Review of factors that make for successful online introduction to programming courses.

Moses Netshitangani
NTSNDI017
University of Cape Town

ABSTRACT

Technological advances have made it possible for a multitude of courses to be offered online, although there are some known problems with this method of learning, one of them being high drop-out rates. The purpose of this review is to investigate the critical success factors that are essential in the creation and delivery of online courses. We review and compare different literature and identify the key success factors that are most effective in creating a productive online learning environment for students enrolled in introductory programming courses, such as short lecture videos, informative feedback for assessments, gamification, technical competence, interactive learning, and self-regulated learning.

CCS CONCEPTS

• Social and professional topics ~ Professional topics ~ Computing education ~ Computing education programs ~ Computer science education ~ CS1

KEYWORDS

Online course, Learning Management System, Critical Success Factor, Integrated Development Environment

1. INTRODUCTION

Online learning is changing the way in which courses are offered to students. It is a method of learning where course materials (notes, slides, etc.) are made available to students, through use of the internet, using Learning Management Systems (LMSs) as a mode for delivery. LMSs are platforms that facilitate learning and teaching and provide a single, centralized repository for hosting course resources. Online learning can be beneficial in that it affords academic institutions the capability of reaching students from a wide geographical range [5]. For the students, this can prove to be advantageous. It allows them flexible access to the course materials, they benefit from forums/chatrooms by seeing how other students are thinking, and shy students who would rather not say anything in lectures can now use forums and chatrooms to ask questions [2, 6, 8, 12].

However, there are certain limitations that have not been addressed yet, or have but ineffectively, in online courses. This review discusses a subset of the Critical Success Factors (CSFs) that can be used to stretch some of the limitations imposed by online courses, and their relevance in introductory programming courses. But due to the scarcity of relevant literature in these specific courses, significant points found in literature addressing general online courses has also been considered. We realize that these points apply to online courses in general, and thus can be deemed relevant, to some extent.

Learning programming is already a difficult task, and doing it online adds on to the difficulty, which contributes to higher drop-out rates [17]. Even if students do manage to complete their undergraduate studies in programming, most of them do so with just sufficient general knowledge in the field, and lacking the specific skills to create complex applications. Some of the main issues that students have in an online learning setting are the lack of adequate social interactions, and motivation [14]. Appana [2] lists asynchronous feedback, poor student retention and completion rates of assessments as some of the issues. Campbell et al. [3] also noted that first year computer science students were less likely to finish the whole course than students who were offered the same course but using the traditional method of teaching. It is also worth noting, that the latter study compared the final exam scores for both groups of students (ignoring those that had dropped out in online courses), and found no significant difference, although poor course completion rates were seen mostly in the online group.

Inconsistencies with the Integrated Development Environments (IDEs) used, the availability of timely feedback, unavailability of an efficient way for learner-tutor editing and discussion of code in real time, and limitations to learner-learner and learner-instructor interactions are issues that have been reported in online introductory programming courses [19]. It becomes difficult for instructors to address each of the students' issues if they're using different IDEs. The learning is also halted when students cannot get timely and meaningful feedback from instructors/tutors and automatic grading systems, about how their code is supposed to work. These grading systems are known to just notify students that their code has failed certain test cases, but do not provide any hints about how to solve the issue. Therefore, code that is close to working would receive the same mark as code that does not even attempt to solve the given problem.

In order to appreciate the importance of CSFs, we first need to understand the current issues with online learning, and their causes. Section 2 addresses this as it gives a brief overview of the two types of online learning delivery modes, and the issues that accompany them. Section 3 then discusses the CSFs that can be used to solve some of these issues. Section 4 discusses supportive and contradictory points identified in the literature. Section 5 holds the conclusions.

2. MODES OF DELIVERY AND THEIR LIMITATIONS

LMSs play a significant role in the administration of online courses, by allowing instructors to upload resources so that they are accessible to students. Methods of delivery do vary, and this review will discuss two: synchronous and asynchronous, and their limitations. It is also common for academic institutions to use a combination of both methods to cater to all students [18]. For example, a web conference may be held for students who are able to attend (synchronous), and the conference can be recorded and later uploaded as a lecture video for students who could not attend in real time (asynchronous). We do note that some of the issues are present in both methods of delivery, but we will only discuss them once.

2.1 Synchronous Delivery

In the synchronous method, LMSs are used to host web conferences, which closely mimic the traditional method of teaching, by affording students and the instructor the opportunity to engage in real time through use of video and audio systems. The interaction of learners and instructors through forums and chatrooms can also be synchronous, if the questions being asked are being responded to immediately after. In introductory programming courses, this method contributes to a pool of known problems:

2.1.1 Excessive Focus on Programming Language

The instructor's goal is to equip students with programming skills and techniques, not to teach them a specific programming language. Yet, introductory programming courses spend a great amount of time teaching language constructs and syntax to students. Hulls et al. [9] found that a total of 40% of the course notes for their introduction to programming course dealt with language syntax and constructs, which takes away from the 'actual' learning that is important for the students to know in order to improve their programming skills. For students that already have some coding background, the focus on syntax can be boring, while for the inexperienced students it can lead to confusion and frustration [9]. It is worth noting that this issue is not unique to online courses and is also present in face-to-face learning.

2.1.2 Poor Instructor Characteristics

The instructors play a central role in online courses. They convey information to the students, and improve their understanding. Their characteristics need to change if online learning is to be effective. Volery et al. [18] concluded that instructors have to inspire interactive teaching methods and encourage interactions between everyone. More often than not, instructors are not given proper training when transitioning from face-to-face to online teaching, and not all instructors are willing to change their teaching style to improve the 'effectiveness' of online courses [10]. The unwillingness to teach online comes from the fact that instructors are used to teaching face-to-face and some may not feel delighted because of the effort that goes into creating and offering online courses. For some, it is just the fear of not being able to connect with students in an online setting [8]. Most of the time, instructors can't even see the faces of the students, so the disconnect is really there.

2.1.2 Poor Student Retention and High Drop-out Rate

Online learning makes it hard for instructors to gauge student participation. Instructors should accept that students tend to fall victim to many distractions, one of them being the internet (the very tool that allows for online learning) [1]. While Lam et al. [13] bring up difficulty of the course as a cause for poor student retention and high drop-out rates, Appana [2] notes that issues of student retention and drop-out rates are yet to be researched thoroughly, which makes it hard to tackle them with certainty. Some studies, however, disagree that online learning leads to poor student retention rates. Chase et al. [4] say that online learning in a programming environment actually improves student retention rates, and that their introductory course saw an improvement from 68% to 75%. However, we do note that their results were subject to bias, since their course was coupled with an orientation course (called UNIV 100) that ensured students were equipped with basic programming skills, prior to the actual introductory course. Therefore, the issue of student retention and drop-out rates is still valid.

2.1.3 Proficiency with Technology

Problems with technology proficiency are still relevant. Both students and instructors fall victim to them. However, it is mostly important for instructors to be familiar with the LMS being used, to be able to perform basic tasks such as admitting a student, setting up quizzes, and uploading resources [18]. Students also need to know how to interact with the LMS's interface to locate resources, and be able to take assessments.

2.2 Asynchronous Delivery

In the asynchronous method, course materials are uploaded by the instructor to the LMS, which gives students 'anytime' access, and encourages self-paced learning. This leads to some issues:

2.2.1 Limited Human Interactions

An online setting removes the 'physicality' aspect of traditional face to face lectures. Communication is limited and it can appear as an 'individual' task which makes students feel 'isolated' from the rest of the class, and this contributes to the insufficient interactions that occur [1, 12]. Teaching programming requires extensive demonstration of techniques, examples/tutorials, and interaction with the students, which makes online delivery even more challenging [6]. Even when interactions do occur, they usually happen on forums and lose their value because they happen asynchronously, with delayed replies and sometimes no replies at all. Krasnov et al. [12] however, concluded that online learning had positively impacted learner-instructor interaction, which contradicts the prior studies. We feel that the interaction has not been improved since most of it takes place in forums, where replies can suffer from delays, like the other studies say.

2.2.2 Uninformative Feedback

Code grading systems are designed to run the submitted code against multiple test cases, and award marks for each case passed. These systems do not provide any informative feedback on how to correct the code, which leads to student frustration [19]. They mainly help with detecting syntax errors, but the main issue in resolving the code is to find the logical faults that make the code fail to work as expected [13]. It would also be unreasonable for the staff to provide manual feedback to all the students, in an efficient and timely manner.

2.2.3 Real Time Tutor Accessibility

Learning programming for the first time can be difficult and having an experienced guide by your side can be helpful. Students have tutors for this reason, but the problem is they have no effective way of accessing tutor help in real time without having to publicly share their code with the rest of the class (perhaps on the forum) [19]. The use of email to send code and receive feedback/help from the tutor is not an effective way of communication. Migration to a different application other than the LMS circles back to the problem of technological proficiency.

2.2.4 Video Quality Demands

To produce quality educational videos, the weight is all placed on the instructor's shoulders. Producing quality video material takes a serious amount of time. It might take between five and twenty hours to produce a single high quality one hour lecture video, and the whole process might even require several parties, to handle different parts of the whole process [11]. Video length is also a concern. It becomes very difficult to pluck out the main points and remain focused, when the videos are too lengthy [10].

3. CRITICAL SUCCESS FACTORS

CSFs in an online learning context are the factors that are at the core of successful online courses. They help solve issues that prohibit productive teaching and learning, some of which were identified in section 2. Many studies identify different kinds of CSFs, and this review will only discuss a subset of them.

3.1.1 Supportive Learning Environment

Learning environment refers to all the aspects surrounding the LMS, that can either enhance or impede teaching and learning. Rafique et al. [17] concluded that in order for programming courses to be effective, academic institutions need to provide support to the students, by using LMSs' live conferencing to enhance engagement and address issues. Their study goes further to say that students perform better when they have secure social environments, human interactions and instructor feedback. This complements a study by Cheawjindakarn et al. [5] which found that students learn better when the learning environment is supportive, 'familiar', and comfortable. Alqahtani et al. [1] also listed learning environment as one of the CSFs. Their results were based on a survey they conducted, that proved the significance of LMSs in the learning environment, and was conducted during the COVID-19 pandemic, when most academic institutions were 'forced' to adopt online learning without any adequate level of preparedness [1].

3.1.2 Short Lecture Videos

A huge part of online learning includes the use of pre-recorded lecture videos. Students find watching lecture videos to be more helpful and 'practical' than reading course materials [11]. A study by Guo et al. [7] concluded that shorter and specific videos (0-3 minutes) were best received by students, videos with instructor heads felt more personal, and videos where instructors spoke with enthusiasm and with a relatively fast tone were more engaging. Research by Krasnov et al. [12] concluded that the length of the videos need not exceed 15 minutes in length, if they are to be effective. These studies were conducted using data from e-learning course providers who use sites such as YouTube to upload content, and not actual data from academic institutions. They measured the success of this CSF using factors such as the number of views and likes. We find that data from this study is still relevant as students are known to watch course-specific tutorials from sites such as YouTube. Volery et al. [18] however, noted that pre-recorded lecture videos are sometimes the least used feature of LMS, especially when students attend the live

conferences. We still believe that shorter lecture videos are a CSF, in the event that some students fail to attend the live conferences.

3.1.3 Interactive Learning

Introductory programming courses require demonstration of coding techniques, and effectively student-instructor and student-student interactions [6]. This CSF can be measured by the number of question-answer pairs in forums, and surveys to gauge the student satisfactory levels of interaction within the course [15]. LMSs should allow for student-student interactions as students often find it easier to discuss problems with their peers, than with other people [17]. Live conferences should be used as a catalyst for student-teacher discussions that assist students with grasping the content of the course. Campbell et al. [3] say that computer science students in their study were very active in asynchronous media such as forums and chatrooms. Studies note that these tools are used to post/answer questions, which can also supplement the understanding of those who are not able to attend the live conferences [6]. Krasnov et al. [12] noted that in order to be effective, answers to questions in the forums and chatrooms should be posted within 24 hours of being asked.

3.1.4 Gamification

Most students lose interest in lectures and stop attending. Chase et al. [4] conducted a study on a first year programming course and found that student retention rates had increased from 68% to 75%, when students were offered a programming orientation course prior to the start of actual classes. This adds on to a study by Hulls et al. [9] which found that programming introductory courses spend a lot of time teaching syntax instead of actual programming techniques. This suggests that some form of coding background helps at least maintain current student retention rates. The use of gamification to improve retention rates has also been tested. Its use results in students who are motivated and spend more time than average interacting with the course, due to the 'gaming' experience offered by the LMS [11]. The structure of some assessments is altered to give a 'game' feel, and the effectiveness of this CSF is measured by number of submissions/attempts as well as the feedback from surveys [16]. Students are rewarded with badges and other forms of incentives when they complete quizzes and score good marks. This keeps them focused and motivated. Piccioni et al. [16] saw a 500% increase in quiz attempts after they introduced gamification in their programming course.

3.1.5 Informative Feedback

Students need feedback when their programs do not work as expected. Yet, most of their projects are usually marked by an automated system. This makes sense since freshmen classes tend to be very populated. Warren et al**.** [19] noted that improvements to the automatic grading systems do help. The researchers reported that their improved automatic grader resulted in 2/3 of informative feedback on 'incorrect' submissions, for programs that were not too complex. To test this CSF, changes in assessment scores can be noted, and student surveys can be carried out to determine its usefulness.

3.1.6 Technical Competence

This CSF has to do with the ease that students and the instructor can access and navigate the LMS. More importantly, the instructor should be able to use the LMS and other accompanying technologies to provide a smooth online experience [18]. Faculties must provide lab sessions where the instructors and students can be taught how to use the LMS. Instructors should also encourage students to improve their technical skills, by giving out projects (such as voice-over presentations) which require familiarity with new technologies [5].

3.1.7 Self-Regulated Learning

Self-regulated learning (SRL) has to do with students orienting their focus and behaviour towards achieving a learning outcome [3]. There has not been much research into SLR in a computer science setting. Cheawjindakarn et al**.** [5] argue that SLR skills by themselves are not sufficient to have an impact on performance, and that students also required self-motivation in order to properly channel them. We do note, that although their study addresses SLR, it was conducted in a face-to-face environment, and may not be exactly relevant.

3.1.8 Code Exposure

In programming, it is important for students to be able to understand and work with pre-existing code. In order to improve students' understanding of such code, instructors are encouraged to have quizzes in which they ask students to 'complete' an incomplete program, by filling in missing parts from the choices provided [16]. When instructors include examples in slides and other course materials, the LMS being used should grant students the ability to edit this code, so that they can be able to 'control' the environment and learn by observing what happens when they change certain parts of the example code [9].

4. DISCUSSION

The study by Piccioni et al. [16], unlike others, lists attention span recovery as a CSF [16]. Their methodology included breaking lectures in between, to allow students to take quizzes about content they just learned, and rewarding the students with badges after successful quiz completion. This demonstrates the close relationship that some of these CSFs have. For example, in their methodology, gamification, which has been widely identified as a CSF, is used to help boost the students' attention span recovery,

which is also a CSF. It becomes difficult to address each CSF on its own without sounding repetitive, due to the similarities they possess. This is also true for several other CSFs.

In the discussion about creating supportive learning environments as a CSF, Rafique et al. [17] concluded that academic institutions need to provide support to the students, by using LMSs' live conferencing to enhance engagement and address issues, and overall improve the online programming courses. A study by Cheawjindakam [5] et al. supports this, but it also notes that LMSs alone do not have a significant impact on the students' and instructors' learning outcomes. The methodology used in their paper was merely the synthesis of several literatures spanning from 2000 to 2012.

In discussing short video length, Guo et al. [7] concluded that short videos where the instructor spoke quickly and with enthusiasm, were more engaging. Enthusiastic instructors sounded like something to consider, however instructors speaking fast was not too convincing, and Piccioni et al. [16] found that for beginner programmers, frustration levels were actually high when instructors were speaking fast. It is worth mentioning that their study failed to mention any method(s) used to measure the levels of frustration.

When it comes to interactive learning, studies such as those conducted by Campbell et al. [3] found that beginner programmers were highly active in asynchronous media such as forums and chatrooms, where they regularly posted and provided answers to peer questions. Research by Rafique et al. [17] however, demonstrated that forums and chatrooms had a negative impact on student performance. Their study conducted several experiments with computer science students and found that synchronous communications were the ideal, as the presence of an instructor encouraged interactions and 'guided' learning. It does not dive further into exactly how forums and chatrooms had a negative impact on student performance, and it is also important to note that as in synchronous media, instructors are also present in forums and chatrooms. Therefore, this does not completely rule out forums and chatrooms, as studies note that these tools are useful (mostly when questions are answered within 24 hours) and supplement the understanding of those who are not able to attend the live conferences [6, 12].

While the study by Warren et al. [19] speaks of the effectiveness of improving automatic code grading systems to offer informative feedback, there is a lack of literature surrounding this topic, and the study itself noted that providing feedback for complex applications would be beyond the scope of our current technology.

Hulls et al. [9], who researched quiz gamification in an introductory programming course found that students really liked the quizzes more and found them to be interactive and helpful [9]. The data from their study, however, was constrained as it was obtained from a very small subset of the students, who gave consent to the study.

5. CONCLUSIONS

Most CSFs are composed of other CSFs. For example, multiple studies have identified Interactions and Instructor Characteristics (not addressed in this review) as two, separate CSFs, but in order to carry out the objectives that address Instructor Characteristics, studies say instructors need to encourage 'Interactions' among students [8]. The introduction of a programming orientation course, as in the study by Chase et al. [4], saw a significant increase in Student Retention rates, even though this 'supplementary' course has not been identified as a CSF on its own. From this we can conclude that there are multiple elements that help carry out the objectives of CSFs, and some of those elements may be other CSFs.

Various studies have identified different CSFs in different learning settings. This review discussed those that appeared multiple times in literature, and those that seemed 'relevant' in online introductory programming courses. We feel that Informative Feedback from automatic grading systems is the most relevant, because this is where students apply the knowledge and skills they have learnt, and to have a system that provides them with feedback about where their code logically went 'wrong', re-enforces their understanding of programming concepts. The lack of literature in these specific courses, however, means there is a gap in research, as we cannot say with certainty that the other CSFs we identified would also be ideal in online programming introductory courses, which is what we are interested in. Further research about these courses would be required.

6. REFERENCES

[1] Alqahtani, A.Y. and Rajkhan, A.A. 2020. E-learning critical success factors during the covid-19 pandemic: A comprehensive analysis of e-learning managerial perspectives. *Education Sciences*. 10, 9 (2020), 1–16. DOI:https://doi.org/10.3390/educsci10090216.

[2] Appana, S. 2008. A review of benefits and limitations of online learning in the context of the student, the instructor, and the tenured faculty. *International Journal of E-Learning*. 7, 1 (2008), 5–22.

[3] Campbell, J. et al. 2016. Factors for success in online CS1. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. 11-13-July, (2016), 320–325. DOI:https://doi.org/10.1145/2899415.2899457.

[4] Chase, J.D. and Okie, E.G. 2000. Combining cooperative learning and peer instruction in introductory Computer Science. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*. (2000), 372–376. DOI:https://doi.org/10.1145/331795.331888.

[5] Cheawjindakarn, B. et al. 2012. Critical Success Factors for Online Distance Learning in Higher Education: A Review of the Literature. *Creative Education*. 03, 08 (Jan. 2012), 61–66. DOI:https://doi.org/10.4236/ce.2012.38b014.

[6] El-Sheikh, E. 2009. Techniques for Engaging Students in an Online Computer Programming Course. *Systemics, Cybernetics and Informatics*. 7, 1 (2009), 1–12.

[7] Guo, P.J. et al. 2014. P41-Guo (1). *Proceedings of the first ACM conference on Learning@ scale conference*. (2014), 41–50.

[8] Al Hamad, A.Q. et al. 2014. Key Factors in Determining Students' Satisfaction in Online Learning Based on 'Web Programming' course within Zarqa University. *International Journal of Global Business*. 7, 1 (2014), 7–14.

[9] Hulls, C.C.W. et al. 2005. First Programming Course. *IEEE Transactions on Education*. 48, 4 (2005), 719–728.

[10] Kebritchi, M. et al. 2017. Issues and Challenges for Teaching Successful

Online Courses in Higher Education. *Journal of Educational Technology Systems*. 46, 1 (2017), 4–29. DOI:https://doi.org/10.1177/0047239516661713.

[11] Korkut, S. et al. 2015. Success factors of online learning videos. *International Journal of Interactive Mobile Technologies*. 9, 4 (2015), 17–22. DOI:https://doi.org/10.3991/ijim.v9i4.4460.

[12] Krasnov, S. V. et al. 2018. Problems of Quality of Education in the Implementation of Online Courses in the Educational Process. *International Conference on High Technology for Sustainable Development, HiTech 2018 - Proceedings*. (2018), 0–3. DOI:https://doi.org/10.1109/HiTech.2018.8566618.

[13] Lam, M.S.W. et al. 2008. Designing an automatic debugging assistant for improving the learning of computer programming. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 5169 LNCS, 6000145 (2008), 359–370. DOI:https://doi.org/10.1007/978-3-540-85170-7_32.

[14] Muilenburg, L.Y. and Berge, Z.L. 2005. Students Barriers to Online Learning: A factor analytic study. *Distance Education*. 26, 1 (2005), 29–48. DOI:https://doi.org/10.1080/01587910500081269.

[15] Picciano, A. 2002. Picciano (2002).pdf. *Journal of Asynchronous Learning Networks*. 6, 1 (2002), 21–40.

[16] Piccioni, M. et al. 2014. SPOC-supported introduction to programming. *ITICSE 2014 - Proceedings of the 2014 Innovation and Technology in Computer Science Education Conference*. (2014), 3–8. DOI:https://doi.org/10.1145/2591708.2591759.

[17] Rafique, W. et al. 2020. Factors influencing programming expertise in a web-based e-learning paradigm. *Online Learning Journal*. 24, 1 (2020), 162–181. DOI:https://doi.org/10.24059/olj.v24i1.1956.

[18] Volery, T. and Lord, D. 2000. Critical success factors in online education. *International Journal of Educational Management*. 14, 5 (2000), 216–223. DOI:https://doi.org/10.1108/09513540010344731.

[19] Warren, J. et al. 2014. Facilitating human interaction in an online programming course. *SIGCSE 2014 - Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. (2014), 665–670. DOI:https://doi.org/10.1145/2538862.2538908.