Journal of Educational Technology & Online Learning
Volume 4 | Issue 1 | 2021
http://dergipark.org.tr/jetol

Competition-based learning of blockchain programming

Davut Çulha [a*]

[a]Aselsan, Turkey,

Suggested citation: Çulha, D. (2021). Competition-based learning of blockchain programming. *Journal of Educational Technology & Online Learning, 4*(1), 46-55.

Article Info

Keywords:

blockchain programming
smart contract development
competition-based learning
blockchain development

Research Article

Abstract

Blockchain, which is a disruptive technology, affects many technologies, and it will affect many other technologies. Main property of blockchain technologies is assuring trust without central authorization. This is achieved through immutable data and decentralization. Moreover, blockchain is founded on the principles of cryptography, which provides the required infrastructure for the trust. First application of the blockchain technologies is Bitcoin cryptocurrency. After the birth of Bitcoin, cryptocurrencies began to change financial systems. Learning of blockchain is difficult because blockchain and its related technologies are strange for most of the people. In order to figure out blockchain technologies, the concepts like cryptography, cryptocurrency, immutable data and decentralization should have been understood. Therefore, blockchain and its related technologies should be learned through efficient learning mechanisms. Project-based learning, team-based learning, active learning and competition-based learning can be used for efficient teaching of blockchain. Competition-based learning has been used in many areas successfully for years. Smart contract development is the programming part of blockchain technologies. In this paper, competition-based learning is applied to blockchain programming to increase learning efficiency. In addition, a methodology is presented to apply competition-based learning to blockchain programming.

1. Introduction

Blockchain is related to decentralization. Decentralization means that control of a system is distributed among nodes in network. In other words, the system cannot be controlled by central authorities. This property provides trust, security, and consensus to systems. Decentralization is achieved by distributing the data of the system to enough network nodes. The data of this kind of systems is called blockchain. Blockchain is empowered with cryptography to provide its principal properties.

Blockchain technologies were revealed after the birth of Bitcoin cryptocurrency. Bitcoin proved that trust to a system can be achieved without central authorities. In other words, decentralization can provide trust to systems. After ensuring the trust to the blockchain, various applications of decentralization became possible. Therefore, programming entered to the scene. Programs on blockchain were named as smart contracts. Especially, the word contract indicates the trust property of blockchain. Vitalik Buterin developed a general purpose blockchain programming platform Ethereum. Ethereum has also its native cryptocurrency called Ether, which is the required gas to drive smart contracts in addition to its being a cryptocurrency.

Blockchain technologies are spreading fast to new application areas by extending its current circle of applications. However, this fast progress does not go parallel with the understandability of blockchain

* Corresponding author: Davut Çulha, Aselsan, Turkey
e-mail addresses: culha@aselsan.com.tr

Doi: http://doi.org/10.31681/ jetol.831645
Received 26 Nov 2020; Revised 17 Jan 2021; Accepted 19 Jan 2021.
ISSN: 2618-6586 .

technologies. In other words, learning and comprehending these technologies are very difficult. For students to understand blockchain, they should understand also its related technologies. Consequently, efficient learning mechanisms are needed to teach blockchain technologies to students. Competitions can be used to improve the efficiency of learning. In this paper, competition-based learning is applied to blockchain programming to increase the learning performance of blockchain technologies.

Smart contract development is part of software development. However, it needs special treatments because of its characteristics. Therefore, special software engineering methods should be applied. In particular, agile practices are crucial for smart contract development because of divide-and-conquer approaches of agile methodologies. Iterative and incremental approaches of agile methodologies can be included in an efficient learning process.

Competition-based learning is well applied with a project in teams. The project-based learning has also good contribution to learning processes. In addition, teams should be formed so that their members can embrace the power of mutual work of team-based learning.

Also, games can be used to improve learning performance. Especially, tournaments in the games can be included to preserve the willingness of learning of students to a long period of time. Tournaments in games correspond iterations in agile methodologies. Both provide sustainability in works.

Students can be included in the learning process for the efficiency of learning. Students can take active roles during learning. This time, active learning comes in view. Competition-based learning supports also active learning. In order to get good results in competitions, students act actively to learn earlier than others.

The rest of this paper is structured as follows. In the next section, related work is given. Then, the methodology is described. Application of the methodology will come next. Results of the application are handled then. After discussion, the paper will end up with conclusion.

2. Related Work

Blockchain (Swan, 2015) is a shining technology term. Blockchain is simply a database which is immutable and has chains among its data parts. The chains, or simply the links, are empowered by cryptography. This immutable database is distributed in networks so that blockchain is decentralized. In other words, it cannot be controlled by central authorities.

As recognizing the power of blockchain, its applicability to many technology areas becomes clear. Bitcoin which is a cryptocurrency is the first application area of blockchain technologies (Nakamoto, 2008). Cryptocurrencies are virtual money systems, and probably will take the place of current fiat currencies. Smart contracts or blockchain programs form another important application area of blockchain technologies (Buterin, 2013). A smart contract is a distributed program running on blockchain. In future, probably most of the real contracts will be implemented using these distributed blockchain programs. Ethereum (Buterin, 2013) is a blockchain platform where the first Turing-complete execution environment is created for smart contracts. Ethereum presents a virtual machine for execution of smart contracts. The virtual machine is called Ethereum Virtual Machine (EVM). EVM executes assembly level codes of smart contracts. In other words, EVM executes bytecodes of smart contracts which can be generated from higher level blockchain programs. Solidity (Dannen, 2017) is the main high-level programming language of Ethereum platform. Ethereum platform has also its own cryptocurrency named Ether. Smart contracts can only be executed on Ethereum with some Ethers. This method prevents Ethereum blockchain from denial-of-service type attacks.

Smart contract development is difficult for learning because blockchain technologies have not been understood sufficiently. In order to figure out blockchain technologies, people should become familiar with cryptography, cryptocurrency, immutable data, and decentralization. These are new for most of the people so that learning something related to blockchain technologies is difficult. In this paper, competition-based learning (Altin, 2013) is applied to smart contract development to increase learning efficiency.

Smart contract development tasks increase with the increase of applications of blockchain technologies. Therefore, software engineering for smart contract development becomes important. Software engineering methods and practices should be adapted to smart contract development because of unique characteristics of blockchain (Chakraborty et al., 2018). In Marchesi, Marchesi and Tonelli (2018), a software development process is proposed for blockchain development, which is based on agile practices. Agile practices are enhanced for the specific requirements of blockchain development.

Competition-based learning uses the power of competitions in learning. Competitions increase eagerness of students. Competition-based learning (Marchesi, Marchesi & Tonelli, 2018) finds place in many education areas nowadays. It is applied in robotics (Altin, 2013; Jung, 2013). Another natural application area for competition-based learning is autonomous robotic vehicles. In the references (Paulik & Krishnan, 2001; Paulik & Krishnan, 1999; Hyyppä, 2004), it is applied in design courses where are competitions among autonomous robotic vehicles. The competitions are carried out through a project in teams. The project presents some challenges to the students, and motivates them to overcome them to produce a solution. Therefore, this kind of project-based learning also increases the efficiency of learning. Teams are also good ingredients for learning. Members of a team collaborate and do teamwork among themselves.

Game-based learning uses games to achieve learning outcomes (Plass, Homer & Kinzer, 2015). In Burguillo (2010), competition-based learning is combined with other learning methods as well as game-based learning under tournaments which are based on Game Theory (Myerson, 2013). Game Theory deals with equilibria in games. Each player in games internalizes a strategy which is unlikely to be changed. There are many equilibrium concepts in Game Theory. The most popular one is the Nash equilibrium (Nash, 1950). Game Theory tournaments make students to get intrinsic motivation which increases learning performance (Bergin & Reilly, 2005). Moreover, competitions mimic real-world projects where usually are strict deadlines to be obeyed. In Cagiltay, Ozcelik and Ozcelik (2015), competition-based learning is applied to a game-based learning environment. It is showed that learning performance is increased significantly by adding competitions to serious games.

Augmented Reality (AR) is used to better understand real-world environment by providing additional virtual information according to the real-world context. In Hwang et al. (2016), a learning environment based on AR is powered by applying competition-based learning. It is showed that not only learning attitudes of students but also learning performance of them are improved.

In Çulha (2016), competition-based learning is applied to agile software engineering. Agile software development has short development cycles including all the activities of a software development project (Schwaber, 2004). These iterations in software development resembles tournaments in games. Both freshen motivation and increase performance. In Latih et al. (2018), competition-based learning is applied to a computer programming course using a competition tool. The results show that competitive atmosphere of the course motivates students and improve their performances.

In Chung (2008), competition-based learning is applied in classes to get active learning. In active learning, students take active roles in learning processes. The results show that students become more motivated, study harder, and ask more questions. Eventually their learning performance is enhanced. A drawback is observed as a result of applying competition-based learning in teams. There is less cooperation among the teams because of competitions. This drawback can be eliminated by requiring teams to share their studies via presentations.

In Bornstein and Erev (1994), the relation between competition and cooperation in teams is investigated. When the competition among the teams is intensified, it results in an increase in the cooperation among in the members of the same team. Competitions increase communication within teams.

In Francese et al. (2015), competition-based learning is applied to a project-based learning course on mobile application development. The project is realized in teams. The competition among the teams increases the collaboration in the same teams.

In Desai (2014), an innovative business capstone course, which deals with real business problems, is powered by applying competition-based learning to prepare students to real world business environment.

Competition-based learning has not been applied to blockchain programming before. However, difficulties in learning new concepts of blockchain technologies point competition-based learning. In this paper, competition-based learning is applied to blockchain programming using a proposed methodology. Competition-based learning, active learning, team-based learning, and project-based learning work well together. Competition-based learning supports active learning and collaboration in teams. Project-based learning creates a goal that supports active learning. In this paper, these methods are harmonized to achieve stronger learning performance for blockchain programming.

3. Methodology

Competition-based learning is applied to smart contract development using a proposed methodology. The steps of the proposed methodology are the following:

- Determine the concepts which should be learned by students
- Prioritize the determined concepts, and sort them according to their priorities
- Take at most 5 determined concepts from the sorted list, and call them as learning concepts
- Design a project which includes the learning concepts
- Redesign the project to include competitions, which should not have preferably an ultimate solution
- Assign the project to teams, which should include 3-4 students
- Do 3 competitions, finish them in the class, and order the teams according to their competition results

The methodology is seen in Figure 1. Teams start a competition, finish it with an order, and start a new competition. After each competition, teams get stronger.

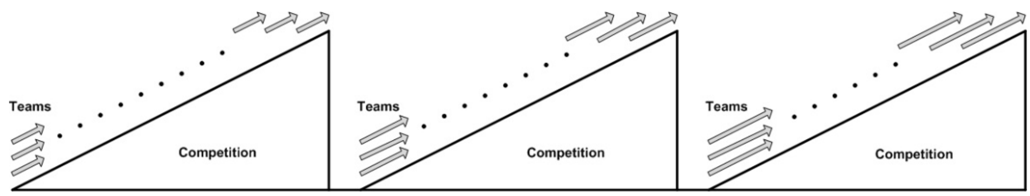

Fig. 1. The Methodology

3.1. Application of the Methodology

Competition-based learning is applied in the course SE427 Blockchain and Cryptocurrency Technologies in the Department of Software Engineering at Atilim University. 16 students are enrolled to the course. 5 teams are formed, of which one has 4 students, and the rest have 3 students. First, learning concepts are determined, and they are prioritized. A scale of 1 to 10 is used for prioritization where 10 represents the highest priority and 1 represents the smallest one. Table 1 shows the learning concepts with their priorities. In the second column, some details about learning concepts are given.

Table 1.

Learning concepts

Learning Concepts	Details	Priority (1-10)
blockchain	structure of blockchain, exploring blockchain	10
execution of smart contracts	editors, deployment, execution	9
public and private keys	signing	8
Solidity	programming with Solidity	8
interaction of smart contracts	calling smart contracts from a smart contract	8
smart contract	concept of smart contract	7
cryptocurrency	transferring, usage	7
cryptography	basics of cryptography	6
smart contract development	Solidity development	6
smart contract deployment	deployment in Remix	6
usage of cryptocurrency	transferring cryptocurrencies	5
user interaction with smart contracts	using Metamask	5
Ethereum	Ethereum platform	5
Ether	usage of Ethers	4
Ethereum Virtual Machine	bytecodes of Solidity programs	4
Remix	default editor of Ethereum platform	4
test network	Rinkeby test network	4
gas	required money to execute smart contracts	3
getting Ethers	getting test Ethers form https://faucet.rinkeby.io/	3
Metamask	connector to Ethereum blockchain	2
web3	connector library to Ethereum blockchain	2

According to the priorities, the most important learning concepts are taken. These are blockchain, execution of smart contracts, public and private keys, Solidity, and interaction of smart contracts. These most important learning concepts are included in the design of the project. Firstly, students should figure out what a real blockchain looks like. Therefore, the blockchain of the first Turing-complete smart contract development platform Ethereum is the starting point. The main Ethereum network which works with real currency Ethers is not suitable for the project. However, there are test networks for Ethereum. Rinkeby test network which is one of them is placed as the platform of the project. The smart contracts will be deployed there and will interact with others. The interaction of the teams with the smart contracts will be through their special accounts. Actually, these accounts are private keys. Therefore, for each team an account is generated via Metamask, which is a connector to the blockchain. Private keys of the accounts are given to corresponding teams who are responsible for the safety of their special accounts. At the center of the project, a central smart contract is thought, and the smart contracts of the teams can interact with it only through their accounts. All these developments should be done with Solidity because it is the main high-level programming language of the Ethereum platform.

After designing the framework of the project, competition logic is embedded to the project. For the competition logic, it is preferable that there should be no ultimate solution. Therefore, even teams can present their best performances, there would be enough room for later performances. These types of competition problems can be taken from mathematics because flexible problems can be easily constructed with numbers. However, complex mathematical problems should not be selected because they change the aim of the project. As a simple competition, a counter is thought. The central smart contract will have a counter, and the smart contracts of the teams will increment the counter. How much teams increment the counter will be their competition scores which will be used to measure their performances.

The counter will be incremented by teams, and the teams will not reach the upper limit of the counter. Therefore, there is no ultimate solution for the teams so that they can increment the counter every time. Consequently, the problem is very suitable for competitions.

The description of the designed project is the following:

Smart contracts will be developed in Solidity and a central contact with a counter will be incremented on the Ethereum test network Rinkeby. Solidity program version “pragma solidity ^0.5.1” will be used in smart contracts.

The name of the central smart contract is CommonCounterContract. The address of the contract will be given. The interface of the central contract is shown in Figure 1.

```
contract CommonCounterContract {

  function setTeamContract(address contractAddress) public;

  function getMyTeamContract() public view returns (address);

  function getCounter() public payable returns (int);

  function getMyTeamCount() public view returns (int);

  function incrementCounter(int prediction) public payable;

}
```

Fig. 2. Interface of the central smart contract

Each team will develop a smart contract, and then set their contract address to the central contract by calling the function “setTeamContract”. This operation can be realized only by the team own account. Teams may check their active contract address via the function “getMyTeamContract”. Teams can learn the current value of the counter using the function “getCounter”. This function is a “payable” function, and can be called with a 0.1 test Ether. Teams can increment the counter using the function “incrementCounter”. This function is also a “payable” function, and can be called with a 0.5 test Ether. It takes the last counter value, and if the counter value is correct, it increments the counter by 1. Teams can learn their total increment count using the function “getMyTeamCount”.

The project will be completed in 3 competitions. Each competition will be 2 weeks. For each competition, the following documents will be submitted also.

- The source code of the developed smart contract and its address
- Competition report which will include the followings:
  - Description of the developed smart contract
  - Challenges in the development, and solutions of them
  - Tasks which will be done in the next competitions or in the future
  - Effort which has been spent in hours

4. Findings

Table 2 shows the results of the competitions. In the first competition, Team 2 and Team4 incremented the counter whereas other teams did not increment. In the second competition, all the teams incremented the counter. Therefore, the second competition shows properties of a good competition. In the last competition, properties of a good competition continued because total count of the counter incrementations increased by nearly two times from the second competition to the third competition.

Table 2.

Competition results

	Competition 1		**Competition 2**		**Competition 3**	
	Count	**Order**	**Count**	**Order**	**Count**	**Order**
Team 1	0	3	16	4	81	5
Team 2	334	1	117	3	148	3
Team 3	0	3	263	1	452	1
Team 4	1	2	201	2	94	4
Team 5	0	3	9	5	387	2

In the first competition, only two teams incremented the counter. In the second competition, the other teams also incremented the counter. Therefore, in two competitions teams learned smart contract development and execution. In the third competition, all the teams presented good performances.

5. Discussion

At the beginning of the project, there are many things to learn for students. First of all, they realized that they need an editor to write code in Solidity. Remix which is the default online editor of the Ethereum platform was the first choice of the students, whereas some of them used also Visual Studio Code editor.

Teams have their own private keys. They learned how to use these keys. All of them used Metamask, and imported their private keys using Metamask to manage their special accounts. Some of them also searched for usage of Mist which is a wallet and a browser for Ethereum. Moreover, some students tried to use the library web3 for dealing with accounts.

Rinkeby test network needs test Ethers to execute smart contracts. In addition, the central contract works with Ethers. Therefore, the students began to get Ethers from the Rinkeby Faucet at the address https://faucet.rinkeby.io. Limited test Ethers are given for a duration from the Rinkeby Faucet so that students applied there periodically. Students learned how to transfer and get test Ethers. Students used test Ethers to deploy and execute smart contracts. Used Ethers for execution of this kind of transactions are called gas. Students learned the gas concept. In short, they have learned cryptocurrency concept.

Students need to explore transactions on the blockchain. Most of them used Etherscan at the address https://rinkeby.etherscan.io/ as a blockchain explorer. They analyzed their transactions and transactions of other teams. Moreover, they examined internal transactions and bytecodes of smart contracts.

The central contract is designed to allow interaction of smart contracts. In other words, its counter can only be incremented by other smart contracts. It cannot be incremented by team accounts directly. Therefore, teams developed their smart contracts, and incremented the central counter with their smart contracts. Moreover, getting current value of the counter and incrementing it are designed as payable. In order to execute them, the calling smart contract should have test Ethers, and should have been coded accordingly. These are new concepts and a bit tricky. Students also learned these mechanisms. Briefly, students learned interactions of smart contracts.

Team size was 3 except one team with size 4. According to the observations, 3 as a team size is very optimal for competitions. 2 as a team size is poor for being a team. Moreover, team size greater than 3 is redundant for course projects because at that time some team members could not interact enough with other team members. Competitions force teams to increase their internal communication. It is observed that communication among team members is maximized when the team size is 3. Maximized communication results in a good teamwork, and teams become real teams.

Number of competitions was kept as 3. According to the observations, 3 is optimal to preserve excitement of students. Usually, two competitions is not enough because students actually realize the excitement in the second competition. Therefore, another competition is required by students. However, after 3 competitions the excitement of students deplete. In short, 3 competitions are suitable to maximize learning period of students.

Competitions also resemble mining in blockchain. Mining means that if blockchain nodes find a valid block for appending to the blockchain, they get rewards of the current block mining competition. Then, new block mining competition starts for finding new valid blocks. Therefore, competitions fit to blockchain technology teaching very well in this perspective also.

Students said that they asked many questions to their friends and their teachers about blockchain. However, all of them could not answered their questions. They searched in internet. Likewise, they could not find enough material there. In brief, they tried to learn blockchain technologies actively. They practiced active learning. Some other students said that they had not studied for the capstone project as much as they had studied for this project. Some of the students said that they learned blockchain. This was the aim of the application of the competition-based learning, and the aim was realized. Students have become to view the whole picture of blockchain and cryptocurrency technologies.

Table 3 shows the efforts of the teams. In the first competition, they spent their maximum effort to learn many new concepts. In the second competition, their total effort shows that their attention to the project is good. In other words, they felt the excitement of the competition. In the third competition, their total effort decreased.

Table 3.

Efforts of the teams (in hours)

	Competition 1	Competition 2	Competition 3	Total
Team 1	15	9	7	31
Team 2	21	12	8	41
Team 3	38	46	14	98
Team 4	31	27	18	76
Team 5	13	8	22	43
Total	118	102	69	289

Grading of the project is designed as 30%, 40%, and 30% for the competitions respectively. The first competition passes with learning activities muchly, and in the last competition the excitement of students decreases. However, in the second competition teams present their best performances. Therefore, the grading is favored for the second competition.

6. Conclusion

Blockchain and its related technologies are comprehensive. Cryptography, cryptocurrency, immutable data, and decentralization are main concepts in these technologies. Smart contract development means programming on blockchain. In order to learn smart contract development, the related concepts should have been understood also. However, these concepts are new or not easily understandable currently. Therefore, learning of smart contract development is difficult. In order to increase learning efficiency, competition-based learning is applied to blockchain programming.

Competition-based learning has been used in many areas successfully for years. In this paper, this approach is applied to the uncharted smart contract development area. A methodology is presented to design competition-based learning. Learning is realized through a project with teams. Therefore, project-based and team-based learning are also included in the methodology. Moreover, competitions force students to active learning implicitly. Students try to take active roles in the learning process to win competitions.

Results of the application of competition-based learning are presented. Basically, students studied harder and tried to learn blockchain and blockchain technologies through many channels like friends, teachers, books, and internet. They practiced active learning.

As an observation, project-based learning is well suited for teams of size 3 with 3 competitions. The applied methodology also emphasizes these numbers. Moreover, competitions should be finished in front of the students to increase their willingness. Main contribution of this paper is the application of competition-based learning to blockchain programming.

Blokzincir Programlamanın Yarışma-Tabanlı Öğrenimi

Özet

Bir yıkıcı yenileşim teknolojisi olan blokzincir, birçok teknolojiyi etkilemektedir ve birçok başka teknolojiyi de etkileyecektir. Blokzincir teknolojilerinin ana özelliği, merkezi yetki olmadan güven sağlamasıdır. Bu özellik, değiştirilemez veri ve merkezsizleşme aracılığıyla gerçekleşmektedir. Ayrıca, blokzincir, güvenli altyapı sağlayan şifreleme biliminin ilkeleri üzerine kurulmuştur. Blokzincir teknolojilerinin ilk uygulaması, Bitcoin kripto parasıdır. Bitcoin'in doğuşundan sonra kriptoparalar, finansal sistemleri değiştirmeye başlamıştır. Blokzinciri öğrenmek zordur çünkü blokzincir ve ilgili teknolojileri, birçok kişi için tuhaftır. Blokzincir teknolojilerini kavramak için şifreleme, kripto para, değiştirilemez veri ve merkezsizleşme gibi kavramları anlamış olmak gerekmektedir. Bu nedenle, blokzincir ve ilgili teknolojilerini, etkin öğrenme yöntemleriyle öğrenmek gerekmektedir. Proje-tabanlı öğrenme, takım-tabanlı öğrenme, etkin öğrenme ve yarışma-tabanlı öğrenme, blokzinciri etkin öğretmek için kullanılabilir. Yarışma-tabanlı öğrenme, birçok alanda yıllardır başarıyla uygulanmaktadır. Akıllı sözleşme geliştirme, blokzincir teknolojilerinin programlama bölümüdür. Bu yazıda, yarışma-tabanlı öğrenme, öğrenme etkinliğini arttırmak için blokzincir programlamaya uygulanmıştır. Ayrıca, yarışma-tabanlı öğrenmeyi, blokzincir programlamaya uygulamak için bir yöntem de sunulmuştur.

Anahtar kelimeler: Blokzincir programlama, rekabet tabanlı öğrenme, blokzincir geliştirme

References

Altin, H., Pedaste, M. (2013). Learning approaches to applying robotics in science education. *Journal of Baltic Science Education, 12*(3), 365–377.

Bergin, S., & Reilly, R. (2005). The influence of motivation and comfort-level on learning to program.

Bornstein, G., & Erev, I. (1994). The enhancing effect of intergroup competition on group performance. *International journal of conflict management, 5*(3), 271-283.

Burguillo, J. C. (2010). Using game theory and competition-based learning to stimulate student motivation and performance. *Computers & Education, 55*(2), 566-575.

Buterin, V. (2013). Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

Cagiltay, N. E., Ozcelik, E., & Ozcelik, N. S. (2015). The effect of competition on learning in games. *Computers & Education, 87*, 35-41.

Chakraborty, P., Shahriyar, R., Iqbal, A., & Bosu, A. (2018, October). Understanding the software development practices of blockchain projects: A survey. *In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (p. 28). ACM.

Chung, C. J. (2008, April). Learning through competitions–competition based learning (CBL). In LTUCTL Conference poster session.

Çulha, D. (2016). Applying Competition-based Learning to Agile Software Engineering. Comput. Appl. Eng. Educ. doi: 10.1002/cae.21716

Dannen, C. (2017). *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners.* New York, NY, USA.

Desai, A., Tippins, M., & Arbaugh, J. B. (2014). Learning through collaboration and competition: incorporating problem-based learning and competition-based learning in a capstone course. *Organization Management Journal, 11*(4), 258-271.

Francese, R., Gravino, C., Risi, M., Scanniello, G., & Tortora, G. (2015). Using Project-Based-Learning in a mobile application development course—An experience report. *Journal of Visual Languages & Computing, 31*, 196-205.

Hwang, G. J., Wu, P. H., Chen, C. C., & Tu, N. T. (2016). Effects of an augmented reality-based educational game on students' learning achievements and attitudes in real-world observations. *Interactive Learning Environments, 24*(8), 1895-1906.

Hyyppä, K. (2004). *Competition - An Efficient Method to get Students Committed.* ICALT. IEEE Computer Society. ISBN: 0-7695-2181

Jung, S. (2013). Experiences in Developing an Experimental Robotics Course Program for Undergraduate Education. *IEEE Trans. Education, 56,* 129-136.

Latih, R., Bakar, M. A., Jailani, N., Ali, N. M., Salleh, S. M., & Zin, A. M. (2018). Challenge-based programming learning design. *International Journal on Advanced Science, Engineering and Information Technology, 8*(5), 1912-1918.

Marchesi, M., Marchesi, L., & Tonelli, R. (2018, October). An agile software engineering method to design blockchain applications. *In Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia* (p. 3). ACM.

Myerson, R. B. (2013). *Game theory.* Harvard university press.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Nash, J. F. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences, 36*(1), 48-49.

Paulik, M. J., Krishnan, M. (1999). An autonomous ground vehicle competition-driven capstone design course. *Proc. 1999 Frontiers Educ. Conf.* Nov 1999.

Paulik, M.J., Krishnan, N.M. (2001). A competition-motivated capstone design course: the result of a fifteen-year evolution. *IEEE Trans. Education, 44*, 67-75.

Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist, 50*(4), 258-283.

Schwaber, K. (2004). *Agile Project Management with Scrum,* Washington, Microsoft Press.

Swan, M. (2015). *Blockchain: Blueprint for a new economy.* " O'Reilly Media, Inc.".