Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools

Çağdaş Üsfekes[1, 2], Murat Yilmaz[1], Eray Tuzun[2], Paul M. Clarke[3, 4], Rory V. O'Connor[3, 4]
[1]Computer Engineering, Çankaya University, Turkey
myilmaz@cankaya.edu.tr
[2] HAVELSAN A.Ş, Turkey
{cusfekes, etuzun}@havelsan.com.tr
[3]School of Computing, Dublin City University, Ireland
[4] Lero – the Irish Software Research Centre
{paul.m.clarke, rory.oconnor}@dcu.ie,

Abstract. Application lifecycle management (ALM) highlights the rules of the road for the entire software ecosystems' lifecycle. Successful ALM enables clarity around the entire delivery effort, from defining requirements to deploying the software product. One of the challenges in software engineering today is to orchestrate ALM tools to a set of software projects effectively. In particular, it is challenging for software practitioners to continuously fully engage with the tasks that are assigned to them. The goal of this study is to address such situations using a game theoretic approach by utilizing a reward mechanism, which we intent to test in a medium-sized software development organization. Based on a set of game elements, this study proposes an auction mechanism to address human resource allocation and task optimization issues, and consequently tackle the potential problem of software practitioners' engagement.

1 Introduction

The notion of games is relevant to studies of social aspects of software development, which have gained an increasing attention among researchers. Recently, a number of researchers have conducted research to explore the potential usage of games in software development activities in terms of collective behavior: altruism and selfishness that ultimately affect the health of a software project. Games are special kind of social activities, which can easily highlight the social interactions or engagements that could offer a variety of measurable societal outcomes. Over the last decade, games have reshaped the ways of communication by the help of the social media to promote cooperation and competition. Serious games are used for game-based social skill training that helps individuals to gain social responsibility through the creation of fun and engaging environments. Emerging trends improve the popularity of among researchers and practitioners who have redefined the notion of games in non-gaming contexts. Consequently, the term gamification (i.e. the use of game elements in non-gaming practices) becomes an emerging subject for improving the software development processes. It not only has a great potential to align individuals' motivations with software development task but also helpful to address a variety of information technology related issues.

Defining application lifecycle management (ALM) is not straightforward. ALM can be grouped in three distinct areas. These are governance, development, and operations. In governance step, we have to be sure the application always provides what the business needs [1]. Governance includes the all periods of ALM so this is the most important area of ALM. Development is a basic part of every software product's custom lifecycle.

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

Operation step comes after development step. After deployment, every product needs to monitored and managed.

The whole spectrum of ALM process is addressed with various ALM tools. In particular, it is challenging for software practitioners to continuously fully engage with the tasks that are assigned to them in these ALM tools. In this study, we will address such situations using a game theoretic approach by utilizing a reward mechanism. Based on a set of game elements, this study proposes an auction mechanism to address human resource allocation and task optimization issues, and consequently tackle the potential problem of software practitioners' engagement.

The remaining part of the paper proceeds as follows: Section 2 gives a brief overview of theory of games in software engineering literature. In Section 3, we discuss the foundations of our proposed game based resource distribution framework, and finally in Section 4 provides a discussion and outlines future work.

2 Theory of Games in Software Engineering Literature

Research into games has a long history. The theory of games first appeared in the literature at 1930s. A game highlights strategic interactions among individuals, teams, units, or infrastructures. Historically, research investigating the individuals' interactions associated with games has focused on analytical methods and tools to aid the decision-making process [2]. Around the early 1960s, small-scale research and case studies began to emerge linking theory of games with social science successfully. Especially, in last fifteen years, games become popular. In last fifteen years, companies are using game elements to analysis their employee characteristics.

Game Theory is a set of analytical tools, which can be used to model the interactions between participants (e.g. individuals, companies, nations, etc.) in a game form. In addition, it can be used to explore the actual or essential decisions and behaviors, and ultimately their consequences that may include tradeoffs or conflicts among individuals. The most important fact about game theory is that it assumes all players as rational. In other words, all players follow the rules of a game and hence their goal is to win. In the last decade, game theory was not used only in economy. It was used in psychology, biology, and computer science [3]. Game theory has both cooperative and non-cooperative forms. However, it is mostly known with its "non-cooperative" form [3]. In this approach, the goal is to design a controlled competition where selections of participants are likely to affect every single player's benefits. These players are considered as successful when they mind their own benefits based on a choice architecture. Nash [4] coined "Nash Equilibrium", which describes the optimal outcomes of a game by predicting the outcome of strategic interactions. In recent years, we can see many examples about using game theory in software engineering [5,6]. By the result of these examples, we can conclude that game theory can be used to address many challenges that are known in software development.

There are many examples about using the theory and practices of games and the use of game elements to address a set of problems in software development. For example, Lagesse [7] created a game theoretic model for assigning tasks to software practitioners. Cockburn [8] accepted that software development is a kind of game based on limited project resources, communication and coordination skills. Baskerville [9] analyzed high-speed internet development from a balancing game viewpoint that depends on high usage of resources. Sullivan [10] worked to evaluate software design decisions by

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

economic approach. Sazawal and Sudan [11] combined the theory of games and decision modeling structure to improving software design. In this work, they designed a game called "software design evaluation". This game aims to address problems between developers and customers. Moreover, they suggested a lightweight game theoretical analysis technique to assess software development teams.

Gao [12] designed a game theoretic model to configure software products and decision errors. Gao-hui [13] worked about depending corporate software developments to game theory. Soska et al. [14] worked about students in academic life. In this work, they designed a card game to teach students about software testing. In addition, Pedreira et al. [15] created a systematic map about usage of gamification in software engineering. By this work, they aimed to find opportunities for future works. In recent years, gamification becomes more popular in software engineering research. Sweedyk [16] worked about the popularity of gamification in academic programs and conferences. In 2016, Kitagawa and others created a game on code review. Code review has a big effect on software quality in development process and it aims to decrease the number of bugs [17]. Szabo [18] applied "Game Dev Tycoon" game to students for teaching software engineering. This game is about business simulation. Amir [19] worked about getting systems more gamified and effective with using gamification. Ranganathan [20] used gamification in hardware engineering. He supported a low power timer on circuit by a game theoretic model which is based on the "Nash equilibrium".

A game is a useful tool to reveal interpersonal conflicts. This situation is known as a "social dilemma". "Prisoner's dilemma" is a basic framework that often used by researchers to observe such issues. Hazzan and Dubinsky [21] suggest that "prisoner's dilemma" is useful to highlight the problems in software development. Fejis [22] designed a game theoretic model for software developers and testers. He worked about the results of this game and said that these results may cause "prisoner's dilemma". Costa [23] combined the "Prisoner's dilemma" with gamification and designed war and peace game by using this combination. In another work, Mortensen used "prisoner's dilemma" in security and privacy of web technologies. In this work, he defined seven strategies and created a strategy to exceed "prisoner's dilemma" of web technologies by using a set of strategies [24].

The software process improvement methods should cover various activities so as to improve the quality of the software product [25, 26]. These activities should be reevaluated by taking into account factors affecting software development activities (e.g. human factors in software development, social interaction problems, etc.). In software development process, people are not working alone; they are working in teams so all of these working activities accepted as a social activity [27]. The practitioners working in these development teams are affected by several social factors including but not limited to their working conditions, personalities, rationality and interdependence [28]. An important goal of software process improvement is to increase the quality of software development projects and comply the project plan and budget [3]. To this end, a coordination mechanism among development, maintenance and management is essential. For example, the readability of not only source code but also technical documentation is decreasing when project is getting bigger. Therefore, a software unit that has to work in a coordination and number of employees is increasing. The coordination level of these units affects the quality of a software product. At this point, the problems, which occur in software development, can be addressed by assigning responsibilities to the right people as more competent they become.

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

Several lines of evidence suggest that building a mechanism for automating the software development activities by designing game-like activities is essential [29, 30, 31]., Yılmaz [32] designed a game-like approach to explore the effects of team personality characteristics in software development activities. Yılmaz et al. [31] created a gamification approach to improve the software development process. The idea of creating an economic mechanism for software development is introduced by [30], which was one of the first serious discussions about the subject matter. One study by Yilmaz et al. [33] proposed an economic mechanism for improving the software development process. Yılmaz and O'Connor [34] suggested a complementary approach to ScrumBan to improve the software development process using gamification. In another work, Yilmaz and O'Connor [35] considered software development as an economic activity and created a market-based approach to investigate task assignment problems. Collectively, these studies confirm that using game-like approaches in the software development activities have a significant impact for software productivity improvements.

2.1 Reward Mechanisms

A reward mechanism is a feedback device, which is an important aspect of game design. A considerable amount of literature has been published on computational features of these mechanisms. Houk et al. [36] investigated the models of intelligent behaviors and its relation to reward mechanisms. Singh [37] proposed a reward mechanism for online learning systems. Lua [38] worked on a reward mechanism which is designed for P2P systems. Wang and Sun [39] explored reward mechanisms which was designed for computer games.

Reward mechanisms have a crucial impact on human learning and cognition. In addition, they are related to game elements. If a reward system is constructed properly, it is likely to improve the motivation of the participants. Game elements potentially assist people to solve problem in an enjoyable way, e.g. while they are working on routine tasks. Walz [40] defined a game as a closed system that depends on social and cultural fundamentals of cultural values. Gonzales [41] described the advantages of games for teaching a process in computer engineering. Qu [42] worked about teaching software engineering. Largo [43] collected lots of feedbacks from students and he examined game elements in learning process.

Big corporations are using more complex systems. These can be engineering management tools, financial automation tools etc. To use these systems efficiently, employees must be experienced. At this point, employees make more effort to use these systems efficiently. In this process using gamification accelerates the employees learning process. For example, in software engineering, Pariza [44] designed a game about traceability in software tests and while conducting source code inspections. He designed a game about traceability in software tests and code artifacts [45].

Application lifecycle management (ALM) defines the rules of an outlet for the entire software development lifecycle, which provides clarity around the entire delivery effort, from defining requirements to building, packaging, and deploying the software product [46]. Therefore, it supports a powerful reporting ability and traceability in development process [47]. These tools provide engineers with a single framework for the many modules that the software development process requires [48]. These modules can be requirement management, test management, build management, project management and source code management. All these different processes are integrated with each other successfully and this ability is very important on software project delivery. In this work, our aim is to create a reward mechanism to use ALM tools more efficiently.

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

3 Game Based Resource Distribution Framework for Application Lifecycle Management

Application lifecycle management systems do not suggest the most efficient methods to software developers while they are assigning tasks. The goal of this model is using individual choices to improve software productivity while developers are assigning tasks. Users can join multiple auctions which is defined in this software model. Auctions can be related with requirement analysis, software testing or etc. Therefore, users can choose the tasks that motive them the most from a pool with resource distribution method. This model is proposed as a resource management framework to define the task choices based on priority of software developer's selection. This system aims to make the task assignment and time planning in an efficient way.

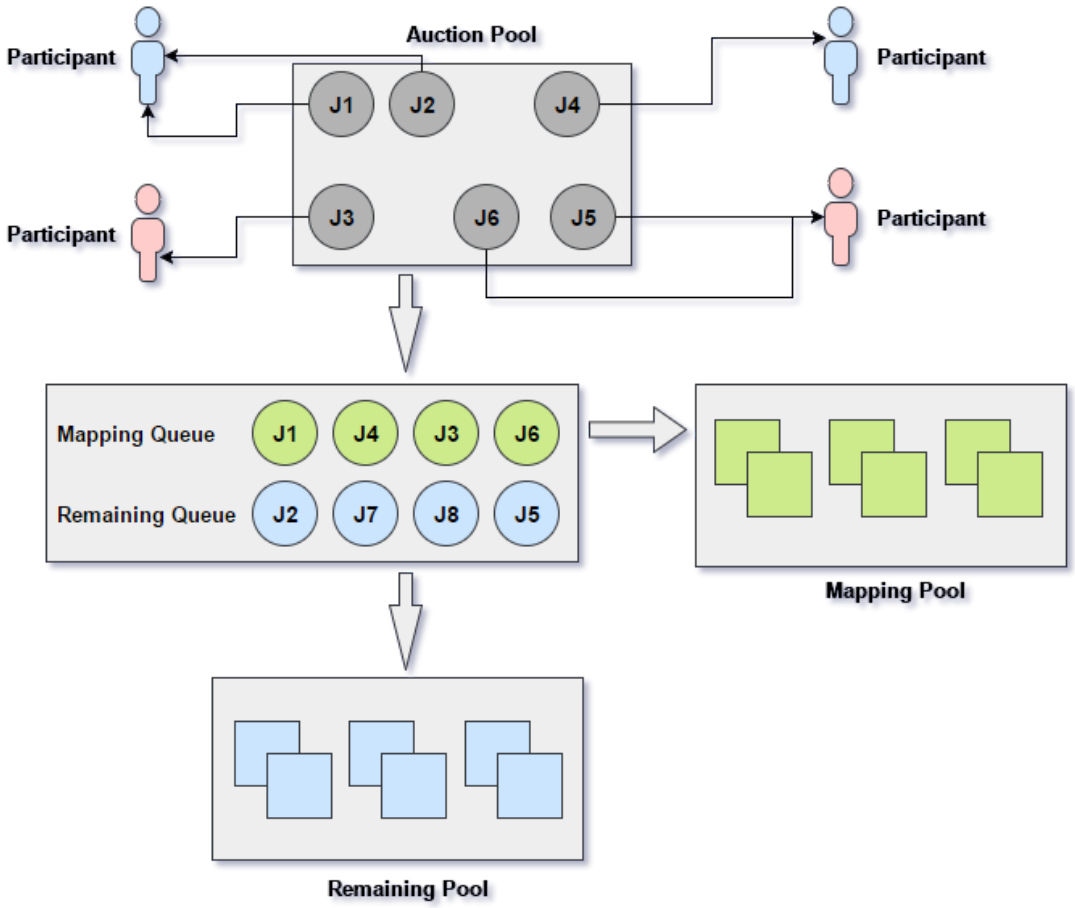

Figure 2: Auction-Based Resource Distribution Model to Software Lifecycle Management

The main aim of this mechanism is to reform the software development activities in a resource economy model where software practitioners have initial credits, which enable them to select these tasks regarding their preferences. Based on the proposed model, we announce the tasks to the software developers in an auction like structure. Similar to story cards, these tasks are based on their effort and complexity points. A practitioner requests a set of specific tasks depending on the amount of credit they might be able to pay by using auction mechanism. From these requests, proposed mechanism selects the

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

practitioner who desires to do this job the most. In this way, a gamification based value mapping occurs between tasks and resources. The system ensures that a user has to bid on their own budget and allows the price stay constant over time. The system uses a set of game elements to motivate its users such as giving reputation, badges and leaderboards (i.e. to create community leader with more privileges). Consequently, participants who finish their tasks in time are rewarded by the system based on the importance of their achievements. All this information is announced to system participant to foster their motivation (see Figure 2).

Using this technique, individuals in a team can bid for the work they would like to perform and in the context of their available credits. We believe that this could have interesting ramifications for productivity and knowledge diversity among individuals in software development teams. Here, we suggest that this is a useful vehicle for risk reduction in software companies, since everyone has the right to bid for work in the context of their credit position. Let's look to a metaphor – a golfing handicap. In amateur golfing competitions, individuals participate in competitions but their score is modified on the basis of their handicap/ability with the result that the winner is not the player who shot the absolute score for the round of golf but rather the winner who shot the lowest score taking into account their own ability. This means that everyone competes with the ability to win the competition and everyone is trying to improve his or her own personal performance.

4 Discussion

This preliminary resource allocation model explains the possibility of using an auction based reward mechanism in the service of distributing the tasks of a software development process. The goal is to propose a reward mechanism based on game design concepts and explore potential methods for application game elements to software development activities and ultimately improve software development practices. Overall, this study strengthens the idea that software practitioners should be able to select which tasks they prefer in an auction style mechanism based on a series of decision criteria (e.g. credits, reputation, and defined skills). To benefit from the power of game elements these choices are represented in a game like structure. Games may foster motivational factors, which potentially create a positive impact on practitioners' performance with a marked effect on task awareness. This preliminary study claims that playing an auction style business game provides participants a "mental workout" and routine activity pattern becomes less boring. In addition, it would guide individuals to create a habit of working in a more structural way. An implication of this approach is the possibility to improve the social structure of a software organization. A further benefit of this practice is to improve overall project awareness (as is noted in the discussion section) and potentially improved tacit knowledge distribution.

Our initial proposal was submitted to the management board of HAVELSAN where we received positive feedback, and it is selected for an initial funding. HAVELSAN is the largest software development company in Turkey with around 800 engineers. There are four main divisions in HAVELSAN. These are education and simulation systems, cyber security systems, command control and combat systems and information technologies. All these areas focus different sectors so various technologies and software development methodologies (such as agile, waterfall) are using in projects. Project groups are using different ALM tools such as TFS (Team Foundation Server), Atlassian etc. in software development processes. We are designing a web based serious game application that can communicate with these tools using restful services. Firstly, we selected TFS as a pilot

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

ALM tool. In the future, tools like Jira is also supported, so we can position this serious game application on top of the all software projects in HAVELSAN.

This study has surfaced many questions in need of further investigation. Future research should therefore concentrate on the implementation of the proposed model and more assessments is essential to determine its effectiveness.

Acknowledgement.

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094.

References

1. D. Chappell, "What is Application Lifecycle Management?", Chappell & Associates, 2008.

2. Yilmaz, Murat and O'Connor, Rory, Oyun Kuramı Kullanarak Yazılım Takımlarının Üretkenliğini Artırmak İçin Geliştirilen Bir Yazılım Süreç Mühendisliği Yaklaşımı, Turkish National Software Engineering Symposium, 2011

3. S. Zahran, Software Process Improvement: Practical Guidelines for Business Success. Addison Wesley, 1998.

4. Eric Maskin, Nash equilibrium and mechanism design, Institute for Advanced Study, Princeton University, United States, 2008

5. T. Dingsøyr, T. Dybå, and N. B. Moe, Agile Software Development: Current Research and Future Directions, 1st ed. Springer, 2010.

6. F. P. Deek, J. A. McHugh, and O. M. Eljabiri, Strategic software engineering: an interdisciplinary approach. CRC Press, 2005.

7. B. Lagesse, "A Game-Theoretical model for task assignment in project management," in 2006 IEEE International Conference on Management of Innovation and Technology, Singapore, 2006, pp. 678-680.

8. A. Cockburn, Agile software development: the cooperative game. Addison-Wesley, 2007., "A Game-Theoretical model for task assignment in project management," in 2006 IEEE International Conference on Management of Innovation and Technology, Singapore, 2006, pp. 678-680.

9. R. L. Baskerville, L. Levine, B. Ramesh, and J. Pries-Heje, "The high speed balancing game: How software companies cope with internet speed," Scandinavian Journal of Information Systems, vol. 16, no. 1, pp. 11–54, 2004.

10. K. Sullivan, P. Chalasani, and S. Jha, "Software design decisions as real options," University of Virginia, Tech. Rep., 1997.

11. V. Sazawal and N. Sudan, "Modeling software evolution with game theory," Trustworthy Software Development Processes, vol. 5543, pp. 354–365, 2009.

12. Xing Gao, Weijun Zhong, Shue Mei, A game-theory approach to configuration of detection software with decision errors, 2013

13. Nie Gao-hui, Analysis on Enterprise's Software Project Management Based on Game Theory, Management Science and Engineering, 2006

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

14. Alexander Soska, Jürgen Mottok, Christian Wolff, An experimental card game for software testing: Development, design and evaluation of a physical card game to deepen the knowledge of students in academic software testing education, Global Engineering Education Conference (EDUCON), 2016 IEEE, 2016

15. Oscar Pedreira, Félix García, Nieves Brisaboa, Mario Piattini, Gamification in software engineering – A systematic mapping, Information and Software Technology, v. 57, 2015

16. Elizabeth Sweedyk, Robert M. Keller, Fun and games: a new software engineering course, ITiCSE '05 Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education, Pages 138-142, 2005

17. Norihito Kitagawa, Hideaki Hata Nara, Akinori Ihara, Kiminao Kogiso, Kenichi Matsumoto, Code review participation: game theoretical modeling of reviewers in gerrit datasets, CHASE '16 Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering, Pages 64-67, 2016

18. Claudia Szabo, Evaluating GameDevTycoon for teaching software engineering, Proceeding SIGCSE '14 Proceedings of the 45th ACM technical symposium on Computer science education, Pages 403-408, 2014

19. Bilal Amir, Paul Ralph, Proposing a theory of gamification effectiveness, Proceeding ICSE Companion 2014 Companion Proceedings of the 36th International Conference on Software Engineering, Pages 626-627, 2014

20. N. Ranganathan, Ashok K. Murugavel, A low power scheduler using game theory, CODES+ISSS '03 Proceedings of the 1st IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis, Pages 126-131, 2003

21. O. Hazzan and Y. Dubinsky, "Social perspective of software development methods: The case of the prisoner dilemma and extreme programming," in Extreme Programming and Agile Processes in Software Engineering. Springer, 2005, pp. 74–81

22. L. Feijs, "Prisoner dilemma in software testing," Computer Science Reports, vol. 1, pp. 65–80, 2001.

23. Carlos J. Costa, Pedro J. Costa, A peace war game application, OSDOC '11 Proceedings of the 2011 Workshop on Open Source and Design of Communication, pp. 71-74, 2011

24. Peter Mortensen, Conrad Wai, Avoiding the prisoner's dilemma of the web, DUX '07 Proceedings of the 2007 conference on Designing for User eXperiences, 2007

25. Yilmaz, Murat and O'Connor, Rory, Oyun Kuramı Kullanarak Yazılım Takımlarının Üretkenliğini Artırmak İçin Geliştirilen Bir Yazılım Süreç Mühendisliği Yaklaşımı, Turkish National Software Engineering Symposium, 2011

26. R. Conradi and A. Fuggetta, "Improving Software Process Improvement," IEEE Software, vol. 19, no. 4, pp. 92–99, 2002.

27. Y. Dittrich, C. Floyd, and R. Klischewski, Social thinkingsoftware practice. The MIT Press, 2002.

28. M. Grechanik and D. E. Perry, "Analyzing Software Development as a Noncooperative Game," in IEE Seminar Digests, 2004, vol. 29.

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

29. Yilmaz, Murat (2013) A software process engineering approach to understanding software productivity and team personality characteristics: an empirical investigation. PhD thesis, Dublin City University.

30. Yilmaz, Murat and O'Connor, Rory (2010) Maximizing the value of the software development process by game theoretic analysis. In: 11th International Conference on Product Focused Software, 21-23 Jun 2010, Limerick, Ireland. ISBN 978-1-4503-0281-4

31. Yilmaz, Mert and Yilmaz, Murat and O'Connor, Rory and Clarke, Paul (2016) A gamification approach to improve the software development process by exploring the personality of software practitioners. In: Clarke, Paul and O'Connor, Rory and Rout, Terry and Dorling, Alec, (eds.) Software Process Improvement and Capability Determination. Communications in Computer and Information Science, 609 . Springer, pp. 71-83. ISBN 978-3-319-38980-6

32. Schwaber, Carey, et al. "The Changing Face of Application Lifecycle Management" Forrester Research, August 18, 2006.

33. Yilmaz, Murat and O'Connor, Rory and Collins, John (2010) Improving software development process through economic mechanism design. In: 17th European Software Process Improvement Conference, 1-3 Sept 2010, Grenoble, France. ISBN 978-3-642-15666-3

34. Yilmaz, Murat and O'Connor, Rory (2016) A Scrumban integrated gamification approach to guide software process improvement: a Turkish case study. Tehnicki Vjesnik (Technical Gazette), 23 (1). pp. 237-245. ISSN 1330-3651

35. Yilmaz, Murat and O'Connor, Rory (2012) A market based approach for resolving resource constrained task allocation problems in a software development process. In: 19th European Conference on Systems, Software and Services Process Improvement (EuroSPI 2012), 25-27 June 2012, Vienna, Austria.

36. James C. Houk, Joel L. Davis, David G. Beiser, " Models of Information Processing in the Basal Ganglia", MIT Press, pp. 185 - 185, 1994

37. Neetu Singh, Narendra S. Chaudhari, "Differential Reward Mechanism Based Online Learning Algorithm for URL-based Topic Classification",IEEE, Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, 2014

38. Kun Lua, Shiyu Wanga, Ling Xiea, Zhen Wanga, b, Mingchu Li, "A dynamic reward-based incentive mechanism: Reducing the cost of P2P systems", vol. 112, pp. 105 - 113, 2016

39. Hao Wang, Chuen-Tsai, Game Reward Systems: Gaming Experiences and Social Meanings, 2011

40. Steffen P. Walz, Sebastian Deterding, "Gamification and Learning", MIT Press, pp. 688, 2014

41. Carina Soledad González, Alberto Mora Carreño, "Methodological proposal for gamification in the computer engineering teaching", IEEE, Computers in Education (SIIE), 2014 International Symposium on, 2014

Üsfekes, Ç., Yilmaz, M., Tuzun, E., Clarke, P. M., & O'Connor, R. V. (2017). Examining Reward Mechanisms for Effective Usage of Application Lifecycle Management Tools. In J. Stolfa, S. Stolfa, R. V. O'Connor, & R. Messnarz (Eds.), Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6–8, 2017, Proceedings (pp. 259–268). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-64218-5_21

42. Wei-Qing Qu, Yan-Fei Zhao, Ming Wang, Bang-Quan Liu, "Research on teaching gamification of software engineering", IEEE, Computer Science & Education (ICCSE), 2014 9th International Conference on, 2014

43. Faraón Largo, Francisco Durán, Carlos Arnedo, Patricia Rosique, Rosana Cuerda, Rafael Carmona, "Gamification of the learning process: lessons learned", IEEE, IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, pp. 1 - 1, 2016

44. Reza Meimandi Parizi, "On the gamification of human-centric traceability tasks in software testing and coding", IEEE, Software Engineering Research, Management and Applications (SERA), 2016 IEEE 14th International Conference on, 2016

45. Reza Meimandi Parizi, Asem Kasem, Azween Abdullah, "Towards gamification in software traceability: Between test and code artifacts", Software Technologies (ICSOFT), 2015 10th International Joint Conference on, 2015

46. D. Chappell, "What is Application Lifecycle Management?", Chappell & Associates, 2008.

47. Ahsen İkbal Aytekin, Eray Tüzün, Yagup Macit, Bedir Tekinerdoğan, Uygulama Yaşam Döngüsü Yönetimi - Sistematik Eşleme Çalışması, UYMS, 2015

48. Shaw, K. (2007) "Application Lifecycle Management for the Enterprise", Serena Software, White Paper, http://www.serena.com/Docs/Repository/company/Serena_ALM_2.0_For_t.pdf (available 24.04.2008).