Journal of Universal Computer Science, vol. 22, no. 12 (2016), 1558-1574
submitted: 23/3/16, accepted: 21/11/16, appeared: 1/12/16 © J.UCS

Software Engineering Education and Games: A Systematic Literature Review

Mehmet Kosa
(Middle East Technical University, Ankara, Turkey
kosa.mehmet@metu.edu.tr)

Murat Yilmaz
(Cankaya University, Ankara, Turkey
myilmaz@cankaya.edu.tr)

Rory V. O'Connor
(Dublin City University, Ireland
roconnor@computing.dcu.ie)

Paul M. Clarke
(Dublin City University, Ireland
paul.m.clarke@dcu.ie)

Abstract: The trend in using games in elementary level education also spreads through higher education levels and specific domains such as engineering. Recently, researchers have shown an increased interest in the usage of games in software engineering. In this paper, we are presenting a systematic review and analysis of 350 papers regarding games in software engineering education that was published in the last fifteen years. After applying our inclusion criteria and manual inspection of these studies, we have ended up with 53 primary papers. Based on a systematic process, we reported and discussed our findings with possible future research directions. The main results of this study indicate that the studies are accumulated around 5 categories: Games that learners/students play, games that learners/students develop as projects, curriculum proposals, developing/coming up with new approaches, tools, frameworks or suggestions and others.

Keywords: Systematic Review, Games, Software Engineering
Categories: D.2, K.3.1, K.3.2

1 Introduction

Software engineering is a discipline that addresses valuable solutions for complex problems, which requires a good combination of theoretical and practical information [Sommerville, 04]. However, typical software engineering education with theoretical formation could not be able to highlight potential practical problems that a novice software engineer should have to deal directly with. Games can be considered as a tool to train novice software practitioners and may address such problems that might cause by a lack of practical information [McGonigal, 11].
With the increased usage of games in many domains including education and the related research in academia, software engineering education also took its part in this

trend. The motivational and other charming aspects of games gain importance in software engineering education [Kurkovsky, 09] [Aydan, 16]. Also, it is stated that software engineers need to work in a highly social and collaborative environment [Kosa, 15] and games are good at providing the collaboration needed while being played and also while being designed [Zagal, 06] [Brandt, 04]. Interestingly, agile development, for instance, was likened to a cooperative game itself [Cockburn, 07]. Therefore games appear to be an appropriate tool to be used in software engineering education.

Games can be used in creative ways to enhance learning [Gee, 05] and provide motivation to the learners/students in-class [Nemerow, 96]. Because of the affordance of creativity that games supply to the developers, it is intriguing to research how they are being utilized in different settings by different software engineering educators. There are many sub-topics in software engineering curriculum that games can be used such as requirements engineering, process modeling, software management, software testing, risk management and so on. Simulating a real-world situation may provide students with the experience they are lacking which also helps unifying the practical understanding with the theoretical knowledge. Also, different approaches (digital/non-digital) may facilitate different learning environments that are worth studying. Therefore, the goal of this study is to conduct a systematic review, and synthesize the available evidence regarding the usage of games in software engineering education.

The remainder of the article is structured as follows: Section 2 gives the methodology of the review with the research questions, inclusion criteria, data sources and search strategy. Then the results are described with primary findings in Section 3. In Section 4, the results are discussed and lastly at Section 5, the study is concluded and future directions for research are briefly provided.

2 Review Method

As briefly mentioned above, the purpose of this study is to review and characterize the state of the art of the utilization of games in software engineering education. To carry out this review, we followed the recommendations in [Kitchenham, 04] [Keele, 07] and [Petersen, 08]. In this section, the planning of the review is explained and the details of the review method are explained below. The details include the research questions, inclusion criteria, data sources and search strategy.

To get an explicit view of the current trend in game-based software engineering education, this literature review is conducted with the following specific objectives in mind.

The objectives of our work are multifold. First, we would like to understand the nature and types of games that are used in software engineering education. Second, we would like to investigate and find out what benefits are obtained from such games if any. Third, we would like to observe the studies to find if any methodologies to develop games for software engineering education exist and if they are empirically validated. Finally, we would like to see if there is a growing interest in the field or not. Overall, we would like to highlight the nature of research results in the literature.

2.1 Research Questions

We translate our research objectives into specific research questions as follows:
(RQ1) With what kinds of purposes are games used in software engineering education? Are they mostly utilized in the sense that students play games or students develop games in the learning process?
(RQ2) What kinds of games are used in software engineering education? Are they digital or non-digital?
(RQ3) How positive is the outcome of the usage of games?
(RQ4) Do the studies being carried out provide empirical results?
(RQ5) Are there any design guidelines specific to the software engineering domain?
(RQ6) What is the distribution of studies being conducted according to the years? Is there an increasing/decreasing trend?

2.2 Inclusion Criteria

Studies that are either empirical or not, either composed of digital games or not, written either by academics or otherwise are included starting from 2000. The review did not differentiate between qualitative or quantitative studies and included workshop papers as well. Papers that are hit by the keyword search that are about coding and programming are included if it claims that the study is in for software engineering education. Only the studies that are written in English are considered. Other literature reviews on the subject are excluded since it is thought that they would not be directly contributing to the answers for the research questions described above. We have subjectively included studies which are at the intersection of games, software engineering and education in a very broad sense. Our evaluation was non-stringent, however studies that are about games and software engineering but not education were omitted, for instance.

Inclusion Criteria	• Papers come up from the search string • Journal, conference or workshop papers • Papers written in English • Published in or after 2000 • Studies that add value to software engineering education
Exclusion Criteria	• Papers that do not focus on software engineering education • Other literature reviews • PowerPoint presentations • Blogs or websites

Table 1: Summary of the inclusion/exclusion criteria

2.3 Data Sources

2.3.1 Digital Databases

The search has been performed on electronic databases which are:

- ACM Digital Library
- IEEE Xplore Digital Library
- ScienceDirect
- SpringerLink
- Wiley Online Library

We tried to keep the keyword scope narrow since if one fails to determine the scope appropriately, game-based learning may bring loads of studies not necessarily reside in the software engineering scope. A couple of initial informal searches have been conducted to determine the best keyword combination. "Video game" keyword is avoided for the non-digital games to be also included in the study. Other than that, "computer science" or "information systems" keywords were not chosen not to stray too far off and to be able to remain in the exact domain of software engineering. The keywords for the search have been selected as: "software engineering education" and "games". They are searched in titles, abstracts and keywords.

2.3.2 Games and Software Engineering (GAS) Workshop

Other than searching for studies in the digital databases, papers published in "Games and Software Engineering Workshop" have been skimmed since it is obviously relevant to the topic at hand. The papers found from the workshop were generally from the "Games and software engineering education" track. For 4 years (2011, 2012, 2013 and 2015 since the workshop was not held in 2014), there were a total of 43 papers where 3 of them were selected to be included in this review study. Other than these papers, relevantly found 5 papers were identified to be duplicates of the digital database search, which are then ignored. The breakdown of numbers of studies according to the years is below:

- Games and Software Engineering (GAS), 2011 1st International Workshop – 14
- Games and Software Engineering (GAS), 2012 2nd International Workshop – 9
- Games and Software Engineering (GAS), 2013 3rd International Workshop – 9
- Games and Software Engineering (GAS), 2015 4rd International Workshop – 11

2.4 Search Strategy

The search was carried out for the studies since 2000. Selected studies are listed in Appendix A.
Firstly, the systematic review process was started with the search for the studies by using the previously defined keywords ("software engineering education" and "games"). By manual abstract inspection, the irrelevant studies were then excluded. After that, the duplicates were removed from the study. Then the studies were further eliminated by inspecting the content of the paper. Lastly, the remaining studies were classified according to their purpose and scope. The search strategy is summarized in Figure 1.

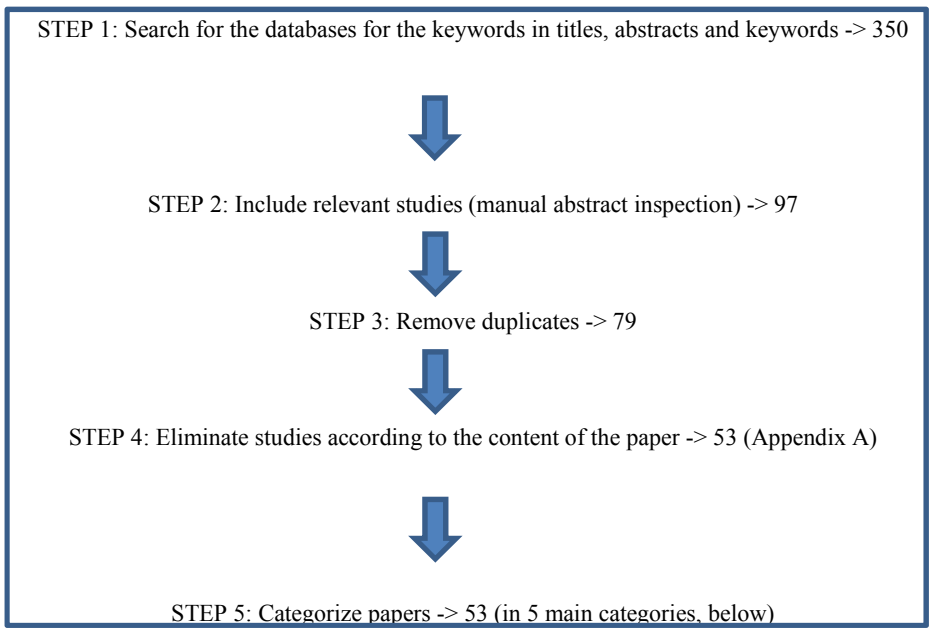

Figure 1: The Search Procedure and Number of Papers Included in the Review

STEP 1: In this step, all the papers written in English that contain "software engineering education" and "games" in their titles and/or abstracts and/or keywords were included. Data sources were selected as several respected digital databases and "Games and Software Engineering Workshop" as mentioned above.

STEP 2: After gathering all the relevant papers according to the search keywords, abstracts of those papers were inspected manually and studies outside of the scope of this review were taken out.

STEP 3: Since we have accumulated the studies from multiple resources (digital databases and a specific workshop), there were duplicates and therefore they were omitted and the number of papers included were calculated again.

STEP 4: Manually inspecting the remaining studies showed that several of them were irrelevant and they were eliminated. Some of the reasons for elimination in this step are:

- The short versions of studies that were already in the review as another study (another form of duplication),
- Explanation of addition of a new feature to an already existing approach,
- Studies that are more inclined to focus on other domains rather than software engineering

- Studies that develop a rather generic method and not necessarily software engineering

STEP 5: In this step, having all the pertinent studies ranging between years 2000 to 2015 at hand, according to how games are utilized, they were categorized in to groups as explained in the results section.

3 Results

The overview of the results of the review is given in Appendix B.

Quantitative results achieved after each step can be seen in Table 2. Studies included from digital databases and from workshops are presented separately.

	ACM	IEEE	ScienceDirect	Springer	Wiley	GASs	Total
STEP 1	61	69	12	179	3	43	350
STEP 2	30	37	4	11	4	11	97
STEP 3	29	28	3	11	2	6	79
STEP 4	20	18	3	7	2	3	53

Table 2: Number of papers according to the search. GASs: Games and Software Engineering Workshops

After STEP 4, it has been seen that the remaining studies are accumulated around 5 main categories (Figure 2):

1. Games that learners/students play (23): A game is developed for students to play that helps the learning process of software engineering concepts. Studies in this category explain the game mechanics of their game and how it enhances learning.
2. Games that learners/students develop as projects (11): Games are developed by students where they experience the software development processes. Games are used as the motivating factor for students to develop projects and they are used to expose students to challenging problems that they cannot generally face with non-game projects. Authors of the papers that fall into this category generally present the experience they had while using this method or the challenges and benefits of the method.
3. Curriculum proposals (4): Seemingly similar nevertheless distinct from the second category, authors present curriculums that can be adopted by teachers of project-based courses where students need to develop a software system. Gamification of the curriculum approaches are also examined in this category.
4. Developing/Coming up with new approaches, tools, frameworks or suggestions (8): Rather than describing a game-based course as in category 2, the studies selected for this category posit new approaches, tools, frameworks or suggestions for the acceptance of usage of games in software engineering education.

5. Others (7): Last category was created for the studies that do not fit into any of the above categories and also do not accumulate enough to be in a separate, distinct category. The studies in this category are about acceptance of game-based learning in software engineering education [S2], importance of experience [S12], interest and success of women [S42], effectiveness of gamified approaches [S40], games as research tools [S7], using games in game design [S44] and game programming and evaluation of an already developed game [S29].
[S39] was identified to be fitting into both first and second categories since the idea of the paper is both to make students design games (by senior students) and to play those games (by junior students). Being said that, we have considered this study to be appropriate to fit in the second category for the sake of this review study.

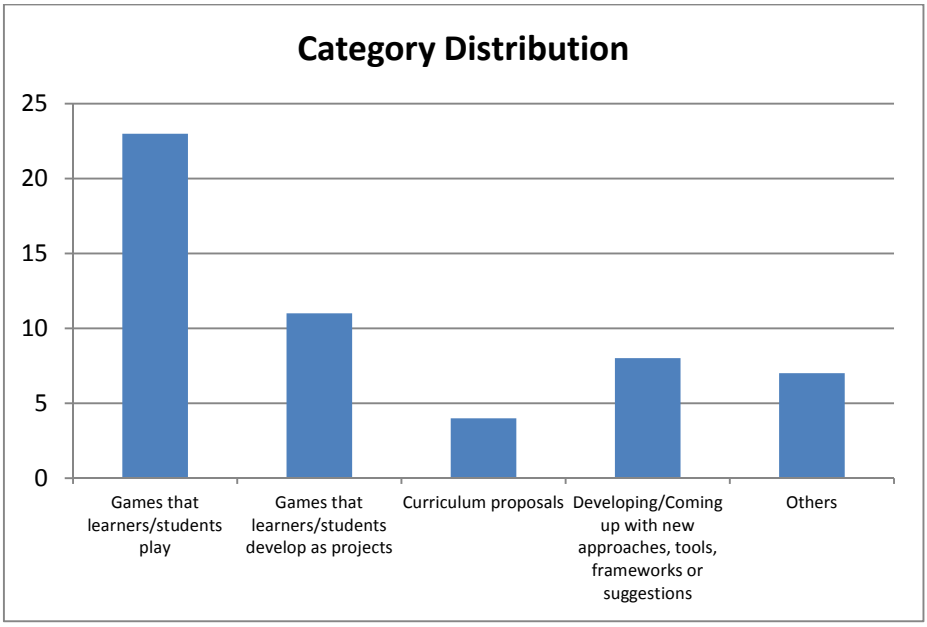

Figure 2: Category Distribution of the Papers Included in the Review

4 Discussion

The most obvious finding to emerge from the analysis is that the studies that are being carried out in the intersection of software engineering education and games are accumulated around 5 main categories. Figure 2 articulates those categories and gives the number of papers included in each category. Returning to the research question (RQ1) posed at the beginning of the study, it is now possible to state that game notion is used both for playing and designing.

With respect to the second research question (RQ2), as can be seen from Figure 3, there is significant amount of study on digital games which may not seem to be surprising in this day and age. Figure 3 outlines the numbers of selected papers which have a digital game approach, non-digital game approach or both. Although utilization from studies from the digital perspective is unquestionable, the capability of non-digital efforts should not be disregarded. Bringing its own unique attributes, non-digital games may provide additional benefit. For instance, educators may utilize them, harnessing its social characteristics (e.g. face-to-face interaction) especially in teaching complex subjects that involve social aspects and that are hard to teach without simulating. Using a hybrid approach, which encompasses both digital and non-digital approaches, may also be an option where necessary.

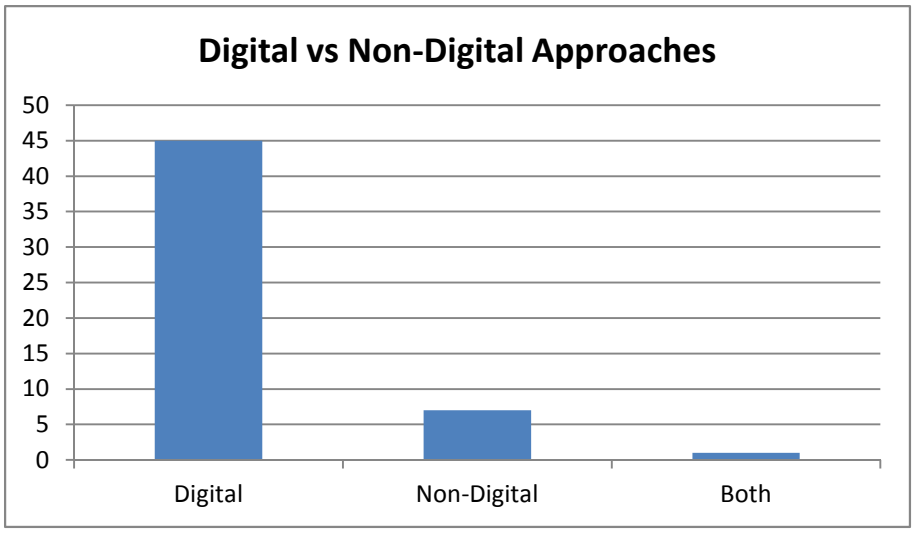

Figure 3: The Numbers of Digital and Non-Digital Approaches

There are studies which provide empirical evidence that the utilized game or method actually enhances learning [S5] [S19] [S39]. Conducting experiments in multiple institutions is a solid validation to show that the developed game or method practically works in different environments [S29]. This approach should be adopted by other game developers (for software engineering education) as well. Also, these kinds of empirical evidence in different settings are essential to support an academic point of view. According to the findings, the third research question (RQ3) can be addressed as the game approach outcomes are mostly reported as positive. As for the forth research question (RQ4), it would be convenient to state that although there are empirical studies in the literature, we can conclude that there needs to be more studies with empirical parts.

For the fifth research question (RQ5) that was posed, it can be claimed that there are not much studies which attempts to develop design guidelines or heuristics for developing games for software engineering education however, one study in this review reports its findings on the efforts for mapping game design elements to the

learning elements for the software management education in particular [S19]. The scarcity of these kinds of studies shows us that there is still space to be discovered on supplying design guidelines for developers.

Returning to the sixth research question (RQ6) that was set forth in the previous sections, Figure 4 shows the number of studies in years. If we were to break them into 3 year intervals (Figure 5) it would be easier to see that there is a somewhat increasing trend in the papers published in the last 15 years on software engineering education and games. This shows also that games are being considered and tested for usage in software engineering education more and more and even further adopted in software engineering classes. In addition to this, we have decided to see the trend in the first category papers which have been defined as the "games that learners/students play" and which is the category that hosts maximum amount of studies in this review. The distribution of the first category papers according to years can be seen in Figure 6. This figure tells us that since 2005, a steady amount of work is being done in category 1. Therefore, if one thinks about the forecast, it can be stated that developers/educators will be designing games and testing them for software engineering education at a steady pace in the upcoming years.

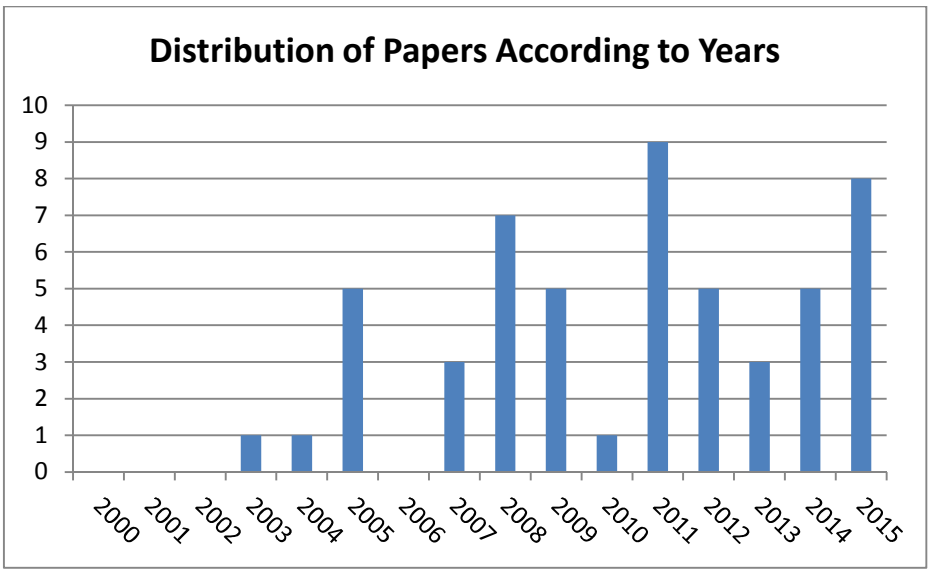

Figure 4: The Numbers of Selected Studies in Years

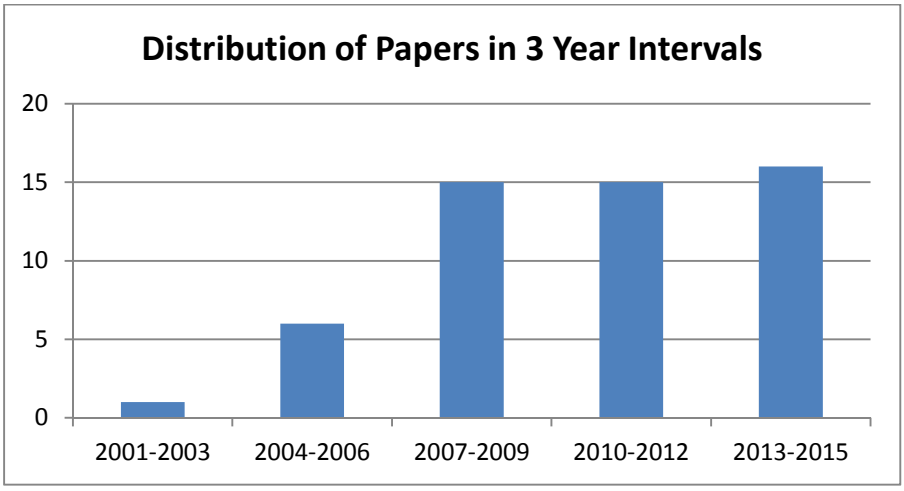

Figure 5: The Numbers of Selected Studies (Category 1) in 3 Year Intervals

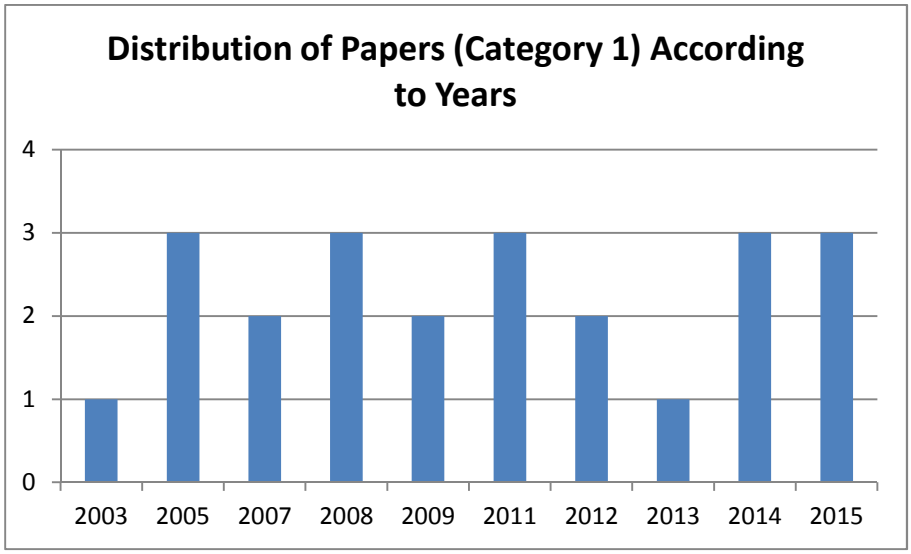

Figure 6: The Numbers of Selected Studies (Category 1) in Years

Other than the answers to the research questions posed, there are some insights that can be deducted from the study which are mentioned below.

Some of the empirical studies in this review suggest that a game-based learning approach enhances positive experiences for participants and ultimately improve the learning levels of students. It has been also reported that the game-based learning approaches are more favored by the software engineering students with respect to traditional teaching methods. In addition, when we have a look at the categories 1 and

2 (where they constitute most of the studies of the review), games are being utilized because of their instant feedback mechanisms and their ability to create/improve motivation (e.g. [S6], [S8]). We argue that when students are playing games or designing games, their actions are mostly voluntary. In general, this review showed that games provide extra motivation and positive experiences which both creates an entering point to the learning process for unmotivated individuals and adds value to the learning process for already motivated ones making it more effective.

If the subject that needs to be taught is a traditional software process model (e.g. waterfall), game project-based courses -where students develop games- may not be the best choice since game projects may not necessarily use sequential design approaches in real life because of its evolving nature with every playtest.

Another observation of this review is that, there are no solid longitudinal studies in the area which shows that learning is more sustainable with game-based approaches with respect to traditional methods. Some effort should also be put from this point of view to show if games provide more lasting learning.

The main limitation of this review study might be that since the keywords "computer science", "learn", "teach", "simulation" and alike are not used in the first place, some of the studies may still be excluded although they are relevant to the software engineering education domain. Despite there might be relevant studies that are not included in this review, we believe that the categorization that emerged at the end of the study would not have been significantly different in the sense that the initial number of papers included in this review is acceptable for generalization.

5 Conclusions and Future Work

In conclusion, this review study shows that software engineering education and games are being approached from several angles that are categorized above. Although there is an increasing trend in the number of studies, there are no clear design guidelines set for the design and evaluation of games in this particular domain. Also, studies on non-digital in-class approaches lack empirical evidence which might be a good starting point for the new researchers in the area. For instance, development of non-digital games by the students as a learning intensifier after they have educated by traditional methods, was not reported in any of the studies which might be a novel approach. None of the studies should propose to change the traditional education to a whole new game-based teaching at this level, however a smooth transition into a more game-like environment blending with conventional methods seems to be the prevalent strategy as of now. Investment in the software engineering education domain will greatly affect the future software engineers which in result will be helping to achieve the overarching goal of software process improvement.

References

[Aydan, 16] Aydan, U., Yilmaz, M., Clarke, P. M., & O'Connor, R. V.: Teaching ISO/IEC 12207 Software Lifecycle Processes: A Serious Game Approach. Computer Standards & Interfaces. Computer Standards & Interfaces, 2016.

[Brandt, 04] Brandt, E., & Messeter, J.: Facilitating collaboration through design games. In Proceedings of the eighth conference on Participatory design: Artful integration: interweaving media, materials and practices-Volume 1 (pp. 121-131). ACM, July 2004.

[Cockburn, 07] Cockburn, A.: Agile Software Development: The Cooperative Game. Addison-Wesley, 2007.

[Gee, 05] Gee, J. P.: Learning by design: Good video games as learning machines. E-Learning and Digital Media, 2(1), 5-16, 2005.

[Keele, 07] Keele, S.: Guidelines for performing systematic literature reviews in software engineering. In Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[Kitchenham, 04] Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University, 33(2004), 1-26, 2004.

[Kosa, 15] Kosa, M., & Yilmaz, M.: Designing Games for Improving the Software Development Process. In Systems, Software and Services Process Improvement (pp. 303-310). Springer International Publishing, 2015.

[Kurkovsky, 09] Kurkovsky, S.: Engaging students through mobile game development. In ACM SIGCSE Bulletin (Vol. 41, No. 1, pp. 44-48). ACM, March 2009.

[McGonigal, 11] McGonigal, J.: Reality is broken: Why games make us better and how they can change the world. Penguin, 2011.

[Nemerow, 96] Nemerow, L. G.: Do Classroom Games Improve Motivation and Learning?. Teaching and Change, 3(4), 356-66, 1996.

[Petersen, 08] Petersen, K., Feldt, R., Mujtaba, S., & Mattsson, M.: Systematic mapping studies in software engineering. In 12th international conference on evaluation and assessment in software engineering (Vol. 17, No. 1, pp. 1-10). Sn, 2008.

[Sommerville, 04] Sommerville, I.: Software Engineering. Pearson, 2004.

[Zagal, 06] Zagal, J. P., Rick, J., & Hsi, I.: Collaborative games: Lessons learned from board games. Simulation & Gaming, 37(1), 24-40, 2006.

Appendix A

[S1] Ahmadi, N., Jazayeri, M., & Repenning, A. (2011, May). Towards democratizing computer science education through social game design. In Proceedings of the 1st International Workshop on Games and Software Engineering (pp. 48-51). ACM.

[S2] Albayrak, O. (2015, May). Instructor's acceptance of games utilization in undergraduate software engineering education: a pilot study in Turkey. In Games and Software Engineering (GAS), 2015 IEEE/ACM 4th International Workshop on (pp. 43-49). IEEE.

[S3] Baker, A., Navarro, E. O., & Van Der Hoek, A. (2003, May). Problems and programmers: an educational software engineering card game. In Software Engineering, 2003. Proceedings. 25th International Conference on (pp. 614-619). IEEE.

[S4] Baker, A., Navarro, E. O., & Van Der Hoek, A. (2005). An experimental card game for teaching software engineering processes. Journal of Systems and Software, 75(1), 3-16.

[S5] Bollin, A., Hochmuller, E., & Mittermeir, R. T. (2011). Teaching software project management using simulations.

[S6] Cagiltay, N. E. (2007). Teaching software engineering by means of computer-game development: Challenges and opportunities. British Journal of Educational Technology, 38(3), 405-415.

[S7] Caulfield, C., & Maj, S. P. (2008). Come Play. In Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education (pp. 86-91). Springer Netherlands.

[S8] Chang, W. C., Chen, Y. L., & Lee, T. P. (2008). Computer assisted learning with card game in system design concept. In Advances in blended learning (pp. 93-101). Springer Berlin Heidelberg.

[S9] Chen, W. F., Wu, W. H., Wang, T. L., & Su, C. H. (2008, October). Work in progress-a game-based learning system for software engineering education. In Frontiers in Education Conference, 2008. FIE 2008. 38th Annual (pp. T2A-12). IEEE.

[S10] Claypool, K., & Claypool, M. (2005). Teaching software engineering through game design. ACM SIGCSE Bulletin, 37(3), 123-127.

[S11] Connolly, T. M., Stansfield, M., & Hainey, T. (2007). Using Games-Based Learning to Teach Software Engineering. In Web Information Systems and Technologies (pp. 304-313). Springer Berlin Heidelberg.

[S12] de Lima, L. G. R., de Lima Salgado, A., & Freire, A. P. (2015, November). Evaluation of the user experience and intrinsic motivation with educational and mainstream digital games. In Proceedings of the Latin American Conference on Human Computer Interaction (p. 11). ACM.

[S13] Eagle, M. J., & Barnes, T. (2012, February). A learning objective focused methodology for the design and evaluation of game-based tutors. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 99-104). ACM.

[S14] Emam, A., & Mostafa, M. G. (2012, July). Using game level design as an applied method for Software Engineering education. In Computer Games (CGAMES), 2012 17th International Conference on (pp. 248-252). IEEE.

[S15] Galvão, T. A. B., Neto, F. M. M., Bonates, M. F., & Campos, M. T. (2012). A serious game for supporting training in risk management through project-based learning. In Virtual and Networked Organizations, Emergent Technologies and Tools (pp. 52-61). Springer Berlin Heidelberg.

[S16] Garneli, V., Giannakos, M. N., Chorianopoulos, K., & Jaccheri, L. (2015, May). Serious game development as a creative learning experience: lessons learnt. In Proceedings of the Fourth International Workshop on Games and Software Engineering (pp. 36-42). IEEE Press.

[S17] Goulding, T. (2008). Complex game development Throughout the college curriculum. ACM SIGCSE Bulletin, 40(4), 68-71.

[S18] Hainey, T., Connolly, T. M., Stansfield, M., & Boyle, E. A. (2011). Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level. Computers & Education, 56(1), 21-35.

[S19] Letra, P., Paiva, A. C. R., & Flores, N. (2015, October). Game Design Techniques for Software Engineering Management Education. In Computational Science and Engineering (CSE), 2015 IEEE 18th International Conference on (pp. 192-199). IEEE.

[S20] Linhoff, J., & Settle, A. (2008, June). Teaching game programming using XNA. In ACM SIGCSE Bulletin (Vol. 40, No. 3, pp. 250-254). ACM.

[S21] Lino, J. E. N., Paludo, M. A., Binder, F. V., Reinehr, S., & Malucelli, A. (2015, October). Project management game 2D (PMG-2D): A serious game to assist software project managers training. In Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE (pp. 1-8). IEEE.

[S22] Liu, C., & Wu, B. (2011). Enabling collaborative learning with an educational MMORPG. In 2011 IEEE International Games Innovation Conference (IGIC).

[S23] Ludi, S. (2011, May). The benefits and challenges of using educational game projects in an undergraduate software engineering course. In Proceedings of the 1st International Workshop on Games and Software Engineering (pp. 13-16). ACM.

[S24] Lund, H. H., & Pagliarini, L. (2011, June). An educational tool for creating distributed physical games. In Proceedings of the 6th International Conference on Foundations of Digital Games (pp. 219-226). ACM.

[S25] Kiniry, J. R., & Zimmerman, D. M. (2011, May). Verified gaming. In Proceedings of the 1st International Workshop on Games and Software Engineering (pp. 17-20). ACM.

[S26] Marques, B. R., Levitt, S. P., & Nixon, K. J. (2012, October). Software visualisation through video games. In Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference (pp. 206-215). ACM.

[S27] Naoya, N., Takemura, Y., & Kume, I. (2009, October). A practice of collaborative project-based learning for mutual edification between programming skill and artistic craftsmanship. In Frontiers in Education Conference, 2009. FIE'09. 39th IEEE (pp. 1-5). IEEE.

[S28] Nassal, A. (2014, June). A general framework for software project management simulation games. In Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on (pp. 1-5). IEEE.

[S29] Navarro, E., & Van Der Hoek, A. (2009, March). Multi-site evaluation of SimSE. In ACM SIGCSE Bulletin (Vol. 41, No. 1, pp. 326-330). ACM.

[S30] Navarro, E. O., & Van Der Hoek, A. (2005). Software process modeling for an educational software engineering simulation game. Software Process: Improvement and Practice, 10(3), 311-325.

[S31] Oh Navarro, E., & Van Der Hoek, A. (2005, April). Design and evaluation of an educational software process simulation environment and associated model. In Software Engineering Education &# x00026; Training, 18th Conference on (pp. 25-32). IEEE.

[S32] Peixoto, D. C., Prates, R. O., & Resende, R. F. (2010, March). Semiotic inspection method in the context of educational simulation games. In Proceedings of the 2010 ACM Symposium on Applied Computing (pp. 1207-1212). ACM.

[S33] Pieper, J. (2012, June). Learning software engineering processes through playing games: suggestions for next generation of simulations and digital learning games. In Proceedings of the Second International Workshop on Games and Software Engineering: Realizing User Engagement with Game Engineering Techniques (pp. 1-4). IEEE Press.

[S34] Pieper, J. (2015, March). Discovering the essence of Software Engineering an integrated game-based approach based on the SEMAT Essence specification. In Global Engineering Education Conference (EDUCON), 2015 IEEE (pp. 939-947). IEEE.

[S35] Rong, G., Zhang, H., & Shao, D. (2014, April). Where does experience matter in software process education? An experience report. In Software Engineering Education and Training (CSEE&T), 2014 IEEE 27th Conference on (pp. 129-138). IEEE.

[S36] Rusu, A., Russell, R., Burns, E., & Fabian, A. (2011). Employing software maintenance techniques via a tower-defense serious computer game. In Edutainment Technologies. Educational Games and Virtual Reality/Augmented Reality Applications (pp. 176-184). Springer Berlin Heidelberg.

[S37] Ryoo, J., Fonseca, F., & Janzen, D. S. (2008, April). Teaching object-oriented software engineering through problem-based learning in the context of game design.

In Software Engineering Education and Training, 2008. CSEET'08. IEEE 21st Conference on (pp. 137-144). IEEE.

[S38] Schaefer, S., & Warren, J. (2004). Teaching computer game design and construction. Computer-Aided Design, 36(14), 1501-1510.

[S39] Shabalina, O., Sadovnikova, N., & Kravets, A. (2013, August). Methodology of teaching software engineering: game-based learning cycle. In Engineering of Computer Based Systems (ECBS-EERC), 2013 3rd Eastern European Regional Conference on the (pp. 113-119). IEEE.

[S40] Su, C. H. (2015). The effects of students' motivation, cognitive load and learning anxiety in gamification software engineering education: a structural equation modeling study. Multimedia Tools and Applications, 1-24.

[S41] Sweedyk, E., & Keller, R. M. (2005, June). Fun and games: a new software engineering course. In ACM SIGCSE Bulletin (Vol. 37, No. 3, pp. 138-142). ACM.

[S42] Sweedyk, E. (2011, March). Women build games, seriously. In Proceedings of the 42nd ACM technical symposium on Computer science education (pp. 171-176). ACM.

[S43] Taran, G. (2007, July). Using games in software engineering education to teach risk management. In Software Engineering Education & Training, 2007. CSEET'07. 20th Conference on (pp. 211-220). IEEE.

[S44] Thompson, R. H., & Tanimoto, S. L. (2013, May). Game design as a game. In Proceedings of the 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change (pp. 27-31). IEEE Press.

[S45] Tillmann, N., De Halleux, J., Xie, T., & Bishop, J. (2014, July). Constructing coding duels in Pex4Fun and code hunt. In Proceedings of the 2014 International Symposium on Software Testing and Analysis (pp. 445-448). ACM.

[S46] Tillmann, N., De Halleux, J., Xie, T., & Bishop, J. (2014, March). Code Hunt: Gamifying teaching and learning of computer science at scale. In Proceedings of the first ACM conference on Learning@ scale conference (pp. 221-222). ACM.

[S47] Von Wangenheim, C. G., Thiry, M., & Kochanski, D. (2009). Empirical evaluation of an educational game on software measurement. Empirical Software Engineering, 14(4), 418-452.

[S48] Uskov, V., & Sekar, B. (2014, October). Gamification of software engineering curriculum. In Frontiers in Education Conference (FIE), 2014 IEEE (pp. 1-8). IEEE.

[S49] Wang, A. I., & Wu, B. (2009). An application of a game development framework in higher education. International Journal of Computer Games Technology, 2009, 6.

[S50] Wang, T., & Zhu, Q. (2009, March). A software engineering education game in a 3-D online virtual environment. In Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on (Vol. 2, pp. 708-710). IEEE.

[S51] Xie, T., Bishop, J., Tillmann, N., & de Halleux, J. (2015, April). Gamifying software security education and training via secure coding duels in code hunt. In Proceedings of the 2015 Symposium and Bootcamp on the Science of Security (p. 26). ACM.

[S52] Xie, T., Tillmann, N., & De Halleux, J. (2013, May). Educational software engineering: Where software engineering, education, and gaming meet. In Proceedings of the 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change (pp. 36-39). IEEE Press.

[S53] Ye, E., Liu, C., & Polack-Wahl, J. A. (2007, October). Enhancing software engineering education using teaching aids in 3-D online virtual worlds. In Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE'07. 37th Annual (pp. T1E-8). IEEE.

Appendix B

Study	Category	Year	Games Used/Developed
[S1]	4	2011	Digital
[S2]	5	2015	Digital
[S3]	1	2003	Non-digital
[S4]	1	2005	Non-digital
[S5]	4	2011	Digital
[S6]	2	2007	Digital
[S7]	5	2008	Non-digital
[S8]	1	2008	Non-digital
[S9]	1	2008	Digital
[S10]	3	2005	Digital
[S11]	1	2008	Digital
[S12]	5	2015	Digital
[S13]	4	2012	Digital
[S14]	2	2012	Digital
[S15]	1	2012	Digital
[S16]	2	2015	Digital
[S17]	3	2008	Digital
[S18]	1	2011	Digital
[S19]	4	2015	Digital
[S20]	2	2008	Digital
[S21]	1	2015	Digital
[S22]	1	2011	Digital
[S23]	2	2011	Digital
[S24]	4	2011	Both
[S25]	2	2011	Digital
[S26]	1	2012	Digital
[S27]	2	2009	Digital
[S28]	4	2014	Digital
[S29]	5	2009	Digital
[S30]	1	2005	Digital

[S31]	1	2005	Digital
[S32]	4	2010	Digital
[S33]	4	2012	Digital
[S34]	1	2015	Digital
[S35]	1	2014	Non-digital
[S36]	1	2011	Digital
[S37]	3	2008	Digital
[S38]	2	2004	Digital
[S39]	2	2013	Digital
[S40]	5	2015	Digital
[S41]	2	2005	Digital
[S42]	5	2011	Digital
[S43]	2	2007	Non-digital
[S44]	5	2013	Digital
[S45]	1	2014	Digital
[S46]	1	2014	Digital
[S47]	1	2009	Digital
[S48]	3	2014	Non-digital
[S49]	2	2009	Digital
[S50]	1	2009	Digital
[S51]	1	2015	Digital
[S52]	1	2013	Digital
[S53]	1	2007	Digital