Online Delivery of Intensive Software Engineering Education During the COVID-19 Pandemic

Matthew Barr
University of Glasgow
Glasgow, UK
Matthew.Barr@glasgow.ac.uk

Syed Waqar Nabir
University of Glasgow
Glasgow, UK
Syed.Nabi@glasgow.ac.uk

Derek Somerville
University of Glasgow
Glasgow, UK
Derek.Somerville@glasgow.ac.uk

***Abstract*—The COVID-19 pandemic has resulted in widespread changes to how the higher education sector operates. In this paper, the experience of delivering an eight-week undergraduate Software Engineering programme during the pandemic is discussed. The programme in question exhibits a number of unique features, including the intensive nature of the teaching, and the timing of its delivery, which coincided almost exactly with the introduction of lockdown measures. Reflections are offered on the rapid transition to online delivery of three different modules, including consideration of students' wellbeing. The implications for Software Engineering education, and online education more broadly, are considered.**

***Index Terms*—Computer science education, Distance learning**

I. Introduction

The COVID-19 pandemic has required wide-ranging and significant adjustments to virtually every aspect of our lives. In the higher education sector, perhaps the most immediate such adjustment has been the unprecedented, rapid migration to online delivery of teaching that social distancing measures have necessitated. The Graduate Apprenticeship (GA) in Software Engineering degree programme at the University of Glasgow has been no exception. Indeed, the GA programme has, in some respects, been uniquely impacted by the lockdown.

The GA in Software Engineering is a four-year, work-based degree programme developed in conjunction with industry [1]. As an apprenticeship, students are expected to spend the majority of their time learning in the workplace; however, while many degree-level apprenticeships adopt a traditional 'day release' model, whereby students' time on campus is apportioned into day-long chunks spread across the academic year, the GA programme here employs a 'block model'. This approach sees students on campus for four intensive eight-week teaching blocks across an 18-month period, with the remainder of their time spent in the workplace. The rationale for doing so is described in [1] but, in short, the benefits of such a structure include reduced context switching for students and a more rapid return on investment for employers, with apprentices mastering a larger proportion of the taught material, sooner.

However, the timing of these blocks is such that the apprentices had just begun the second of their eight-week blocks on campus when face-to-face teaching was halted. This is unlike traditional undergraduate degree programmes where, by mid-March (when the move to online-only delivery occurred), the larger part of teaching has been completed. The intensity of the teaching on the GA programme is also a unique feature: While on campus, apprentices receive four hours of instructor-led tuition almost every day for eight weeks, in addition to coursework and self-study. Three modules were delivered during the block that coincided with the lockdown, accounting for 40 of the 120 credits that comprise the entire first year of the degree – equivalent to 400 notional learning hours. Only the first week of the block was delivered on campus, prior to social distancing measures coming into force. During this week, a small number of students were already self-isolating, requiring them to join classes remotely via Zoom video link. From the second week, the modules were delivered entirely online.

The paper is structured as follows. First, the instructors responsible for delivering each of the three modules (*Practical Algorithms*, *Testing and Software Improvement*, *Web Application Systems*) provide an account of their experience, comprising personal observations supplemented by student feedback. The nature of the support offered to students is also touched upon, and the implications of our combined experience are discussed.

II. Experience Reports

A. Practical Algorithms

The module *Practical Algorithms* brings together two subjects that are conventionally taught separately: Discrete Mathematics, and Data Structures & Algorithms. Although programming is involved, the module is considerably theory-heavy. For the apprenticeship students who are 'front-loaded' with applied modules in their first semester, this second semester module was a slight departure from what they were used to. Also, due to the block teaching model, the module concentrated into eight weeks what would otherwise be taught over a 20-week period.

Over the course of the module, three distinct teaching patterns were followed:

1) Live, face-to-face lectures
2) Online synchronous lectures
3) Recorded lectures for preview, and live tutorial sessions

Module delivery began with face-to-face, two-hour classes, organized in a traditional lecture based style, and punctuated by exercises students were asked to do individually or in groups. With the transition to online in week two happening

at a very short notice, the remote version of the classroom at first simply mimicked the face-to-face classroom pattern; that is, live, synchronous, two-hour classes. The pattern of interleaving the lecture with exercises was continued, and occasionally virtual *breakout rooms* were used for group exercises.

It was soon obvious that these two-hour live online classes were tedious for both the instructor and the students. A peer-observer concurred, noting that since there wasn't a lot of interaction, and with the students preferring to keep their cameras off, the live lectures might as well be recorded and uploaded for students to view at their own pace. Based on these observations, the move was made to the third and final teaching pattern in week four: recording video lectures for students to preview, along with shorter, live tutorial-style sessions. It could be said this was a transition towards *flipped classroom* teaching; for example, [5] include previewing video lectures a key aspect of a flipped classroom. The lectures were recorded on the Zoom platform with voice over lecture slides and the instructor's video in the corner. The videos were divided into 'bite-size' chunks, ranging from five minutes to 25 minutes long, depending on where a logical place to break the video could be found. Students were asked to preview between 45 and 75 minutes of content before the live classes, which continued regularly but were shortened to much smaller, tutorial style sessions. During the live class, the instructor would address any specific questions around the preview videos for that day, circulate a tutorial sheet with problems to solve related to those preview topics, and discuss the solution of the previous tutorial, sometimes solving problems live on a virtual whiteboard (e.g. Microsoft Whiteboard, or simply using blank PowerPoint slides).

A student survey was conducted towards the end of the module. Figure 1 shows the result where students were asked to rate the effectiveness of the three teaching methods on a scale of 1-10. What is interesting to note is that students not only preferred the pre-recorded lectures over live online classes, they seem to prefer pre-recorded lectures even over live, face-to-face classes.

Students were generally positive about the move to recorded lectures in their comments as well. For example, this student expressed a view that was also echoed by others:

> The pre-recorded lectures are great as they keep you focused when listening to the content. Whereas the live online sessions, a simple distraction could lose you completely in the topic you were discussing and you're not able to rewind.

Some additional observations can also be made about what did and did not succeed in this module, based on the student feedback from the survey, and through other informal channels. The context is still online teaching, but some of these observations are transferable to face-to-face teaching as well.

Things that went well:

- Weekly online multiple-choice quizzes using the Moodle platform, with immediate explanatory feedback. Students found this helped the learning process.

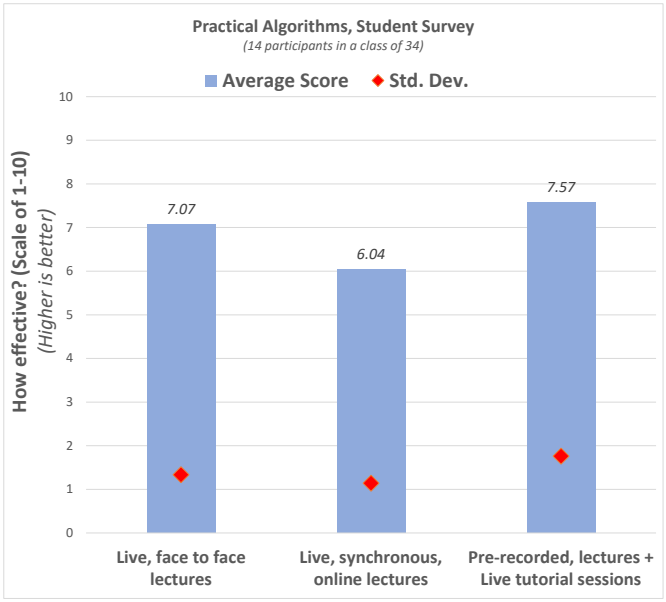

Fig. 1. Result of student survey comparing the three teaching patterns in the module "Practical Algorithms"

- Tutorial sheets: When the pattern moved to previewing recorded lectures online, the live sessions were mostly devoted to discussing and solving problems from the circulated tutorial sheets. Feedback indicated students found these very useful.
- Lab exercises: Four lab exercises were carried out, and the practical aspect in these labs was found to aid the learning.
- Links to additional material: The students were provided pointers to related material available through third parties online (e.g. YouTube, Khan Academy), which some students found very helpful.

Things that went less well:

- The rapid pace of the module was generally found to be overwhelming. Students had to preview around an hour of recorded lectures almost everyday, along with solving an accompanying tutorial sheet.
- Group activities did not seem to go very well, especially when the live classes moved online and virtual breakout rooms were used. Students indicated that there was minimal interaction, and people mostly continued working on their own. So, opportunities for peer learning were missed.
- The dedicated time for labs with a tutor was not utilized very effectively. Both the students and the tutor felt that lab sessions are better served by a face-to-face setup.
- At the beginning of the third teaching pattern, the instructor would start the live tutorial sessions by doing a quick of review of content related to videos that were to be previewed. This was generally considered redundant by the students.
- The instructor did not have a good visibility of student

engagement with the recorded lectures.

Overall, in context of this module, it could be said that the move to online teaching was not as disruptive as feared. In terms of conveying concepts via traditional lecturing, it seems as if the move online with recorded lectures was preferred over even face-to-face lectures. This is an interesting observation as it has implications beyond this COVID-19 forced situation. The theory-heavy nature of the module should be kept in mind though when interpreting these observations, as they may not directly transfer to other, more practical modules.

B. Testing and Software Improvement

Testing and Software Improvement is a ten-credit first year module that aimed to give students practical experience of development methodologies, clean code, refactoring and testing. Module delivery was split into the first two weeks and the last two weeks of the eight-week teaching block. Students were taught within a two-hour slot each day with a mixture of pre-reading, lectures, active learning discussions and labs. The summative assessments included class quizzes, a group project, an individual report and an individual project, with an exam scheduled after the teaching block. The exam was cancelled due to the COVID-19 outbreak.

The practical group project enabled students to learn and demonstrate that they understood all the learning outcomes associated with the end-to-end software lifecycle. The 34 students were split into eight teams, based on programming language preference and their ranking in a practical programming test from the first semester. The top five students were placed together to allow them to excel; weaker students were distributed across stronger teams. Creating smaller, close-knit, groups of students gelling together on their learning journey.

The testing elements of the module focused on continuous integration. To enable a quicker feedback loop, the module covered doubling: stubs, mocks and fakes. This avoided using both input and output calls (which can slow down the continuous integration test runs) and the use of external applications (which may be slow or not available during test runs).

Online Delivery Model: The testing module was intended to be delivered face-to-face. However, on the second day of teaching, two students went into self-isolation, so the delivery method became a mixture face-to-face teaching and Zoom, before all teaching was moved online. Attendance was taken before and after the move to online. The attendance was high, with only a maximum of around three students missing a class or lab. A few different approaches were used to take the online attendance but eventually the Zoom chat was used, asking students to ping the chat to say hello.

Pre-reading: Students were provided with pre-reading for lectures and asked to answer summative questions in class. Early in the first week of teaching, a student working online found the class quizzes difficult to navigate, as they had to switch between the presentation and the online quiz. The format of the answer form was a blank question and blank spaces for an A, B, C or D response, because the question was displayed on screen in the lecture (face-to-face). This was done to encourage students to attend class (so that they could not use the form outside the classroom context). The answer form was reviewed and re-designed to include answer text for online delivery.

The pre-reading was intended to encourage students' active learning, enabling them to discuss topics in class. The active learning online discussions were held in Microsoft Teams in the predefined groups used for the group projects. Later, Zoom breakout rooms were tried, where the allocation is random. Feedback from the class representative indicated that the students preferred the pre-allocated Teams groups and found the random allocation of the Zoom breakout rooms uncomfortable.

Live vs. Pre-recorded Lectures: The two-hour, pre-lockdown lectures were held face-to-face and included elements of the active learning discussion. This allowed the concepts to be covered in enough depth prior to the labs. In the move to online, the initial 30 to 60 minutes was delivered via an online live lecture, which was then followed by an online lab – this added variety whilst still covering the material. This approach was intended to break the material up into smaller topics and then allow each smaller topic to be worked on in the lab. One student highlighted that working at set times (with live lectures) helped them to structure their day during the COVID-19 lockdown.

One of the lectures included live examples of mocking and asked the students to answer questions in the Zoom chat. A significant number of students took part, putting forward their answers. This lecture was well received with unsolicited feedback from students telling the instructor they enjoyed it and the class representative also responding with positive feedback from a number of students.

Group Project Work: The group project was carried out in labs, with the first week mostly supported in person. The students were then asked to create two Teams chats: one for the students in the team and the second to include the instructor. This was intended to help keep the chat with the instructor focused on the work at hand. The instructor then held a Teams video call with each team to check progress. Periodically, the instructor checked in on the teams using the chat function and then focused on the teams that had issues.

The online labs included more practical activities to help the students complete their group work, such as creating a 'mock'. At the beginning of the second week the students had to perform a user demo to the instructor for fifteen minutes. The students also performed a *retrospective* (a reflection on how the team worked together on the project). This was a summative assessment intended to encourage early progress on the group project and provide a framework for discussing any issues. The instructor also gave feedback for improvements.

All teams managed to submit a good group project, where they had self-learnt some new frameworks – javaFX and SQLite. The exception was the team that struggled with their team dynamic; they only got a login screen to work. This team had a member self-isolate in the first week and this separation may have added to the problems with the team dynamic. This

team also had a disagreement on which language to use for the project.

The students submitted an individual report on the group project, with students providing a positive response and reporting that the project had helped them learn more about the topics. The team with the poor team dynamic submitted a weaker report; whilst they felt they had improved, their reports spent time highlighting the challenges of dealing with the COVID-19 outbreak.

C. Web Application Systems

The web applications module was delivered over the middle four weeks of the eight-week teaching block. Students had already experienced a full week of remote teaching before the module began, with a handful of self-isolating students having closer to two weeks' experience of remote learning. As such, the use of Zoom and Teams was already somewhat familiar to students.

Classes were timetabled as two-hour blocks on a near-daily basis – due to the timing of the module's delivery, a couple of days were lost to public holidays. Reflecting the nature of the learning outcomes, the module comprises a mixture of theory and practice. Lectures were to be used to deliver material relating to concepts such as n-tier architecture and the separation of presentation and logical concerns, while practical work would involve students developing a substantial piece of coursework using the Python-based web application framework, Django.

Online delivery of the module largely retained this logical divide, but with some modifications. One such modification saw lectures broken up into smaller chunks of approximately 20 minutes' duration, allowing students (and staff) regular breaks. This did not happen immediately (as, in retrospect, it should) but in response to student feedback: concentrating on a video-based lecture for up to two hours was unexpectedly tiring for students.

Another deviation from the planned instructional design saw significant class time given over to practical coursework: had this module been delivered on campus, students would have been expected to work on their web applications largely during the time between classes. Given that the time available to cover all of the planned material was already curtailed by public holidays, using class time for coursework further exacerbated the time pressures. So, while most of the lecture content was delivered live, some of the less challenging material was packaged up into short pre-recorded video lectures, to be viewed by students in their own time.

Using the timetabled class time for practical work meant that tutors could be assigned to provide support, which proved to be crucial in ensuring that students received the help they needed. Even with a relatively small class size of 34, it is virtually impossible for a single member of staff to monitor progress and provide support during labs. The absence of visual cues including body language and eye contact – particularly when students unanimously opted to keep their webcam video switched off – had the effect that students could 'suffer in silence'. Unwilling to consciously ask for help in front of the whole class and unable to signal the need for help by means of unconscious behaviours, a struggling student could easily fall by the wayside.

The solution adopted here was to divide the class into smaller groups of five or six students, each with a private chat channel on Teams. These groups were not randomly assigned: instead, they were constructed such that there was a mixture of backgrounds and abilities in each. As apprentices exposed to a wide range of workplace contexts, some had enjoyed greater exposure to web development than others. Meanwhile, students had previously shown varying degrees of competence and confidence in their programming abilities. Thus, to help facilitate peer learning and support within each group, students thought to possess a high level of relevant expertise were distributed across the groups.

With six groups in total, and a pair of tutors, each tutor was assigned three groups to assist. This arrangement meant that every group had consistent support from week to week and, if a student required help, they needed only to ask for it in front of a small number of their peers and a tutor whose sole purpose was to help. While the degree to which groups engaged with their tutor varied, the chat in most cases was remarkably active. Furthermore, students within some groups made use of the ability to jump onto a group audio call to discuss issues, and to screen share particularly sticky problems with their tutor. In many respects, Teams was an ideal platform for tutor (and peer) supported group work.

Peer support was actively encouraged, and the expectation that students should support one another within their groups was reiterated. In order to emphasise this peer support, an informal 'Most Helpful Student' award was established and voted on at the end of the module by students[1]. The award was intended to keep the importance of peer support at the forefront of students' minds, as well as acknowledging, albeit in a light-hearted fashion, the very real contribution made by the most supportive students. Indeed, this sense of fun pervaded a number of aspects of the module. Pre-recorded material, for example, contained small 'Easter eggs'[2] and challenges for attentive viewers. Easter eggs might be as trivial as including a pop culture reference targeted at a particular student or group of students, while challenges might require students to message the instructor with the answer to a problem posed in the pre-recorded lecture.

A final deviation from the planned instructional design saw a reduction in the number of short group exercises that would otherwise have taken place during lectures. In the previous, campus-based semester, the delineation between lecture, lab, and tutorial was often blurred. On the first semester module taught by the web applications instructor, a two-hour class might, for example, begin with a practical exercise, interspersed with just-in-time mini-lectures, and conclude with a more tutorial-like discussion. But such a fluid approach is

[1]Two runners-up and an overall winner received PDF certificates in recognition of their efforts.

[2]See https://en.wikipedia.org/wiki/Easter_egg_(media).

highly dependent on the instructor being able to 'read the room', and make *ad hoc* decisions based on perceived student engagement, common misconceptions or hurdles, and group dynamics. Many such subtleties are lost in an online setting, however. So, after a couple of attempts at interposing lectures with short, practical group exercises – and following consultation with students – this style of teaching was suspended, and a more traditional lecture/lab dichotomy adopted.

While the exam for this module was also cancelled, coursework assessment continued online, with some small modifications. The practical test, wherein students had to develop and deploy the simplest of web applications using the Django framework, could be run in an almost identical manner to that planned, albeit with students working remotely. The assessed group presentations took place via Zoom but were otherwise unaffected. The in-class quiz was intended to be conducted via Moodle anyway, so this ran as planned, though in 'open book' form. Finally, the more substantial coursework assignment was adjusted to relax certain technical requirements, acknowledging that, even with tutors available online, technical support was less readily obtained. The due date for this assignment was also postponed, primarily to reduce the pressure on students in difficult circumstances.

D. Student Support

The *Web Application Systems* instructor is also the students' academic adviser, responsible for their wellbeing as well as monitoring academic progress. While there is no requirement that students meet with their adviser during the second semester, it was clear that the lockdown was going to raise questions for the apprentices, and that the situation might have an adverse effect on their wellbeing. In response, Zoom-based drop-in sessions were scheduled after classes every day for the first two weeks of lockdown. While only a handful of students made use of this opportunity to talk to their adviser, and demand tailed off after the first week or so, the fact that any students came forward at all suggests that it was worth doing. These students, in need of some support, might otherwise have had nowhere else to turn.

Perhaps a more obvious success was the establishment of a private channel on Teams, with the informal title of 'Chatter'. Acknowledging that a number of students on the programme were faced with mental health challenges even before the lockdown, the Chatter channel was conceived as a space in which students could chat freely with peers who might also value having someone to talk to. The channel was advertised to all, in terms that avoided mention of mental health concerns *per se* but made it clear that the intention was to provide mutual support. Ultimately, seven of the 34 students signed up (about one fifth of the class) and, while conditions such as anxiety and depression are not often addressed directly, there is a tacit understanding that anyone on the chat might be dealing with such concerns. Three months into the lockdown, the Chatter channel remains active.

III. Discussion

The context in which the teaching described above is broadly similar across all three modules. The relatively small class size allowed for student groups to be 'hand crafted', based on knowledge of students' abilities, for example. Meanwhile, the intensity of the teaching was a feature of all three modules – all three instructors had to cover a great deal of material in a very short space of time, and in an online mode for which they were not prepared. And, of course, the context is that of a SE programme, wherein similarities across the modules would be expected, in terms of subject matter and instructional design.

However, despite these similarities, there are notable differences between the three experience reports. For example, on the web applications module, the instructor opted to retain live lectures as far as possible, while the other instructors embraced pre-recorded lectures more fully. As noted in the Practical Algorithms report, some topics are more theoretical, while others are more applied in nature – a consideration which has implications for the suitability of pre-recorded lectures, perhaps, and how the 'live' portion of the module is best utilised. Providing live interaction, whether in lecture form or a discussion session that builds on a pre-recorded lecture, was found to be critical. Informal feedback on pre-recorded lectures vs. their synchronous equivalent suggested that students appreciated the social interaction that came with the latter. However, as described above, such interaction may be recreated using a flipped approach, and may be effective where blended delivery is adopted – a likely scenario immediately post-lockdown.

Although the recorded lectures in this new model seemed to have worked well, that is just one aspect of a flipped classroom model. When all conventional "lecturing" is recorded for previewing, then the use of live classes, in-person or online, can be devoted to complementary exercises and activities. Tucker [3] emphasizes that what makes a flipped classroom approach effective is not just the online videos, but how well they integrate into an overall approach. It was clear from student feedback that tutorials, lab sessions and group activities suffered for lack of in-person interaction. More thought needs to go into designing exercises and activities in the flipped classroom model, e.g. whether the exercises are pitched at individuals or group, or – in view of the lockdown – if they are suited for in-person or virtual classroom, as the case may be.

Prior work around flipped classroom teaching and related areas can be brought to bear here; e.g., [2] highlight a number of "small group teaching" techniques, noting how their efficacy is perceived by teachers and students. The "just-in-time teaching" approach [4] may be useful in the context of making the best of synchronous sessions, which requires students to engage in a short activity on their own after going over some preview material and before coming to a live class. A pitfall to be avoided is something that was noted for teaching children, but is valid in the current context as well:

"...many schools might think they are implementing peer tutoring or cooperative learning, when all they are really doing is putting children together and hoping for the best" [6]. Online synchronous classes, then, provide an excellent opportunity for clearly specified group activities and collaborative learning, as well problem solving sessions, and lab exercises.

The reports above also reveal differences in how successful online group work proved to be. On the algorithms module, this is identified as a less successful aspect, with students opting to work in isolation or being unsure how to proceed at all when dispatched to their groups (ameliorated by the introduction of detailed tutorial handouts). On the testing module, poor group dynamics were identified as a potential challenge. On reflection, the importance of setting a clearly defined task before sending students into online breakout groups is even more important in an online setting than in a traditional lab or tutorial. And, while group dynamics are a perennial issue in SE education and beyond, it is possible to minimise intra-group clashes through careful consideration of each group's membership – provided the instructor responsible for creating the groups is sufficiently familiar with the personalities and abilities of the students involved. This was the fortunate situation on the web applications module, where group work appears to have operated most smoothly. Where larger cohorts preclude taking such a 'hand crafted' approach to constructing student groups, randomised and non-persistent groups may offer a partial solution. Randomisation should facilitate distribution of students with varying abilities, while eschewing fixed groups can help ensure that fractious groupings are only temporary.

The group work also illustrates how the technology – specifically Zoom and Teams – supported students' learning. While breakout groups must be used with care, as noted above, the use of Zoom in conjunction with Teams proved to be an effective configuration. Using Zoom to host the 'main' class, and using Teams to host the groups, allowed continuous contact with the instructor (akin to remaining in the room while students break into groups) whilst also affording the students all of the features of the Teams platform. These features include not only the ability to screen share, but also for the students to start an audio call, independent of the main class. This sort of flexibility, coupled with clearly defined group tasks and carefully constructed group memberships, here provided the optimum learning experience.

The use of technology also presented some challenges. Alluded to in all of the experience reports above is the issue of lecture length and the time a student may engage with online video while maintaining their concentration. In all cases, classes were split up into smaller chunks, with regular breaks during live lectures and recorded lectures being offered as a series of shorter presentations. Whatever the causes – lack of visual cues, prolonged periods of being seated, or something else – 'Zoom fatigue' is a very real phenomenon. It is also notable that students generally opted not to turn their video on, reducing the Zoom experience to little more than a live stream and further reducing the instructor's ability to monitor student engagement. However, we must consider that not all of our students' home environments are suited to being broadcast to their tutors and peers before insisting that video is turned on. Indeed, it was generally found that the text-based chat provided a perfectly sufficient second channel of communication for students to ask questions or make suggestions while the lecture proceeded.

Finally, the importance of ensuring students' wellbeing should not be underestimated. This is particularly important under these circumstances, where the pandemic has introduced significant uncertainty and anxiety, and left many students isolated. Students' wellbeing is a consideration that clearly extends beyond SE education. However, extensive experience of delivering online SE education under lockdown conditions – where a fifth of the class has signed up for additional ongoing support and other students have sought help on an *ad hoc* basis – suggests a potential deleterious impact on academic achievement if pastoral support is not considered.

IV. Conclusion

The experiences described here demonstrate that Software Engineering education may be successfully delivered online with some relatively straightforward adjustments. Indeed, for these modules, there was simply no time to develop an entirely new instructional design in response to the lockdown. It is also apparent that even across these closely related modules, there is no 'one size fits all' approach to online delivery of SE education. However, as discussed above, a number of useful, generalisable observations may still be made.

Following established best practice such as apportioning online lectures into smaller 'chunks', carefully considering group work and group composition, and paying close attention to student feedback to inform 'in flight' adjustments, it is still possible to deliver a pedagogically sound experience, even under lockdown conditions. Indeed, there are elements of this enforced distance learning paradigm that we may wish to retain in the post-pandemic era.

References

[1] Matthew Barr and Jack Parkinson, "Developing a Work-based Software Engineering Degree in Collaboration with Industry," in Proceedings of the 1st UK & Ireland Computing Education Research Conference, Canterbury, United Kingdom, Sep. 2019, pp. 1–7, doi: 10.1145/3351287.3351292.

[2] Bogaard, A., Carey, S.C., Dodd, G., Repath, I.D. and Whitaker, R., 2005. Small group teaching: Perceptions and problems. *Politics*, 25(2), pp.116-125.

[3] Bill Tucker. The Flipped Classroom, October 2011. Library Catalog: www.educationnext.org Section: Briefs.

[4] Gregor M. Novak. Just-in-time teaching. *New Directions for Teaching and Learning*, 2011(128):63–73, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/tl.469.

[5] Amy Roehl, Shweta Linga Reddy, and Gayla Jett Shannon. The Flipped Classroom: An Opportunity To Engage Millennial Students Through Active Learning Strategies. *Journal of Family & Consumer Sciences*, 105(2):44–49, August 2013.

[6] Keith J. Topping. Trends in Peer Learning. *Educational Psychology*, 25(6):631–645, December 2005. Publisher: Routledge _eprint: https://doi.org/10.1080/01443410500345172.