Review of the Literature

Introduction

Previous work done has shown there are benefits in adding computational thinking into school curriculum. Prior work has shown that students can achieve greater problem-solving skills and confidence in their learning abilities when computational thinking activities are utilized. The work also looked at methods of integrating professional thinking into lessons that achieve both the learning outcomes intended for the subject and measurable outcomes for problem solving growth. Despite the gains in learning, implementing any type of innovation strategy or transformation effort into an organization is difficult. The goal of this review is to identify key areas to help ensure the success of integrating computational thinking into core content classes.

Grizioti and Kynigos (2018) investigated the application of computational thinking within a teacher preparation program for mathematics teachers. The research explored the effects of integrating computational thinking within a mathematics curriculum. The researchers concluded that the integration of computational thinking improved conceptual understanding of mathematics topics and the application of those topics. Furthermore, the study found that better assessment results were achieved when computational thinking was integrated within the mathematics content instead of being a separate topic. More consideration should be placed on integration of computational thinking into other subjects (Voogt, Fissher, Good, Mishra, & Yadav, 2015).

Kalelioglu, Gulbahar, and Kukul (2016) conducted an extensive analysis of over 100 papers on computational thinking. The study highlighted several areas of interest for this review. Initially is important to note that the study found within the papers investigated that there was overwhelming consensus that the implementation of computational thinking in a K-12 classroom lead to improved student outcomes. These gains were observed regardless of the implementation methods or the type of assessment. Their work supports further investigation into the effective use of computational thinking within education.

Teacher Preparation

Yadav, Mayfield, Zhou, Hambrusch, and Korb (2014) explain the necessity for quality teacher preparation when implementing computational thinking in the classroom. The classroom instructors do not have a firm, concrete, and deep understanding of computational thinking. Therefore, the application of computational thinking within the context of their subject area will only be at surface level. If teachers only have an abstract understanding of computational thinking, then the use within the classroom will end up being surface level and not fully integrated within the content itself (Grizioti & Kynigos, 2014). Gadanidis, Cendros, Floyd, and Namukasa (2017) support this in their finding that most new teachers are not adequately prepared to teach computational thinking. Teachers that are not fully aware of computational thinking only have a surface level understanding of the concept. This will limit the gains seen in learner outcomes.

The study conducted by Kalelioglu, Gulbahar, and Kukul (2016) point out that research and specific curriculum regarding computational thinking are relatively new. Currently there has not been much analysis on effective strategies and implementations of computational thinking in the classroom. At the time of the research, there were only about 500 studies that had been concluded. The relative lack of depth in research into computational thinking necessitates an analysis of teaching methods and implementation strategies in other areas. Furthermore, the reliance on these studies alone for support can lead to unforeseen difficulties given the short time span they cover. Analysis of other areas is necessary.

The case study into the implementation of an English language program in Turkey by Kirkgoz (2008) provides a detailed look at the implementation of an innovation strategy in different field of education. The researcher noted that teachers in any organization can have a lot of variation in their understanding and practices within the classroom. The similarity to the variation within teachers understanding of computational thinking by Gadanidis (2017) helps align the results. Kirkgoz's work provides several key factors to be noted in the design of an innovation strategy. These factors are related to the quality and effectiveness of teacher preparation.

Quality training programs are important for both organizational trainers and teachers. The trainers must have a thorough understanding of not only training methods, but also the content that the practicing teachers will be using. Lack of preparation for the trainers will be reflected in poor output by the teachers and then repeated in the classroom. Trainers within an innovation plan need to be respectful of the current realities of the teachers they are working with. Not all teachers are ready or prepared in the same manner or level. Assuming that all teachers are equally ready for the start of a program will lead to struggles or even prevent the successful implementation of the strategies (Dushku, 1998; Wedell, 2003). Implementation plans and organizations that do not provide ongoing support to teachers after the initial implementation will most likely lead to the changes not being long-lasting or complete (Kirkgoz, 2008).

Instruction Considerations

Gadanidis, Cendros, Floyd, and Namukasa (2017) note that teacher attitude directly affects the effectiveness of instruction within the classroom. This is important to remember from both a teacher preparation and strategy implementation standpoint. The confidence the teacher has in regard to the implementation of computational thinking will be reflected within the classroom environment and then reflected again in the student learning. Vallance and Towndrow (2016) support this by showing that learning design and quality teaching practices have a stronger impact than the activity itself. When the same activity was presented to students, low quality instructional methodologies and negative instructor attitudes resulted in lower levels of learner achievement.

Jeon, Kim, Lee, and Kim (2016) investigated the effectiveness of a specific implementation method for computational thinking in their research. From this analysis, the key finding was that blended learning does not have a negative impact on implementation of computational thinking within the classroom. Gadanidis (2017) investigated both technological and "unplugged" activities. Unplugged activities are classroom strategies and methods that do not directly involve technological devices. Both activities demonstrated the ability to enhance learning and provide growth in student achievement. Technology can provide tools to improve learning, but it needs to be implemented intentionally. Providing the proper opportunities, the implementation of technology can enhance learning outcomes. At the same time when technology is misapplied the learning outcomes and computational thinking gains were greatly diminished, or not realized at all (Vallance and Towndrow, 2016).

Kalelioglu, Gulbahar, and Kukul (2016) found that teachers must have a strong definition of computational thinking in order to have a positive influence on student performance and integration with core subject matter. The work shows that many times teachers were not confident in their definitions or understanding of computational thinking strategies. This was shown through the use of incorrect, ambiguous, and conflicting definitions. These situations lead to students only applying computational thinking in a very superficial level and with no lasting impact beyond the lesson. Finally, expectations on effectiveness of implementation strategies and student outcomes needs to be balanced with the limited amount of research.

Assessment Considerations

Djambong and Freiman (2016) explain that there appears to be a need for better or computational thinking assessments. Teachers have limited experience in assessing computational thinking and turn to a limited number of tools. Their research shows that the type of assessment seems to have an influence on the validity of the results. It is therefore important that teachers select quality methods of assessment so as to properly evaluate the desired learning and problem-solving skills. Any attempts to initiate a project that implements computational thinking in order to develop problem-solving of learners must provide quality assessment tools to gauge the effectiveness of the implementation strategies and the methods used.

From the research there appears to be substantially more assessment information for teachers in the thought process than any final answers given by learners. Computational thinking activities provide opportunities where there are more than one right answer and more than one way to achieve a useful result. For this reason, the results the student arrives at is not as important as the method and insights gained in the process of arriving at the result. Assessing open-ended situations such as the thought process in an investigation requires new assessment tools and methods (Djambong & Freiman, 2016; Dolgopolovas, Jevsikova, Dagine, & Savulioniene, 2015).

Student Preparation Considerations

The work of Papadakis, Kalogiannakis; and Zaranis (2016) provide useful information as to the student needs of computational thinking activities in the classroom. The work showed that students as young as preschoolers can learn computational thinking skills. This information is useful for most teachers. It allows for teachers with wide ranges of students to be more comfortable being able to effectively introduce computational thinking activities to their learners. The research further showed that the computational thinking activities must be age-appropriate to be most effective. In order for all learners to effectively employ computational thinking activities, teachers must be aware of the unique needs of each of their learners to provide the best opportunities. Finally, teachers must be consistent in their application and terminology. Different methods can produce different results. Intermixing methods and definitions in the early stages produces fewer effective results.

As teachers create to activities to improve critical thinking and problem-solving skills for their students through the application of computational thinking activities, they need to be mindful of the challenges they provide to the learners. Yiannoutsou, Kynigos, and Daskolia (2014) showed that computational thinking activities that are open-ended improve critical thinking skills. The most important factor found was that the activities needed to be challenging to learners. Traditional lessons and activities about the steps of computational thinking were found to not be effective or engaging. Giving an authentic problem to learners to solve increases engagement over traditional projects and assignments. The increased engagement provided opportunities for deeper and longer lasting learning (Grizioti, & Kynigos, 2018).

Providing students challenges through the use of “half-baked design” concepts, provided significantly greater engagement and problem-solving gains. Half-baked challenges are projects that require manipulation and reconfiguration before they can be used. The activity has to be configured specifically for the particular application at that moment. These types of activities require students to begin problem-solving immediately in working through the challenges. Students and teachers cannot skip the modification step. This type of challenge increases the difficulty and at the same time provides for more engagement and student ownership (Grizioti, & Kynigos, 2018; Yiannoutsou, Kynigos, & Daskolia, 2014).

Giving students a problem to solve increases their engagement over traditional projects and assignments (Grizioti, & Kynigos, 2018). In a typical project-based learning scenario, teachers will give the students a test to solve such as designing a new park layout for the city. These types of projects lead to more engagement for students over traditional lectures, but that effect has some limits. In giving students a problem to solve on top of the project, teachers are able to create more long-lasting learning opportunities. Instead of designing a new park layout for the city, students could be challenged to convert a landfill into a park that is sustainable and environmentally protected. Then they would need to convince the city council and voters to approve the plan.

Vallance and Towndrow (2016) further support the concept of challenging problem-solving projects. Actively doing a project brings the activity closer to the thinking processes and improve learning opportunities. The loosely defined challenges require students to engage more critical thinking processes and more problem-solving skills as they work through the project.

This type of work does not allow for simple research and duplication of previous work. The nature of the challenges are unique and requires extensive creative work to defend the solution.

Djambong and Freiman (2016) caution that there are some apparent correlations between the difficulty of the computational thinking task and its successful completion. As the tasks become more difficult, the rate of success for students decreases. This is important to remember as teachers are developing projects for students. If the problems are too difficult for the students to eventually be successful, then they project will be stopped out of frustration. Students should be challenged and should be pushed through a difficult task. At the same time, students should not be given one that is so overwhelming that it appears impossible.

There are several considerations for teachers to keep in mind as they develop challenges that are appropriate for their learners. Teachers should gradually build up the difficulty of the activities as the students improve their results. Students can build on early gains and then grow confidence in tackling larger problems. Scaffolding is also important for learners as the activities get more difficult and the challenges become more open ended. Not all learners adapt to open-ended challenges as readily as others. The scaffolding will help improve results and develop confidence in problem solving abilities. Computational thinking lessons need to be developed to span multiple days and units. Students do not develop strong problem-solving abilities in a single setting. It is through the repeated use of the skills that the abilities become better ingrained (Grizioti, & Kynigos, 2018; Jeon, Kim, Lee, and Kim, 2016).

Specific learning targets need to be intentionally placed within the scope of the computational thinking activity. Teachers need to be acutely aware of the intended outcomes they desire from the activity to ensure that the students achieve those learning targets (Yiannoutsou, Kynigos, & Daskolia, 2014). As teachers are developing activities and creating assessments, they need to be aware that computational thinking activities do not teach all coding concepts equally. Some concepts such as loops are reinforced more frequently than others (Grizioti & Kynigos, 2018). At the end of the unit students must engage in intentional practice and meaningful reflection for the learning in the computational thinking activity to be realized. Students need to be purposeful in understanding what they learned so that the desired concepts are not overshadowed by the challenge itself (Vallance & Towndrow, 2016).

Lessons Learned from other Industries

Case studies of innovation projects in industries other than education can be useful as well. Forcadell and Guadamillas (2002) and Kotter (1995) both show the role that leaders play in the success of the project. The study by Forcadell and Guadamillas (2002) shows that a clear vision by the leaders will facilitate any implementation strategy or adjustment. The vision that the leader sets out is and carried on by the rest of the team. Everyone on the team knowing that they're working towards and having the same goal shows to have a significant impact on the success of the project. Kotter continues by saying that clearly communicating a clear vision and then removing obstacles to their division is he pulling critical to success. Leaders need to take all steps available to support the project.

As organizational leaders begin planning for the implementation, Du Preez and Katz (2007) show that leaders do not need to wait until 100% of the organization is been trained before beginning the project. Work can begin with the part of the organization that has been trained and still get positive results. Learning by smaller teams is internalized and then will spread to the rest of the organization. As initial teams work on the project, teams not yet trained can see the progress that is happening and then achieve quicker buy-in. Having a flat organizational structure will facilitate and speed up of implementation of the project. The reduction in bureaucratic systems helps to move information quicker, allow teams a greater sense of ownership of the changes and then make the changes become longer-lasting (Forcadell & Guadamillas, 2002). To make the effects of the changes more readily noticeable by the teams that are participating and the teams that are yet to be trained, leaders need to develop systems that allow for short-term goals and gains to be clearly visible (Kotter, 1995).

Organizational leaders need to create a core team to help start and implement the change process (Kotter, 1995). This core team will help spread the message by multiplying the number of people communicating the vision. It is important to have a collaborative effort in spreading the vision and training to the rest of the staff. It is also important to have a single point of project management. The single point of contact will allow organizational leaders and staff leaders to provide clear communication channels. In the process, questions and issues that arise can be quickly dealt with. Throughout all of the change process it is important that administration show that they are supportive of the work being done and they are fully committed to the change efforts (Du Preez and Katz, 2007).

How an organization begins the implementation of and innovation strategy and the point at which they stop focusing on it both have an impact on the overall success of the project. Not impressing a great innocence of urgency at the start will likely result in most people not working hard at the transformation. They will not see the need for this to happen over the day-to-day activities they were already engaged in. As the transformation effort becomes more difficult, they will cease to work at it and cause roadblocks for its future success. Likewise, stopping too soon is a situation to avoid. The momentum will be stopped before the work has taken hold and can cause all of the efforts to be completely undone. Too often leaders see the success starting to take place and take off pressure to continue the implementation strategies. Both of these issues will make the changes achieved to be temporary. The goal of any transformation strategy is to make the change is part of the culture and then keep it a sustained part of the organization (Kotter, 1995).

Conclusion

Several key areas have been identified that will help lead to a more successful implementation of an innovation project. The first area of focus is that of teacher preparation. Prior to beginning the project teachers must be adequately prepared to teach both the content subject and computational thinking. Teachers must have a deeper understanding of computational thinking and how to assess the open-ended nature of the problem-solving process. The failure to address these areas will lead to a reduction of computational thinking instruction in the classroom and it eventually be eliminated from the curriculum.

The second area of focus is on student preparation. Student activities must be appropriately scaffold and differentiated to allow for learners to not be overwhelmed but be sufficiently challenged to learn new problem-solving methods. Student activities must be designed to ensure engagement in the process. Developing problem-solving skills through computational thinking does not come from studying the steps but instead the practical application of the steps in solving ever-increasing challenges.

The third and final area of focus is that of leadership and the organization. The organizational leadership must set out a clear vision for the plan that is clearly communicated to the entire team. In conjunction with this vision, trainers, facilitators, along with key staff need to be adequately prepared and given the proper support to lead the rest of the staff through implementation and development of the project. Organizational leaders need to account for all members of the organization can address the needs of each of them.

Integrating computational thinking into the core curriculum is an achievable but difficult task. The key findings presented here can help to ensure that these changes do take place and become a more sustained part of the curricular culture.

References

Cai, J., Yan, H. H., Gong, D., MacLeod, J., & Jin, Y. (2018). A case study to promote computational thinking: The lab rotation approach. *International Conference on Blended Learning: Conference Proceedings*, 393-403. https://doi.org/10.1007/978-3-319-94505-7_32

Davies, R., & Harty, C. (2013, March). Implementing 'site bim': A case study of ict innovation on a large hospital project. *Automation in Construction*, 30, 15-24. https://doi.org/10.1016/j.autcon.2012.11.024

Djambong, T., & Freiman, V. (2016, Oct). Task-based assessment of students' computational thinking skills developed through visual programming or tangible coding environments. *13th International Conference on Cognition and Exploratory Learning in Digital Age: Conference Proceedings*, 41-51. Retrieved from https://files.eric.ed.gov/fulltext/ED571389.pdf

Dolgopolovas, V., Jevsikova, T., Dagiene, V., Savulioniene, L. (2016). Exploration of computational thinking of software engineering novice students based on solving computer science tasks. *In International Journal of Engineering Education*, 32(3(A)), 1-10.

Du Preez, N., & Katz, B. R. (2007). Case study: The implementation of a radical innovation project. *International Conference on Competitive Manufacturing Conference Proceedings*. Retrieved from http://www.indutech.co.za/attachments/135_Case%20study_The%20Implementation%20of%20a%20Radical%20Innovation%20Project.pdf

Dushku, S. (1998). ELT in Albania: Project evaluation and change. *System*, 26, 369–388.

Forcadell, F. J., & Guadamillas, F. (2002, July). A case study on the implementation of a knowledge management strategy oriented to innovation. Knowledge and Process Management, 9(3), 162-171. Retrieved from https://www.researchgate.net/publication/229608830_A_case_study_on_the_implementation_of_a_knowledge_management_strategy_oriented_to_innovation

Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). Computational thinking in mathematics teacher education. *Contemporary Issues in Technology and Teacher Education*, 17(4), 458-477. Retrieved from https://www.citejournal.org/volume-17/issue-4-17/mathematics/computational-thinking-in-mathematics-teacher-education

Grizioti, M., & Kynigos, C. (2018, August). Constructionist approaches to computational thinking: A case of game modding with choico. *Constructivism 2018: Conference Proceedings*. Retrieved from https://www.academia.edu/37351229/Constructionist_Approaches_to_Computational_Thinking_A_Case_of_Game_Modding_with_ChoiCo

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015, March). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers and Education*, 82, 263-279. https://doi.org/10.1016/j.compedu.2014.11.022

Jeon, Y., Kim, S., Lee, S., & Kim, T. (2016, July). A case study on the application of the computational thinking-based creative problem solving (ct-cps) instructional model. *Proceedings of EDULEARN16 Conference*, 8713-8720. Retrieved from http://happyclass.net/home_pdf/edulearn16.pdf

Kalelioglu, F., Gulbahar, Y., & Kukul, V. (2016, May). A framework for computational thinking based on a systematic research review. *Modern Computing*, 4(3), 583-596. Retrieved from https://www.researchgate.net/publication/303943002_A_Framework_for_Computational_Thinking_Based_on_a_Systematic_Research_Review

Kirkgoz, Y. (2008, Oct). A case study of teachers' implementation of curriculum innovation in english language teaching in turkish primary education. *Teaching and Teacher Education: An International Journal of Research and Studies*, 24(7), 1859-1875. http://dx.doi.org/10.1016/j.tate.2008.02.007

Kotter, J. P. (1995, March-April). Leading change: Why early transformation efforts fail. *Harvard Business Review*, 59-67.

Papadakis, S. J., Kalogiannakis, M., & Zaranis, N. (2016, July). Developing fundamental programming concepts and computational thinking with scratchjr in preschool education: A case study. *International Journal of Mobile Learning and Organisation*, 10(3), 187-202. Retrieved from https://www.researchgate.net/profile/Stamatios_Papadakis/publication/305390965_Developing_fundamental_programming_concepts_and_computational_thinking_with_ScratchJr_in_preschool_education_A_case_study/links/

Vallance, M., & Towndrow, P. A. (2016, May). Pedagogic transformation, student-directed design and computational thinking. *Pedagogies: An International Journal*, 11(3), 218-234. https://doi.org/10.1080/1554480X.2016.1182437

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728.

Wedell, M. (2003). Giving TESOL change a chance: Supporting key players in the curriculum change process. *System*, 31, 439–456.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J.T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1-16.

Yiannoutsou, N., Kynigos, C., & Daskolia, M. (2014, January). Constructionist designs in game modding: The case of learning about sustainability. Constructionism 2014: *Constructionism and Creativity: Conference Proceedings*, 1-10. Retrieved from https://www.researchgate.net/publication/265556884_Constructionist_Designs_in_Game_Modding_The_case_of_Learning_about_Sustainability