the major stages of the SDA project and the inputs of the project partners at each of the stages.

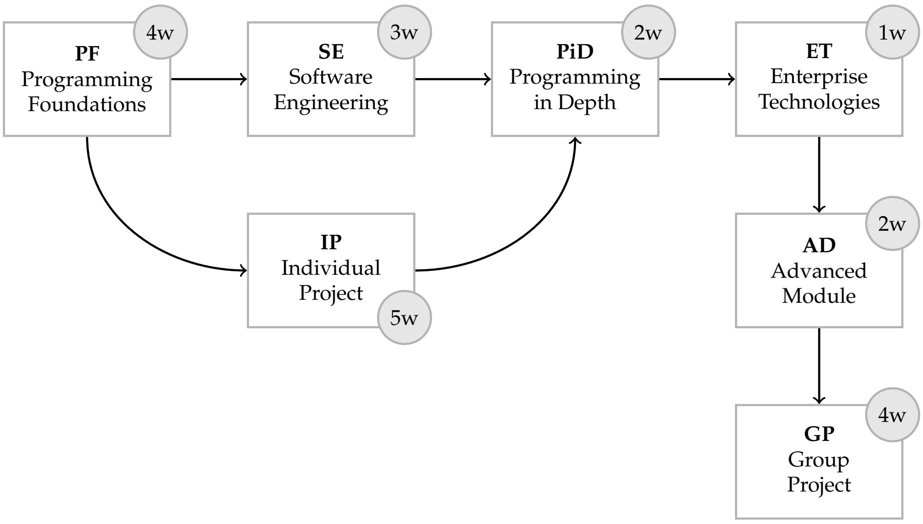

PF Programming Foundations 4w
SE Software Engineering 3w
PiD Programming in Depth 2w
ET Enterprise Technologies 1w
IP Individual Project 5w
AD Advanced Module 2w
GP Group Project 4w

**Figure 2.** The SDA education is organised into seven modules. The individual project (IP) runs in parallel with both programming foundations (PF) and software engineering (SE) modules. Numbers in the white ovals indicate the normal duration of each module in terms of weeks.

In terms of practical arrangements, a dedicated teaching space was reserved only for the use of participants and teachers. Participants were expected to be present for the entire working day, spending this time on the various learning activities that were structured for them. As this was felt to be an important part of the commitment, daily attendance was taken to ensure that participants were present. This was sensed in the early iterations when the majority of dedicated students felt that those who were not in attendance would still obtain the same industry networking opportunities. Teaching assistants (recruited from the local student population with a computer science background) were present throughout the day in case participants had a problem that they wanted to resolve. As the project ran for about 14 weeks, this was quite an intensive setting, but also a place where a community of practice readily formed with little need for intervention.

To give a sense of the impact of the project, the results of the first six iterations are shared below. Due to the pandemic affecting essentially everything (education, coaching, job hunting, general industry appetite to recruitment), the final three iterations are not included.

In terms of participants, 215 were recruited for the six non-pandemic iterations of which 54.8% were women. The drop-out rate was 12.9%. In terms of program completion, 89.0% of the female participants and 86.6% of the male participants finished the program. In terms of employment success (i.e., finding a full-time job in a relevant industry, and not simply more training or unpaid internships), on average, 82.6% of the participants secured an IT job within 5 months after graduating.

### *3.3. Case Overview*

The following four cases highlight different approaches to promoting sustainability that combine the efforts of teachers, students, processes and technology within the SDA project. The research questions, method, results and discussion for each case study are available in previously published work [8,18,29,31], whilst here, we seek to focus on how these cases are good exemplars of sustainable education, where quality is improved whilst conserving the resources used for production.

In the first case [18], agile processes are adopted in order to create a feedback loop that meaningfully engages with students' experience, confidence and knowledge every week, and present the opportunity for teachers to adapt quickly; see Section 4. In the second case [8], question-based learning is used via an online teaching platform with rich feedback that accelerates students' learning without sacrificing quality or performance; see Section 5. In the third case [29], students are engaged in the production process of multiple choice questions, driving a continuous supply of fresh learning material; see Section 6. In the final case [31], a solution for keeping pace with emerging topics is shown by creating learning teams, where students work together to master a new topic, then become the teachers that teach it to the rest of their fellow students; see Section 7. Taken individually, these are novel interventions; however, when blended together through an education process, they become greater than the sum of their parts, encouraging higher confidence and competence in students, involving students through communication and participation, and finally helping students to realise that they can help others to learn. All these cases are examples of sustainable education in the sense that resource use was reduced while the learning outcomes were improved.

## 4. Case One: Agility in Course Management

The first case (see [18] for full details) describes how the pace and dynamics of an intensive education was managed by taking inspiration from the world of agile models in software engineering [32]. Typically, a course is designed, delivered, evaluated and then improved in advance of the next iteration. This maps onto the waterfall model that has long dominated various fields of engineering [33]. For software engineering, this predictable model of a sequence of well-defined stages that lead to the completion of a project did not work well, given the degree of unpredictability with time/effort estimation in projects, coupled with the rapid pace of progress in both hardware and software [34]. Through the 1990s, there was an increasing sense that other methods might work better, culminating in the publication of the Agile Manifesto [35], which sparked an even wider search for more lean and agile methods, which has continued to the present time.

Returning to an educational context, lean and agile methods were proposed as alternative ways to manage the iterative improvement of a course [36,37]. Time and effort costs are prohibitive for teachers that are already burdened by their teaching and administrative duties; however, the benefits to both students and teachers are an attractive counterbalance:

- Students' ongoing evaluation means that they experience benefits during their course.
- Teachers are much more in tune with student experience and can easily increase course evaluation outcomes.
- The course content evolves with progress, rather than drifting behind.
- Small problems are detected quickly and do not become big problems.

### *4.1. Case Context and Motivation*

In the first offering of the SDA project, there was no prior experience available to anticipate problems. Teachers simply compressed and concatenated their existing traditional courses into the block format modules and relied upon their instincts during delivery. Given the daily engagement with students, teachers were able to build up an idea of the student cohort and communicate this with the rest of the teachers. This led to the establishment of a weekly meeting of teachers on a Friday to share lunch together and exchange stories of what was happening in the classroom. This informal exchange of crucial information sparked the idea of a more formalised model.

The first issue to address was the student voice. It was mostly missing or filtered in these meetings. Teachers, despite their best intentions, could either limit the negatives or simply fail to notice something amongst the students. Whilst the students in SDA were mature, they also felt a sense of competition for job interviews, so even if some felt confident enough to speak up, others might have preferred to stay silent.

The second issue to address was the wider environment. As the SDA project was operating as a prototype, there was a lot of freedom in what happened during the education, in terms of scheduling, rooms, equipment, etc. The block teaching model also meant that 30+ students had to share a cramped environment with their peers as well as all the computing equipment for five full working days over three months. Even if, by chance, the teaching space was perfect on day one, it is quite certain that the evolution of the group dynamics would certainly introduce unforeseeable challenges.

The final issue to address was academic progress. The cohort came from a variety of diverse backgrounds and their knowledge and prior experiences were quite diverse. Some students easily navigated through the first weeks, whereas others had to adjust from a different field (e.g., business, navy, and biology, to name but a few) and had no prior experience in programming. Therefore, there needed to be some way of continuously assessing the students that was not too demanding, but provided vital indication of students that might need more attention than others.

As there was already a consensus established on teachers holding a weekly meeting, this provided the perfect pivot to gather data beforehand, process them during the lunch period, and then act on it before the next week of education would begin.

### *4.2. Intervention: ECK Evaluation Model*

Figure 3 shows a high-level overview of the experience, confidence and knowledge evaluation (ECK) model developed for the SDA project [18]. In week 1, the planned teaching activities are delivered. At the end of the week, the 30 min before lunch time is dedicated to running a short evaluation. The results of the evaluation are immediately made available for teachers to discuss during the lunch time. This discussion then concludes with a plan to address the most pressing matters that are revealed. Finally, there is a meaningful effort to respond by demonstrating to the students that their voices have been heard, and the adapted teaching activities are made clear for the second week. In essence, the first week was similar to a sprint in a SCRUM agile process [38]. At the end of any sprint, there must be a review and reflection before the next sprint can begin. The ECK evaluation is our version of this review and reflection before we begin the next sprint of adapted teaching activities.

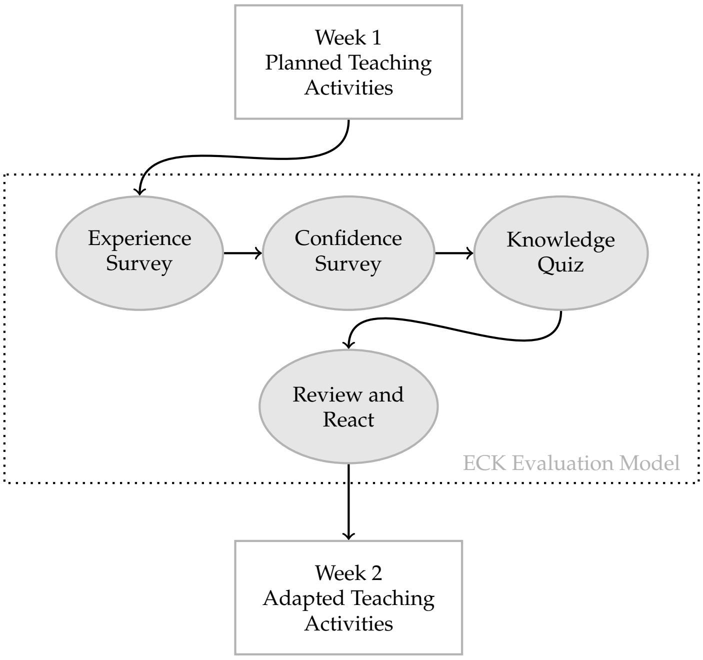


**Figure 3.** The ECK evaluation model is performed each week, capturing the present experience, confidence and knowledge of participants. Changes are then made for the following week to ensure that the pace and participant progress align.

The components of the evaluation address the issues mentioned in the previous section. The experience component is perhaps the easiest element of the students to complete, as it simply presents a series of Likert-style items on a five-point scale as shown in Table 1. These statements remained the same across the three month program and focused on such aspects as the level of comfort with the teaching environment, the level of stimulation the module gave them, whether they could get help with problems and if they could learn by collaborating with their peers. This component ended with an open reply box for anything that did not fit, for example, if the water cooler had run out or the heating was not comfortable in the teaching space. Finally, this was an anonymous survey in order to encourage students to express their concerns or worries.

The confidence component listed the topics that were covered during the past week. Once again, Likert-style items were used with a five-point scale that ranged from not very confident to very confident. As such, the list of topics changed each week. Table 2 shows the topics for the first four weeks. Note that iterators (a programmatic way to manage stepping through a collection of elements one at a time) are marked in bold. This is not an error, but rather this topic generated numerous problems, so it was also revisited during the next week as a result of the students marking their confidence as very low. In terms of time cost, this was a relatively easy task for teachers to complete and gave insightful information with low effort. The identity of students was taken for this instrument. This was considered a worthwhile risk in order to be able to compare the reported confidence with the results of the knowledge component. The justification was that when we discussed the results with the whole class, we could also map confidence and achievement correlations to help students understand the weak relationship.

**Table 1.** Experience survey set of Likert-style items with ratings on a 5-point scale from strongly disagree to strongly agree.

| ID | Statement |
|---|---|
| 1 | The pace of the course is manageable at this stage. |
| 2 | The level of course content is within my ability. |
| 3 | I feel comfortable in the teaching environment. |
| 4 | I would like more help with particular topics. |
| 5 | The course is challenging in a stimulating way. |
| 6 | I understand what the teachers are talking about. |
| 7 | I am able to learn from concrete examples that I can relate to. |
| 8 | The intended learning outcomes help me understand what I am to achieve. |
| 9 | The course activities help me to achieve the intended learning outcomes. |
| 10 | My background knowledge is sufficient to follow the course. |
| 11 | I am able to learn by collaborating and discussing with others. |
| 12 | I am able to get support if I need it. |

**Table 2.** Confidence survey topics for the first four weeks, highlighting that topics that students did not feel confidence in (iterators) were reviewed in the following week to ensure that student misconceptions were resolved.

| Week 1 | Week 2 | Week 3 | Week 4 |
|---:|---:|---:|---:|
| Class definition | Collections hierarchy | Basic Git | Exceptions |
| Constructors | ArrayList | Merge conflicts | Try/catch block |
| Methods | Generic types | **Iterators** | Checked vs. unchecked |
| Types of variables | Array types | Streams | IO Streams |
| Conditionals | For-loop statements | Coupling | IO Buffers |
| Boolean expressions | **Iterators** | Cohesion | Inheritance |
| Method header | Documentation | Code duplication | Extends keyword |
| Variable data types | Javadoc tool | Purpose of testing | Super keyword |
| Return values | | Testing limitations | Overriding methods |

The knowledge component took the form of a short multiple choice quiz. The questions assessed the skills that were taught throughout the week. Students were reassured that these quizzes were not used towards assessment, but instead, served as a way for teachers to understand where help was needed throughout the program. Despite this, there still was a palpable sense of stress each Friday as students waited for the quiz to be released.

In terms of implementation, Google Forms were used to create all three components. Links to each component were released by the teacher over the common Slack channel. A key feature of Google Forms was its ability to auto-summarise the gathered results and present the data via descriptive statistics that could immediately be interpreted. For the open text data, the results were small enough to skim during the lunch time meeting.

The final, and perhaps most vital, component was the review and plan for reaction. The standard approach was to start the week with a discussion of the results of the entire group. Presenting the correlations between confidence and knowledge was often insightful and well-humoured, as there were always examples of under-confidence with over-achievement and the reverse. For topics, such as iterators, time was made to revisit and try to work through the stumbling blocks to make sure that everyone was able to move on with confidence.

### *4.3. Findings and Discussion*

The SDA project ran for nine iterations before it concluded. Once the ECK model was adopted, it remained a core aspect of each iteration, running through each module. Given the pace and dynamism of the project, this aspect provided a much needed line of communication between all stakeholders. Students felt that they had ample opportunities to indicate their side of the experience. Whilst repetitive in some ways (experience), the evaluation was always changing in line with the module (confidence and knowledge). This helped to maintain interest and commitment, and perhaps most importantly, the time cost was only 30 min at the end of each week.

For teachers, on the positive side, there was a predictable stream of data to analyse and make changes based on what was shown. The experience component was also very useful for the administrative staff in the project, who received vital signals that they, too, could act upon, especially if there was any discomfort. Over time, the presentation and analysis of the results was the first item on the agenda and could often dominate the discussion at lunch. Unlike the traditional courses that teachers were teaching in isolation, here, there could be 10 or more individuals from different backgrounds helping them make sense of the student situation. This leads to the first important finding.

*Finding 1.1: Continuous course evaluation can become a cooperative, collaborative and collegiate conversation.*

Despite the obvious benefit, the time cost was still an issue for the knowledge component. Whilst teachers were happy to adopt the experience and confidence components, it took the team much longer to develop a good set of multiple-choice questions to help test knowledge. This was especially problematic when a new module was introduced; the effort was put towards content production while the continuous evaluation aspect was often neglected. This leads to the second finding from this case (which we go on to address in Section 6).

*Finding 1.2: Teacher adoption might only be partial when additional assessment material is required.*

The other downside was the potential for slippage with the time plan. If there were too many delays in going over old material, then the entire program would risk breaking its tight schedule. A counter-argument here is that unaddressed knowledge gaps will ultimately ruin the time plan, as students will be in no state to handle more advanced topics in the first place, and rushing ahead is simply teaching while ignoring the students' situation. Part of the attraction of intensive education is the shorter time frame. However, there is still the case that this agile approach to managing the course will better help identify the challenging parts that might need more time than a teacher originally planned.

*Finding 1.3: Adaption is fine up to a point, but time is still the limiting factor in accelerated education.*

Taken together, these findings illustrate how an intervention, such as the ECK evaluation model, can have a positive impact while simultaneously creating new challenges to overcome, such as teacher buy-in and time on topic management. The following case picks up these challenges by attempting to make teaching more efficient so that time can be better spent elsewhere.

## 5. Case Two: Reduced Learning Time with Maintained Outcomes

A key component of accelerated learning is making efficient and effective use of time, without sacrificing learning outcomes. The second case (see [8] for full details) describes how the first module, programming foundations, had its learning reduced by 25% with no harm caused to students and minimal increased time cost for teachers.

### 5.1. Case Context and Motivation

The modules for the SDA project were sourced from existing degree courses along with teachers willing to convert them to a compressed, block format. As such, lecture slides, assignments, assessments and so on were readily available. The programming foundations module was already compressed into block format for the purpose of international teaching exchange trips, so it became the most stable module; as such, it was also a good candidate for reducing the learning time further.

Two factors are important for achieving this goal. First, students have access to the learning material whenever they want and in a convenient format—typically web-based and available via multiple devices. Second, as students are working without anyone to help when they have a misconception, the feedback that the system returns must help assist the student in resolving their misconception whilst simultaneously not revealing the correct answer so that the learning activities remain meaningful. Ultimately, these factors, when combined, can effectively help accelerate student learning by allowing a higher degree of self-regulated learning.

As mentioned in Section 2, the Open Learning Initiative (OLI) at Carnegie Mellon University has reported impressive reductions in learning time when directly compared with traditional teaching. Thus, the decision was made to adapt the programming foundations module from traditional delivery to a question-based learning approach, supported by the OLI platform. Given the flexibility of the SDA project, teachers were free to suggest technologies that they would like to try, which contrasts with the multi-year learning platform commitments that universities typically use.

### *5.2. Intervention: Question-Based Learning with OLI*

Figure 4 shows an overview of how the delivery of the programming foundations course was both flipped and integrated into the OLI platform. In the traditional approach, students were welcome to come to the teaching space early and work on lab tasks and book reading. There were also two teaching assistants on hand to deal with any questions, which was true for all lab-based sessions. At 10:00, a two-hour lecture was given by the module responsible teacher. This followed a typical one-way lecture presenting the main topics and theory of the day, with some interactive elements, such as live coding and question and answer sessions. Students had a one-hour break for lunch before returning to continue working on the lab assignments for the next four hours.

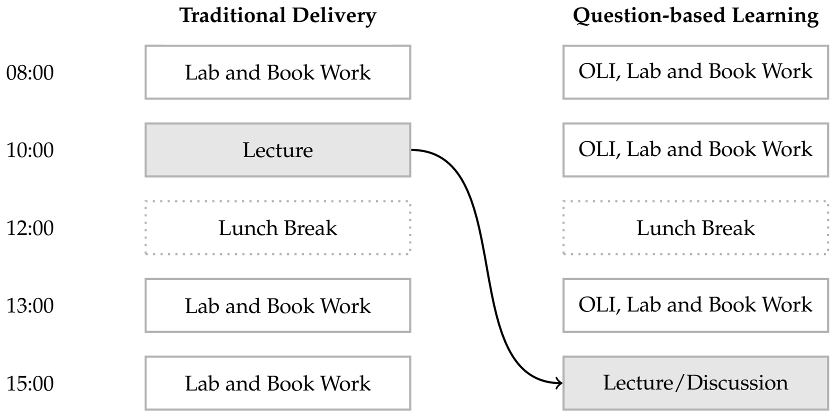
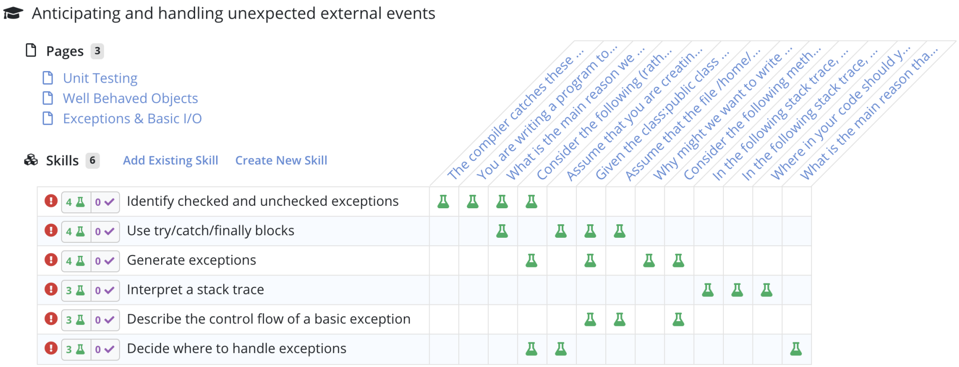


**Figure 4.** The transition from the traditional mode of teaching, where the teachers gave instruction towards the start of each day in a lecture format, to the question-based learning (QBL) approach, where students engaged with OLI material and then had a lecture/discussion about the questions that they struggled with the most.

Preparation for the transition to the question-based learning (QBL) using the OLI approach started well in advance of the fourth iteration of SDA. A group of teachers and teaching assistants spent time over the summer break writing the online text and learning activities for the module. This began with a mapping of the module into learning objectives. Then, the learning objectives were further subdivided into skills that students were expected to master before moving onto the next learning objective. The final piece was to then create learning activities that targeted these skills. The course building interface that the OLI platform uses embeds this model into the material, so it is clear to see how the learning objectives, skills and learning activities fit together. Figure 5 shows an excerpt of this model for one learning objective.



Anticipating and handling unexpected external events

Pages 3

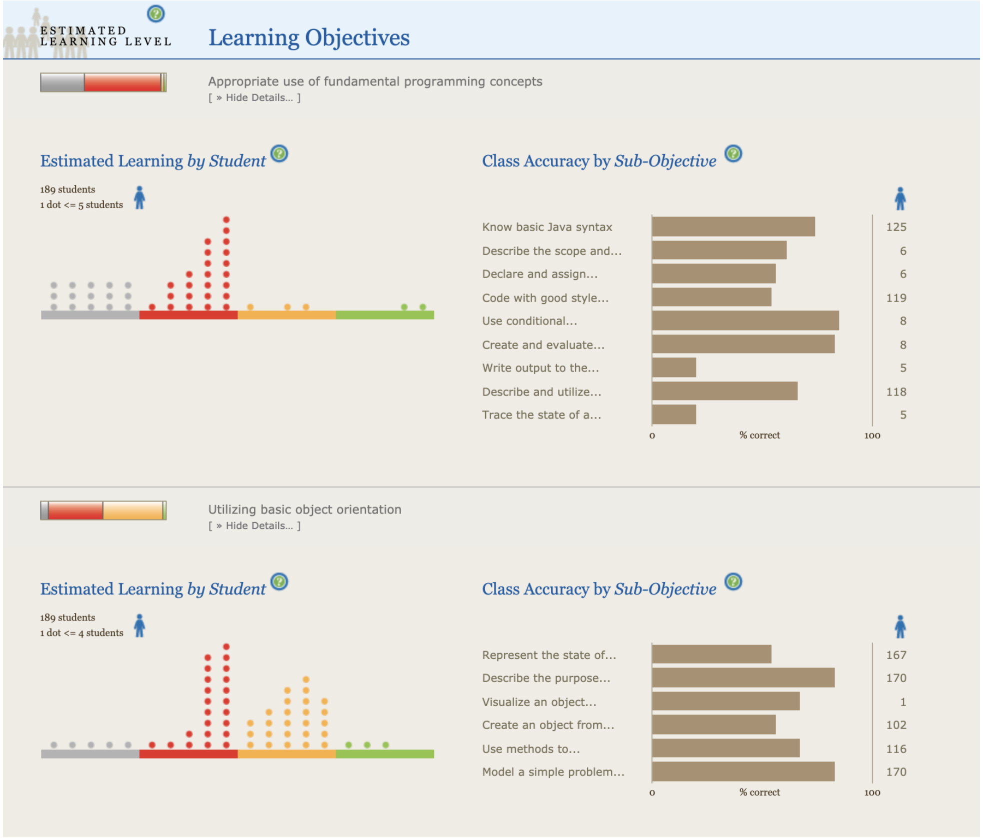
- Unit Testing
- Well Behaved Objects
- Exceptions & Basic I/O

Skills 6 Add Existing Skill Create New Skill

| | | The compiler catches these ... | You are writing a program to... | What is the main reason we ... | Consider the following (rath... | Assume that you are creatin... | Given the class:public class ... | Assume that the file /home/... | Why might we want to write ... | Consider the following meth... | In the following stack trace, ... | In the following stack trace, ... | Where in your code should y... | What is the main reason tha... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ! 4 ⚗ 0 ✓ | Identify checked and unchecked exceptions | ⚗ | ⚗ | ⚗ | ⚗ | | | | | | | | | |
| ! 4 ⚗ 0 ✓ | Use try/catch/finally blocks | | | ⚗ | | ⚗ | ⚗ | ⚗ | | | | | | |
| ! 4 ⚗ 0 ✓ | Generate exceptions | | | | ⚗ | | ⚗ | | ⚗ | ⚗ | | | | |
| ! 3 ⚗ 0 ✓ | Interpret a stack trace | | | | | | | | | | ⚗ | ⚗ | ⚗ | |
| ! 3 ⚗ 0 ✓ | Describe the control flow of a basic exception | | | | | | ⚗ | ⚗ | | ⚗ | | | | |
| ! 3 ⚗ 0 ✓ | Decide where to handle exceptions | | | | ⚗ | ⚗ | | | | | | | | ⚗ |

**Figure 5.** Screenshot of the objectives page from the programming foundations module. A specific learning object is broken into skills that are tested by different learning activities.

At the beginning of the module, students were briefed on the importance of engaging with the OLI material throughout the day before the lecture in the afternoon. For teachers, this provided a new source of information and insights to include in their preparation—for example, the learning dashboard for specific learning objects, skills, performance for students is shown in Figure 6. Rather than working through the slides in sequence, teachers would start a discussion about the results of the student attempts. This typically created a focus for the first period of the lecture. Once exhausted, other topics were reviewed that might not have presented as much of a challenge, but were interesting to discuss.



ESTIMATED LEARNING LEVEL

Learning Objectives

Appropriate use of fundamental programming concepts
[ » Hide Details... ]

Estimated Learning *by Student*

189 students
1 dot <= 5 students

Class Accuracy by *Sub-Objective*

| | |
|---|---|
| Know basic Java syntax | 125 |
| Describe the scope and... | 6 |
| Declare and assign... | 6 |
| Code with good style... | 119 |
| Use conditional... | 8 |
| Create and evaluate... | 8 |
| Write output to the... | 5 |
| Describe and utilize... | 118 |
| Trace the state of a... | 5 |

Utilizing basic object orientation
[ » Hide Details... ]

Estimated Learning *by Student*

189 students
1 dot <= 4 students

Class Accuracy by *Sub-Objective*

| | |
|---|---|
| Represent the state of... | 167 |
| Describe the purpose... | 170 |
| Visualize an object... | 1 |
| Create an object from... | 102 |
| Use methods to... | 116 |
| Model a simple problem... | 170 |

**Figure 6.** Screenshot of the OLI learning dashboard that teachers use to gain insights into student progress and achievement based on learning objects, skills and learning activities.

### *5.3. Findings and Discussion*

Prior to using a question-based learning approach, there was a sense of urgency as teachers attempted to keep students on track. A consequence of the ECK model, discussed in Section 4, was that difficult topics demanded more time than was scheduled, and the schedule of topics inevitably drifted, i.e., the original intended schedule would start to depart from what was happening in class in the middle of the module as the difficulty increased. Two changes occurred with the new approach: (1) topic drift was no longer a serious issue, and (2) the module went from taking four weeks to just three, allowing that saved time to be spent later in the program for more advanced topics.

Underpinning the intervention was the notion that we should "do no harm" with respect to student achievement. The results of the weekly knowledge quiz were available for four iterations: two before the intervention and two after. Whilst there was a slightly lower score for the QBL approach, it was found to not be statistically significant. The same lack of significant difference was found in the confidence scores for the topics in the module. The sources of data collected are summarised in Table 3. This leads to the first finding of this particular case.

*Finding 2.1: Learning time can be reduced without harming learning outcomes*

The teachers involved highly enjoyed the flipped classroom driven by the QBL approach. It did not take long for students to adjust to the mode of delivery. The lecture became a discussion driven by questions from the students regarding difficulties they had in the material. This led to a very positive classroom energy, with the result that scheduled mid-lecture breaks sometimes were ignored, as the students wanted to push on with the discussion. Typically, lectures took a 10 min pause after the first hour to allow students to have a break. Despite being given the option to take a break, sometimes, the students were so engaged that they preferred to keep pushing ahead with the discussion about their questions on the topic. That said, there were some data from the experience surveys that suggested that a few students still would have preferred the theory up front and presented by a teacher before attempting the QBL material and lab work themselves.

**Table 3.** Summary of data collected on the reduction in learning times with maintained outcomes. SDA 4 and 5 used a traditional approach to teaching, compared to SDA 6 and 7 that used the question-based learning approach (QBL).

| | **SDA 4** | **SDA 5** | **SDA 6** | **SDA 7** |
|---|---|---|---|---|
| Delivery Method | Trad. | Trad. | QBL | QBL |
| Course Length (days) | 23 | 23 | 17 | 17 |
| Student Time Logs | No | Yes, last 10 days | Yes | No (Covid) |
| Student Experience Survey | Yes | Yes | Yes | Yes |
| Student Knowledge Quiz | Yes, 57 qns | Yes, 61 qns | Yes, 61 qns | Yes, 61 qns |
| Student Confidence Survey | Yes | Yes | Yes | Yes |
| Teachers Time Report | Yes | Yes | Yes | Yes |

There was some adjustment required in terms of preparation. This divided the teachers, with one finding that it dramatically reduced their preparation time, whilst the other found no reduction in their own prep time. Possible explanations here are the first teacher being more actively involved in the preparation of the OLI course material and having more experience with delivering the module, with the second teacher preferring comprehensive preparation in case unexpected topics are raised by the students.

*Finding 2.2: Teacher prep time varies, but class energy and engagement are consistently better.*

Despite the positive experiences for teachers and students alike, the initial cost of preparation is unavoidable. We solved this by hiring experienced teaching assistants who had worked in the SDA project for several iterations to work over the summer on this mission. Whilst this solved the basic content of the course, the shortfall was the production of high quality multiple choice questions that contained feedback to correct misconceptions without the need for a teacher to be present. Despite using a novel approach of dividing the labour between teaching assistants—one would create the question text, the other would create alternatives, and the third would write the feedback ensuring all questions had multiple inputs and review opportunities—we still struggled to generate enough questions for the first iteration. The consequence here is the loss of learning opportunities to really identify whether an individual student has mastered a skill, and the ability to view learning curves for the entire cohort so as to be able to better tune the course material and enhance the delivery.

> *Finding 2.3: Developing good multiple questions with constructive and specific feedback is the most time-consuming aspect of course material development.*

Despite this challenge, the OLI system does provide very useful information on the learning curves of students, which lets us accurately identify where to target efforts between course iterations. These insights helps to highlight the types of questions that can be culled safely without affecting the mastery of skills, whilst shining a light on the areas in which students need more practice opportunities.

## 6. Case Three: Learnersourcing Multiple Choice Questions

Both previous cases experienced a common problem: the production of quality learning activities is costly in terms of time and effort. It is no use adopting interventions for accelerated learning unless they are sustainable, as eventually, the economy of production will exhaust the resources available. The cost of teacher time can be somewhat offset by recruiting teaching assistants, as was done in Case 2; however this simply shifts the bottleneck along the line by one step, rather than eliminate it.

An alternative approach is learnersourcing [25]. Simply put, students create content for other students to consume. This creates an almost scale-free means to produce academic content, as there is always a linear relationship between the number of students and the number of learning activities that can be produced. The effort of the teacher switches from production to management. Perhaps the most pertinent question is whether the content produced by students can ever match that of an experienced teacher. Research conducted so far in this space suggests that it is not only possible, but when teachers are asked to identify learning activities generated by students or teachers, there can be a genuine struggle to correctly classify what students have produced. Given the scale-free nature, it is possible to add layers of review and filter out the lower quality work and promote the best content; once again, this task can be effectively delegated to the students.

Technology plays a critical role; otherwise, the management task of gathering the multitude of learning activities produced by students would be prohibitive. Tools such as PeerWise [26] and Ripple [39] are good examples of specific tools that are free and simple for teachers to use for this process. They are web-based, integrate with existing learning management systems, and often incorporate motivational features, such as gamification [40], to further heighten student engagement. In the third case (see [29] for full details), we experimented with using PeerWise to have students produce multiple choice questions at scale.

### *6.1. Case Context and Motivation*

The programming foundations module was assessed using a two hour exam that consisted of a section of short multiple-choice questions and then a series of longer questions. Students also had a mock exam in the same format. Due to the SDA project running two iterations per year, this led to a burden of producing questions. Even recycling exam scripts between iterations led to problems, as students could easily link up with former students and receive ‘tips’. Besides the issues with assessing programming knowledge on a written exam, this burden motivated another solution to be found. Furthermore, in other parts of the course development, the weekly knowledge quizzes and OLI material also created demand for more learning activities to be produced.

### *6.2. Intervention: Peer Production of MCQs with PeerWise*

The solution was to base the students' assessment on their ability to produce good multiple-choice questions on the material that they had just learnt. A simple regime was established. At the end of the week, a two-hour session was devoted to writing multiple-choice questions. Students were trained on how to use the PeerWise system to both produce and take MCQs. The basic requirement to pass was for students to produce a total of six questions; two for each week of the module. Students were free to answer as many questions as they liked. Teachers reviewed the submissions each week and left feedback on what improvements were needed. The old exam was abandoned.

The intervention was very positive in terms of achieving its goal. The cohort consisted of about 35 students, on average, across iterations. Over three weeks, students produced six questions, leading to an expected total of 210 questions in just a single iteration. For a single teacher to produce this volume of learning activities in a three-week span is unthinkable (at least if they have their normal activities to contend with). For the students, there was no high-stakes final exam to worry about. Instead, the evidence of their learning was captured in their ability to think deeper and produce learning activities for their fellow students to attempt, creating a virtuous and efficient circle.

Over several iterations of this process, small improvements were made. First, the Friday session became a group activity, where students created questions together and discussed questions created by others. Second, a list of guiding principles, shown in Table 4, to help students better understand what made good questions, answer alternatives and provide effective feedback, was provided to help avoid cold start problems with a new way of working with assessment. Third, a more formalised review process was created from the principles, and students were expected to formally review questions as part of their assessment. Finally, although the COVID-19 pandemic forced SDA to transition to fully remote teaching, the process still held up and worked well as an education activity at distance.

**Table 4.** Summary of the principles of good multiple choice questions derived from the initial and exhaustive analysis.

| | |
|---|---|
| **Principles of Good Questions** | (1) Question is from the course domain.<br>(2) Question is targeted towards a misconception.<br>(3) Question is not based on reference look up.<br>(4) Question is reasonable to solve without external systems. |
| **Principles of Good Answering Alternatives** | (5) Three or more answer alternatives are provided.<br>(6) Answer alternatives are plausible and linked to the misconception.<br>(7) Answer alternatives are formulated to maximize readability. |
| **Principles of Good Feedback** | (8) Feedback is constructive.<br>(9) Feedback is unique and provided for each answer alternative.<br>(10) Feedback for answer alternatives does not reveal the answer. |

### *6.3. Findings and Discussion*

The most immediate finding that was surprising was the extent of the engagement. Students were given no indication of how many questions they should answer. A nice feature of the PeerWise interface is that the landing page divides into the questions you have authored, the questions you have answered, and the questions you have not yet answered. This encouraged students to keep answering questions as new ones were authored and posted to the tool. By way of illustration, Figure 7 shows the participation statistics on PeerWise; 50% of the students have answered 100 or more questions. Even the least active student managed to pass by virtue of authoring their six questions. Furthermore, the other bars show that students also actively rated questions after answering them, with a few going as far as posting comments with feedback on improvement suggestions or corrections.

*Finding 3.1 Peer production intensifies engagement with learning activities.*

There was no special reason for requiring students to produce two questions each week. From our own experience authoring questions, it can be quite tricky to get right, so we did not want to burden the students with meeting a higher target. That said, no students complained of the effort; class time was set aside to create a focus and a space for discussion, and only very few students needed to be reminded to complete the task. Whilst the quality of questions cannot be guaranteed, at least the task of producing learning activities becomes scale-free and the challenge simply shifts to filtering by reviewing, which, as mentioned, works just as well when delegated to students. Perhaps the only downside is that some training is required to help students better understand the root of a good question and how to create it themselves.

*Finding 3.2: Students can be recruited to produce learning activities at scale.*

Most notable in lack of quality was the feedback. We had quite high expectations, as the ultimate goal of building a pool of questions with feedback would be to integrate them into the OLI material, where we had a need for learning activities. However, OLI questions work best when there is accurate and specific feedback for each answer alternative that a question has, that also manages not to accidentally reveal the answer. Table 5 shows the results before we asked students to follow the principles for good MCQs. Students scored highly for both writing the question (76%+ good) and the answering alternatives (59%+ good), but struggled with good feedback as rated by teachers (34%+ good) over a three-week period.

*Finding 3.3: Students struggle to produce high-quality feedback without guidance.*

Given the challenge of producing good questions, the creation and discussion of the principles had a positive effect in improving the quality of the questions overall. Table 6 shows the effects of the principles over two iterations of SDA, where students produced questions. First, we re-reviewed our own judgments of the data in the sixth iteration, using the principles shown in Table 4 and came to a more consistent view of the quality. Then, in the seventh iteration, students followed the principles, leading to an improvement across the board in judgements. This approach of using the principles was then converted into a form that students themselves used to judge questions and ongoing work. Teachers and students mostly agree on aspects of quality in questions; however, there is still work to be done on handling subjective aspects of questions, such as whether the feedback provided would help another student correct their misconception.

*Finding 3.4: Guiding principles can avoid cold start problems and increase overall quality.*

In particular, this case was a big shift in thinking about who is responsible for creating learning activities within the SDA project. Based off the success of this intervention, the foundations were laid for a much more radical shift in responsibilities, as the next case describes.

**Table 5.** Percentage of questions, alternatives and feedback per week for all student generated MCQs in SDA-6 (n = 230).

| | Question | | | Answering Alternative | | | Feedback | | |
|---|---|---|---|---|---|---|---|---|---|
| | Good | Ok | Poor | Good | Ok | Poor | Good | Ok | Poor |
| **Week 1** | 84.2% | 13.2% | 2.6% | 61.8% | 34.2% | 3.9% | 34.2% | 57.9% | 7.9% |
| **Week 2** | 78.5% | 12.3% | 9.2% | 78.5% | 20.0% | 1.5% | 43.1% | 38.5% | 18.5% |
| **Week 3** | 76.7% | 19.9% | 3.4% | 59.2% | 38.0% | 5.6% | 43.7% | 47.9% | 11.3% |

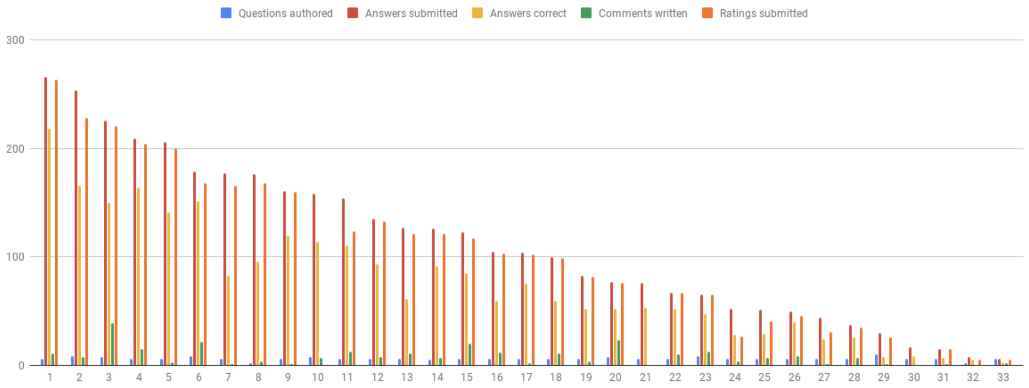


**Figure 7.** Participation data for PeerWise. Students demonstrated high engagement for answering questions (red bars) and all managed the baseline expectation of authoring six questions (blue bars).

**Table 6.** Comparison of quality of constructive feedback that students provided with their questions between SDA-6 (both the initial analysis without the principles and a retrospective analysis with the principles) and SDA-7 (with the principles).

| Iteration/Feedback | Good | Ok | Poor |
|---|---|---|---|
| SDA-6 Pre-principles review | 42.9% | 45.8% | 11.3% |
| SDA-6 Post-principles review | 32.4% | 36.6% | 31.0% |
| SDA-7 Post-principles review | 49.3% | 33.8% | 16.9% |

## 7. Case Four: Peer Teaching for Emerging Topics

In the final case, we see perhaps the largest disruption, with students taking control of the teaching of other students (see [31] for full details). Whilst involving students in the production of learning activities, as presented in Case 3, can be seen by many as an efficient and economical way of producing more content at scale, trusting students to actually carry out the teaching is a leap that many may not feel immediately comfortable with. However, learning by teaching [10], peer tutoring [12] and peer teaching [11] have a long history in education, and their benefits have been reported [9]. Vygotsky concluded that students learn as much from their peers as they do from their teachers [13]. More recently, one of the simplest and most effective forms was proven to be peer instruction, where students are posed questions in a lecture, they submit initial answers independently, they see the results, they engage in discussion with their immediate neighbour, and then they have a chance to submit a final, possibly revised, answer [41]; numerous studies show the benefits of trusting students to help each other correct misconceptions whilst learning.

Whilst this might seem like a dream for teachers to reduce their workload, effective peer teaching demands careful preparation in advance and then attentive facilitation throughout the process to make sure that students stay on track and feel confident assuming their new role. Whilst the programming foundations module has been a crucible for experimentation, this final case addresses a challenging situation for the more advanced modules in the SDA project.

### *7.1. Case Context and Motivation*

The main goal of the SDA project was to find meaningful IT work for newly arrived immigrants in Sweden. As such, the later modules were tuned towards industry needs, such as mobile and web apps. This meant that students would go from introductory programming to developing applications in a complicated framework that itself would take months of learning to master. This created a phase shift in complexity of material that the students felt through each iteration. Something was missing to help them step up to this new level.

There are many software frameworks and libraries that are possible candidates to learn before taking on full-scale mobile app development. By virtue of being smaller, they are less complex and typically focused on something specific, such as configuration management for a project, or providing virtualisation to simplfy project deployment. Furthermore, these are transferable skills that are beneficial to learn regardless. However, this is also a fast moving space, and finding teachers who were (1) up to date, and (2) willing to teach was a perennial challenge.

A solution lay in how software teams typically keep their skills relevant—someone on the team learns a new skill, and then they teach their teammates how to acquire the skill, often showing the most efficient route that worked for them, bypassing endless decisions of which tutorial or documentation to focus on. The biggest leap of faith for teachers designing the new enterprise technologies (ET) module that would run in advance of the mobile app development module was that they themselves had little practical experience with the technologies being selected. They had the awareness that this was something important in the IT industry, but none were anywhere close to being experts and were, at best, conversant in the technology. In spite of this, it was assumed that students would follow the same pathways that a teacher might in bringing themselves up to speed with a particular technology; therefore, why not let that process happen inside a structure that ensures that other students would benefit as well?

### 7.2. Intervention: Peer Teaching of Enterprise Technologies

The ET module was designed to run over a one week time-span and cover six important technologies that would (1) help the students prepare for learning more complex technologies, and (2) become a transferable skill. Figure 8 shows the order of events for the week. Starting with the preparation phase, students received a one-hour lecture that explained the concept, assigned teams and allocated technology to each team. Each learner team (N = 6) had to sit and work together on sourcing existing guides and then create an overview presentation of what the technology was, why it was important and what other similar technologies could also be used. Then, they had to design a workshop that the rest of the class would participate in and the learner team would assume the role of teaching assistants to help them throughout. The learner teams only had one and half working days to prep. As the preparation phase concluded, students were asked to reflect on how they felt working in a learner team.

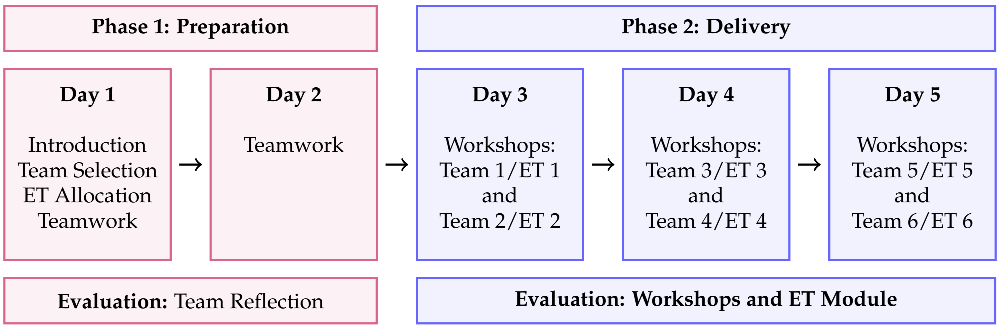


**Figure 8.** The breakdown of days, phases and activities for the enterprise technology module (ET) within SDA.

The delivery phase was structured as six two-hour workshops over three days, concluding with a reflective session. During this phase, a teacher attended the start of each workshop to listen to the ten-minute presentation and to ensure there was good order, and everything that was needed was in place for the workshop to continue. After that, the teacher left the now teaching teams and students. After the final workshop, the teacher returned to have a casual conversation about how the experience went. In total, the teacher spent three hours of their time on the module. In later versions, this was reduced further, as teaching assistants could fulfil all the roles that the teacher played, resulting in zero-cost for the teacher in all future iterations.

Despite the module only running for one week, six technologies were successfully taught. Students had multiple opportunities to reflect and provide feedback on the process used. They reflected on their learner team experience, the experience of each workshop (the teacher teams ran their own evaluations using a supplied template), and then a final reflection on the overall experience of the week. Table 7 shows the types of Likert-style statements that were used to capture student evaluation at the various stages of the module.

**Table 7.** The evaluation was divided into three surveys, delivered during the preparation, delivery and module reflection phases, respectively. Each consisted of a series of statements for which the students could indicate their level of agreement. Responses were recorded using a five-point Likert scale from strongly disagree to strongly agree.

| | |
|---|---|
| **Preparation Phase** | |
| *Prior Knowledge* | I have used this enterprise technology before this module |
| | I would like to learn more about this particular enterprise technology |
| | I feel that I can describe this enterprise technology to another class mate |
| *Motivation* | I am confident that my class will benefit from my workshop |
| | I feel stressed by the prospect of teaching my class mates |
| *Collaboration* | I liked working as a teaching team when learning |
| | Working in a teaching team was more engaging than learning by myself |
| | I would prefer more time alone to focus on my own learning |
| | Working in a teaching team felt more slow than if I had worked independently |
| | I found that I could help my team develop their understanding |
| | I found that my team helped me develop my understanding |
| | The collaborative aspect helped to overcome individual challenges |
| **Delivery Phase** | |
| *Leading a Peer Workshop* | I felt we delivered the learning objectives that were planned during preparation |
| | I was happy working in a teaching team during my workshop |
| | Taking the role of teacher increased my confidence in discussing the ET |
| | Resolving student challenges made me feel more like an expert in the ET |
| | The workshop evaluation provided constructive feedback that I could act on |
| | I am motivated to make improvements to my workshop after delivering and evaluating it |
| *Experiencing a Peer Workshop* | I liked having my fellow SDA students as teachers |
| | I felt comfortable asking my teaching team for help if I needed it |
| | I feel more confident learning about other ETs after the workshops |
| | The consistency of the workshops might have been better with tighter requirements |
| | I got ideas about how to improve my workshop from experiencing others |
| | After each workshop, I felt motivated to learn more about the ETs that were presented |
| **Reflection upon ET Module** | |
| | I feel this was an engaging way to learn about several ETs |
| | The intense format was an efficient use of time |
| | The presentation—workshop—evaluation format was a good fit |
| | The level of detail of information was generally about right |
| | The time allocated was too short to understand each ET sufficiently |
| | It might be better to focus on less ETs |
| | I would have preferred an expert rather than my peers as teachers |
| | If someone outside of SDA mentioned that they one of the ETs, you feel prepared to engage them |
| | I'm surprised by how much I could learn about different ETs in a short time |

### *7.3. Findings and Discussion*

This was a high-risk, high-reward experience. Very little time at all was required to prepare the module, although careful attention was given to its structure and making sure that there were multiple opportunities to evaluate what was happening at any given stage. Very little effort was required to source the technologies—simply knowing it was important to software engineers, and checking for the existence of good documentation and selection of beginner tutorials was sufficient.

*Finding 4.1: Peer teaching removes virtually all teacher prep time.*

This almost carefree abandon was quite liberating for the teachers—we could suggest hot and emerging technologies without the normal moderating forces: (1) it might not stay hot for long and (2) therefore, I do not want to invest in preparing learning materials if it is not to be sustained. In fast-moving fields of study, this becomes a real power to yield, as the cost essentially becomes zero. Thus, the range of technologies explodes beyond the expertise of the teachers, which is especially important, as teachers may change over time, become unavailable, and so on, creating an unsustainable dependency.

*Finding 4.2: Peer teaching can touch emerging knowledge beyond teacher expertise.*

The student evaluation for this approach was overwhelmingly positive, even on the first iteration when no one really knew what was going to happen. Despite the success in covering six useful technologies in just a week, there was a deeper force at work that emerged through the reflective discussions that capped off the week. Students reported a new sense of confidence in their own learning. They were not as afraid of tackling the next challenge, given this experience. They also understood how they had their own ability to teach something to others that they had barely just learnt themselves (most experienced teachers will recognise that the best way to learn something is to teach it to others).

*Finding 4.3: Students realise their innate ability to teach and become more empowered learners/teachers.*

Despite the intensive format of the SDA project, it was quite fascinating to witness how quickly students could transition from beginners to those who felt confident to teach their peers. For teachers, simply creating a well-structured learning environment, then stepping aside, showed how quickly students could make this change.

## 8. Discussion

In Table 8, we have summarised the case studies with regard to the learning quality and the resource conservation. These aspects can be further used to select or reject approaches to learning that may also be useful in sustainable education. Their relation to other research can be found in the subsections below.

### *8.1. Agile Management*

Agile management, as we describe it, includes previously reported good pedagogical practices, such as assessment with immediate feedback [21–23], student evaluation of learning [19] and reflection on learning [20] but also technical support (in this case, Google Forms) to automate the process for collecting and analysing information instantaneously. This mimics low-tech solutions that are often adopted in agile practices [42].

### *8.2. Reduced Learning Time*

Using a methodology that has already been shown to be six times more efficient than reading and watching videos [6] and, over a course, reduced the learning time by 50% [7], it is no surprise that we managed to reduce the learning time by 25%. However, the news here is that we managed to do this already with newly developed learning material. One of the great benefits with the OLI model is using click data from the students to improve the learning material over time. This is something we did not have time for, as the material was fresh for SDA 6, and SDA 7 started too soon after SDA 6 to allow for any changes.

### *8.3. Learnersourcing*

The problem of poor quality in student-generated feedback to student-generated questions is not new [43]. A natural remedy for this is to provide guidelines [44,45], but those 31 rules are probably intended for academics and not suitable for handing out to students, as the time it would take to check all 31 rules would probably surpass the time to create the question. That is also the reason why we started without guiding the students in order to identify the rules they really need (rather than a mixture of needed rules and obvious ones, such as "Use correct grammar, punctuation, capitalisation, and spelling" in [45]. These are also previous attempts to provide scaffolding to students through training that includes MCQ terminology, self-diagnosis quiz, template, and exemplar MCQs [46]. Our guiding principles did improve the quality of the feedback at a much lower cost for the students.

**Table 8.** Summary of the case studies by the key aspects of increased quality of learning and resource conservation.

| Case | Increased Quality of Learning | Resource Conservation |
|---|---|---|
| Case 1:<br>Agile management | • Rapid adaptation of learning material and to students' prior knowledge.<br>• Nip problems in the bud early before they blow up. | • Automation of data collection and summarisation.<br>• Analysis and response is shared with multiple stakeholders instead of a single teacher. |
| Case 2:<br>Reduced learning time | • Teacher time focused on the topics students find challenging and need support with. | • Automatisation of learning and assessment for content learned through the question-based learning methodology. |
| Case 3:<br>Learnersourcing | • Students take a teacher's perspective and have a deeper learning experience. | • Student-generated questions leads to a rapid development of new learning material. |
| Case 4:<br>Peer teaching | • Students take a teacher's perspective and experience that they can both learn together and teach others. | • Students develop and deliver learning material free from teacher intervention. |

### *8.4. Peer Teaching*

It was previously noted that learning how to use software frameworks efficiently may take 6 to 12 months [47] and that developing a framework-based application is closer to maintaining an existing application, rather than creating an application from scratch [48]. These observations taken together with the numerous already available different frameworks for similar needs [49] and the high tempo in which new frameworks are published makes teaching any of these an insurmountable task for a university teacher: the time is simply not available to learn all the frameworks, select the best ones, and revise everything yearly. With these preconditions, it seems that any change would be an improvement. However, while the driving force was identified in Finding 4.1—peer teaching removes virtually all teacher prep time—the two other outcomes (students learning beyond teacher expertise and empowerment of the students) benefited the students in ways that we did not anticipate before our study.

### *8.5. Sustainable Teaching and Learning*

In our search for a definition of sustainable education, the closest we came is a definition by Sterling: "change of educational culture that develops and embodies the theory and practice of sustainability" [50]. However, as we interpret Sterling's definition, this applies to how *sustainability* should be included in all subjects. Our definition, *increasing* *quality of learning while conserving resources*, aims at making education per se sustainable, regardless of whether the education includes sustainability or not.

In the Introduction, we defined the key aspects of *sustainable education* as *increasing quality of learning while conserving resources*, and we described four case studies that highlight how this can be done. We believe these two aspects, *increasing quality* and *conserving resources*, are fundamental building blocks in the definition of sustainable education. Our central argument is that we believe both these attributes must come hand in hand if we should have a chance of transforming education into a more sustainable format. Teachers can tend to be conservative in their practice, and we therefore need strong arguments for them to spend their time on changing entrenched habits. After all, it has only taken a global pandemic and the closure of university campuses to coax reluctant teachers into experimenting with remote and online learning technologies, by way of one prescient example in early 2020.

Increased learning quality is something teachers might agree with, but not if the increased quality comes at the cost of their already scarce spare time. This is the reason why conserving resources as well must be the second fundamental pillar. If teachers can be approached with *both* increased learning quality and reduced workload (or at least not increased work), the change will be much easier to motivate and accept. The same argument holds for the management and administrative levels of staff. All can agree that increased learning quality is good, but if significant resources have to be spent to achieve this, the discussion will be moved from a quality perspective to an economic one.

Arthur C. Clarke is often attributed to the statement "*Any teacher who can be replaced by a machine, should be*". The statement is bold, but the wider context around it is that we always need good teachers, up until the point that they can be replaced by technology. This implies that the good teacher cares to keep pace with change and apply it where there is a good case to be made for it. We propose it should be softened to "*Any teaching that can be replaced by a machine, should be*", as this will leave room for, and include teachers, to continue teaching the parts that cannot be taught by machines, and those that are better taught by teachers. Additionally, as the later cases in our study have clearly shown, the burden of teaching can be shared wider than we normally do by involving students in the equation in a balanced way.

One unavoidable limitation of this study is the dependence on IT and internet connection for these ideas to be implemented. Linking back to the SDG4 goal of Quality Education for All, we must remain painfully aware that we do not live in an equal world, and especially not in the dimension of education. However, while we do not have a solution for the general inequalities of the world, the ideas we propose do have value for those who have access to the internet, and offer ways to reduce the challenges others will face once they have access to the internet.

## 9. Conclusions

There is a risk that developing sustainable education becomes hidden behind education for sustainable development. We believe that this is an area that we can better understand by identifying the key foundational aspects of increasing quality whilst conserving resources, that is, we can do better without unnecessary compromise. The cases and findings we have presented here highlight what can be achieved simply by adopting a sustainable mindset and improving the education that students receive whilst making it better for teachers as well. There are many more good examples of sustainable education out there waiting to be identified, explained and brought into wider awareness for both teachers as well as their many students, which we hope will contribute towards a better quality of education for all.

**Author Contributions:** Conceptualization, R.G. and O.B.; methodology, R.G. and O.B.; writing—original draft preparation, R.G. and O.B.; writing—review and editing, R.G. and O.B.; visualization, R.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This SDA Project was funded by the Marianne and Marcus Wallenbergs Stiftelse and the EU European Social Fund (ESF). The APC was funded by KTH Royal Institute of Technology.

**Institutional Review Board Statement:** The study was conducted according to the guidelines of the Declaration of Helsinki and the Code of Ethics of KTH Royal Institute of Technology (V-2016-0347).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The data presented in this study can be made available on request from the corresponding author in accordance with the interests of KTH Royal Institute of Technology.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. The Sustainable Development Goals Report 2020. United Nations, 2020. Available online: https://unstats.un.org/sdgs/report/2020/ (accessed on 1 October 2021).
2. Daniela, L.; Visvizi, A.; Gutiérrez-Braojos, C.; Lytras, M.D. Sustainable higher education and technology-enhanced learning (TEL). *Sustainability* **2018**, *10*, 3883. [CrossRef]
3. Orozco-Messana, J.; Martínez-Rubio, J.M.; Gonzálvez-Pons, A.M. Sustainable higher education development through technology enhanced learning. *Sustainability* **2020**, *12*, 3600. [CrossRef]
4. Denneen, J.; Dretler, T. The financially sustainable university. *Bain Co.* **2012**, *12*. Available online: https://www.bain.com/insights/financially-sustainable-university/ (accessed on 1 October 2021).
5. Bier, N.; Lip, S.; Strader, R.; Thille, C.; Zimmaro, D. *An Approach to Knowledge Component/Skill Modeling in Online Courses*; Open Learning Initiative: Pittsburgh, PA, USA, 2014; pp. 1–14.
6. Koedinger, K.R.; McLaughlin, E.A.; Jia, J.Z.; Bier, N.L. Is the doer effect a causal relationship? How can we tell and why it's important. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, Edinburgh, UK, 25–29 April 2016; pp. 388–397.
7. Lovett, M.; Meyer, O.; Thille, C. The Open Learning Initiative: Measuring the Effectiveness of the OLI Statistics Course in Accelerating Student Learning. *J. Interact. Media Educ.* **2008**, *2008*, 13. [CrossRef]
8. Bälter, O.; Glassey, R.; Wiggberg, M. Reduced Learning Time with Maintained Learning Outcomes. In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, Virtual Event, USA, 13–20 March 2021; pp. 660–665.
9. Goldschmid, B.; Goldschmid, M.L. Peer teaching in higher education: A review. *High. Educ.* **1976**, *5*, 9–33. [CrossRef]
10. Duran, D. Learning-by-teaching. Evidence and implications as a pedagogical mechanism. *Innov. Educ. Teach. Int.* **2017**, *54*, 476–484. [CrossRef]
11. Wright, S.; Cowen, E.L. The effects of peer-teaching on student perceptions of class environment, adjustment, and academic performance. *Am. J. Community Psychol.* **1985**, *13*, 417–431. [CrossRef]
12. Goodlad, S.; Hirst, B. *Peer Tutoring. A Guide to Learning by Teaching*; ERIC: Washington, DC, USA, 1989.
13. Vygotsky, L.S. *Mind in Society: The Development of Higher Psychological Processes*; Harvard University Press: Cambridge, MA, USA, 1980.
14. Chase, J.D.; Okie, E.G. Combining cooperative learning and peer instruction in introductory computer science. In Proceedings of the Thirty-First SIGCSE Technical Symposium on Computer Science Education, Austin, TX, USA, 7–12 March 2000; pp. 372–376.
15. Wlodkowski, R.J. Accelerated learning in colleges and universities. *New Dir. Adult Contin. Educ.* **2003**, *2003*, 5–16. [CrossRef]
16. Kasworm, C.E. From the Adult Student's Perspective: Accelerated Degree Programs. *New Dir. Adult Contin. Educ.* **2003**, *97*, 17–27. [CrossRef]
17. Gratton, L. New frontiers in re-skilling and upskilling. *MIT Sloan Manag. Rev.* **2019**, *61*. Available online: https://sloanreview.mit.edu/article/new-frontiers-in-re-skilling-and-upskilling/ (accessed on 1 October 2021).
18. Glassey, R.; Wiggberg, M.; Haller, P. Agile and Adaptive Learning via the ECK-model in the Software Development Academy. In Proceedings of the EC-TEL (Practitioner Proceedings), 13th European Conference On Technology Enhanced Learning, Leeds, UK, 3–6 September 2018.
19. Wachtel, H.K. Student evaluation of college teaching effectiveness: A brief review. *Assess. Eval. High. Educ.* **1998**, *23*, 191–212. [CrossRef]
20. Keogh, R.; Walker, D.; Boud, D. *Reflection: Turning Experience into Learning*; RoutledgeFalmer: Abingdon, UK, 1985.
21. Sadler, D.R. Formative assessment and the design of instructional systems. *Instr. Sci.* **1989**, *18*, 119–144. [CrossRef]
22. Kulkarni, C.; Klemmer, S.R. *Learning Design Wisdom by Augmenting Physical Studio Critique with Online Self-Assessment*; Technical Report; Stanford University: Stanford, CA, USA, 2012. Available online: https://hci.stanford.edu/publications/2012/SelfAssessmentInDesignSpring2012TR.pdf (accessed on 1 October 2021).
23. Bälter, O.; Enström, E.; Klingenberg, B. The effect of short formative diagnostic web quizzes with minimal feedback. *Comput. Educ.* **2013**, *60*, 234–242. [CrossRef]
24. López-Alcarria, A.; Olivares-Vicente, A.; Poza-Vilches, F. A systematic review of the use of agile methodologies in education to foster sustainability competencies. *Sustainability* **2019**, *11*, 2915. [CrossRef]

25. Kim, J. Learnersourcing: Improving Learning with Collective Learner Activity. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2015.
26. Denny, P.; Luxton-Reilly, A.; Hamer, J. The PeerWise system of student contributed assessment questions. In Proceedings of the Tenth Conference on Australasian Computing Education, Wollongong, NSW, Australia, 1 January 2008; Volume 78, pp. 69–74.
27. Wheeler, S.; Yeomans, P.; Wheeler, D. The good, the bad and the wiki: Evaluating student-generated content for collaborative learning. *Br. J. Educ. Technol.* **2008**, *39*, 987–995. [CrossRef]
28. Denny, P.; Luxton-Reilly, A.; Hamer, J. Student use of the PeerWise system. In Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education, Madrid, Spain, 30 June–2 July 2008; pp. 73–77.
29. Glassey, R.; Bälter, O. Put the Students to Work: Generating Questions with Constructive Feedback. In Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), Uppsala. Sweden, 21–24 October 2020; pp. 1–8.
30. Barlai, M.; Fähnrich, B.; Griessler, C.; Rhomberg, M. *The Migrant Crisis: European Perspectives and National Discourses*; LIT Verlag: Münster, Germany, 2017; Volume 13.
31. Glassey, R.; Bälter, O.; Haller, P.; Wiggberg, M. Addressing the double challenge of learning and teaching enterprise technologies through peer teaching. In Proceedings of the 2020 IEEE/ACM 42nd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), Montreal, QC, Canada, 5–11 October 2020; pp. 130–138.
32. Poppendieck, M.; Poppendieck, T. *Lean Software Development: An Agile Toolkit*; Addison-Wesley: Boston, MA, USA, 2003.
33. Royce, W.W. Managing the development of large software systems: Concepts and techniques. In Proceedings of the 9th International Conference on Software Engineering, Monterey, CA, USA, 30 March–2 April 1987; pp. 328–338.
34. Gibbs, W.W. Software's chronic crisis. *Sci. Am.* **1994**, *271*, 86–95. [CrossRef]
35. Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; et al. The agile manifesto. *Softw. Dev.* **2001**, *9*, 28–35.
36. Krehbiel, T.C.; Salzarulo, P.A.; Cosmah, M.L.; Forren, J.; Gannod, G.; Havelka, D.; Hulshult, A.R.; Merhout, J. Agile Manifesto for Teaching and Learning. *J. Eff. Teach.* **2017**, *17*, 90–111.
37. Rico, D.F.; Sayani, H.H. Use of agile methods in software engineering education. In Proceedings of the 2009 Agile Conference, Chicago, IL, USA, 24–28 August 2009; pp. 174–179.
38. Schwaber, K.; Sutherland, J. The scrum guide. *Scrum Alliance* **2011**, *21*, 1.
39. Khosravi, H.; Kitto, K.; Williams, J. RiPPLE: A crowdsourced adaptive platform for recommendation of learning activities. *J. Learn. Anal.* **2019**, *6*, 91–105. [CrossRef]
40. Seaborn, K.; Fels, D.I. Gamification in theory and action: A survey. *Int. J. Hum.-Comput. Stud.* **2015**, *74*, 14–31. [CrossRef]
41. Crouch, C.H.; Mazur, E. Peer instruction: Ten years of experience and results. *Am. J. Phys.* **2001**, *69*, 970–977. [CrossRef]
42. Kamat, V. Agile manifesto in higher education. In Proceedings of the 2012 IEEE Fourth International Conference on Technology for Education, Hyderabad, India, 18–20 July 2012; pp. 231–232.
43. Denny, P.; Luxton-Reilly, A.; Simon, B. Quality of student contributed questions using PeerWise. In Proceedings of the Eleventh Australasian Conference on Computing Education, Wellington, New Zealand, 1 January 2009; Volume 95, pp. 55–63.
44. Haladyna, T.M.; Downing, S.M. A taxonomy of multiple-choice item-writing rules. *Appl. Meas. Educ.* **1989**, *2*, 37–50. [CrossRef]
45. Haladyna, T.M.; Downing, S.M.; Rodriguez, M.C. A review of multiple-choice item-writing guidelines for classroom assessment. *Appl. Meas. Educ.* **2002**, *15*, 309–333. [CrossRef]
46. Bates, S.P.; Galloway, R.K.; McBride, K.L. Student-generated content: Using PeerWise to enhance engagement and outcomes in introductory physics courses. *AIP Conf. Proc.* **2012**, *1413*, 123–126.
47. Fayad, M.; Schmidt, D.C. Object-oriented application frameworks. *Commun. ACM* **1997**, *40*, 32–38. [CrossRef]
48. Shull, F.; Lanubile, F.; Basili, V.R. Investigating reading techniques for object-oriented framework learning. *IEEE Trans. Softw. Eng.* **2000**, *26*, 1101–1118.
49. Lancor, L.; Katha, S. Analyzing PHP frameworks for use in a project-based software engineering course. In Proceedings of the 44th ACM Technical Symposium on Computer Science Education, Denver, CO, USA, 6–9 March 2013; pp. 519–524.
50. Sterling, S. *Sustainable Education: Re-Visioning Learning and Change. Schumacher Briefings*; ERIC: Washington, DC, USA, 2001.