
# A SYSTEMATIC REVIEW OF GAMIFICATION IN THE CONTEXT OF SOFTWARE DEVELOPMENT LANDSCAPES

**SERHAN OLGUN**

**JANUARY 2018**

# A SYSTEMATIC REVIEW OF GAMIFICATION IN THE CONTEXT OF SOFTWARE DEVELOPMENT LANDSCAPES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY
SERHAN OLGUN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE IN
COMPUTER ENGINEERING
DEPARTMENT

JANUARY 2018

Title of the Thesis: **A systematic review of gamification in the context of software development landscapes**

Submitted by **Serhan OLGUN**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

---

Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Prof. Dr. Erdoğan DOĞDU
Head of Deparment

This is to certify that we have read this thesis and that in our opinion it is fully adequate,in scope and quality, as a thesis for the degree of Master of Science.

---

Yrd. Doç. Dr. Murat YILMAZ
Supervisor

**Examination Date: 12.01.2018**

**Examining Committee Members**

| | | |
|---|---|---|
| Prof. Dr. Alok MISHRA | (Atılım Univ.) | --- |
| Assist. Prof. Dr. Özgür Tolga PUSATLI | (Çankaya Univ.) | --- |
| Assist. Prof. Dr. Murat YILMAZ | (Çankaya Univ.) | --- |

## STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Serhan OLGUN
Signature :
Date : 29.01.2018

## ABSTRACT

### A SYSTEMATIC REVIEW OF GAMIFICATION IN THE CONTEXT OF SOFTWARE DEVELOPMENT LANDSCAPES

OLGUN, Serhan

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Murat YILMAZ

January 2018, 71 pages

The software development process is a complex activity that has interrelating steps to produce software products. Human related social factors such as user motivation, engagement, communication and collaboration issues affect the activities performed by software development organizations. In fact, software engineering researchers seek better ways to overcome the human related issues so as to improve the quality in software development. Consequently, research has been conducted to introduce new methodologies, techniques and concepts. As such, gamification has been introduced as a novel technique to enhance the software development process and ultimately resolve some of the identified obstacles that are related to human factors. However, there is a bias to adopt the gamification into the software development process because there are still insufficient evidences about its positive outcomes. In this study, a systematic literature review was rigorously conducted to investigate the application and to reveal the benefits of gamification in the software development landscapes. This study contributes to the area of software engineering research by proposing a classification approach to categorize gamification studies in software

development landscapes. The results of this research suggest that there has been a significant increase in empirical and theoretical studies that were recently published and many positive steps being taken towards better integration of gamification in software development organizations.

**Keywords**: gamification, software development, game elements, systematic review.

# ÖZ

## YAZILIM GELİŞTİRME MECRALARINDA OYUNLAŞTIRMA KAVRAMININ KULLANIMINA YÖNELİK BİR SİSTEMATİK GÖZDEN GEÇİRME ÇALIŞMASI

OLGUN, Serhan

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd.Doç.Dr. Murat YILMAZ

OCAK 2017, 71 sayfa

Yazılım geliştirme süreci, yazılım ürünleri üretmek için birbirini takip eden adımlardan oluşan karmaşık bir aktivitedir. Kullanıcıların motivasyonu, iletişimi, katılımı ve iş birliği gibi insan odaklı faktörler, yazılım geliştirme firmaları tarafından gerçekleştirilen bu aktiviteleri büyük ölçüde etkilemektedir. Bu yüzden, yazılım mühendisliği araştırmacıları, yazılım geliştirme süreçlerindeki kaliteyi artırmak ve insan kaynaklı bu sorunları çözmek için yeni ve daha iyi yöntemler araştırmaktadırlar. Bu sebeple, son zamanlarda bu konu ile ilgili yeni metodolojileri, teknikleri ve kavramları ortaya koyan araştırmalar yapılmaktadır. Buna bağlı olarak, oyunlaştırma kavramı yazılım geliştirme süreçlerini iyileştirmek ve insan kaynaklı faktörlerin birçoğunu çözmek için yeni bir teknik olarak tanıtılmıştır. Ancak bu yöntemin yazılım geliştirme süreçlerine olan olumlu katkıları ile ilgili hala yeterince kanıt bulunmadığından, bu yöntemin yazılım geliştirme süreçlerinde kullanılmasıyla ilgili birtakım kuşkular bulunmaktadır. Bu tez, oyunlaştırma kavramının yazılım geliştirme süreçlerine olan faydalarını göstermek için titizlikle yürütülmüş bir sistematik gözden geçirme çalışmasını içermektedir. Bu çalışma, yazılım geliştirme alanlarında oyunlaştırma ile ilgili çalışmaları kategorize etmek için sınıflandırma

yaklaşımını önererek yazılım mühendisliği alanındaki araştırmalara katkı sağlamaktadır. Bununla ilgili araştırma sonuçları ise hem teorik hem de deneysel çalışma yayınlarında belirgin bir artış olduğunu ve oyunlaştırmanın yazılım geliştirme süreçlerine entegre olabilmesi adına bir çok olumlu adımın atılmakta olduğunu göstermektedir.

**Anahtar Kelimeler**: oyunlaştırma, yazılım geliştirme, oyun öğeleri, sistematik literatür taraması.

## ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Assist. Prof. Dr. Murat YILMAZ for his supervision, special guidance, suggestions, and encouragement through the development of this study. I have a chance to make long discussions with him and he gave me valuable feedbacks and comments to make this study better.

Also, I would like to express my deep gratitude to my lovely family due to their endless moral support. During this process they always haunted with me and encourage me to finalize this study.

Finally, I would like to thank my fiancee for her endless moral support. She is the best and the most valuable person ever in my life. During this process she always stayed with me and never gave up loving me. Many thanks to my fiancee once again to make me feel always good and motivated to complete this study.

## TABLE OF CONTENTS

## LIST OF FIGURES

### FIGURES

## LIST OF TABLES

### TABLES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| SLR | SYSTEMATIC LITERATURE REVIEW |
| PBL | POINT, BADGES, LEADERBOARDS |
| SCRUT | Social Code Review Unifying Tool |
| MLQG | Microsoft Language Quality Game |
| BDD | Behavior Driven Development |
| EBM | Evidence Based Medicine |
| EBSE | Evidence Based Software Engineering |
| SE | Software Engineering |
| JAD | Joint Application Development |
| RAD | Rapid Application Development |

# CHAPTER 1

# INTRODUCTION

Software development organizations are professional business firms, which are founded so as to develop high quality software products that satisfy the customer expectations and their business objectives. In these organizations, valuable, robust and reliable software products are produced as a result of the development teams following an appropriate development process. Software product development may require an extensive development process, which includes analysis, development, testing and maintenance steps. Yilmaz states that [1, pp. 1] "*A software development process is considered as the coordination of structural social activities (e.g. management, production and maintenance) coupled and constrained with a set of individuals' (i.e. participants who perform the activities) roles and skills for producing software artifacts in a predefined productivity level*". Theoretically, software development process is an organized structure that has consecutive steps to produce software applications. This process is performed by small, medium or large-scaled experienced software development teams by communicating with their customers in software business companies. In other words, software development is a teamwork that requires quite effective people involvement, engagement, collaboration and motivation to accomplish the process steps, so it concerns both software developers and customers. In support, motivation is the most important key factor that affects the developers' productivity performance during the software development process [34]. As Kusumasari et al. [42] state that collaboration is another important human factor in software development process to reach the goals successfully.

Since it is a human centric activity, in the course of proceedings some obstacles may occur to achieve the goals. Human factors (e.g. user motivation, engagement, communication and collaboration) are major parameters for the success of the development process. This can be investigated by understanding the human role in the software development. Individuals can affect the complexity process by being a stakeholder (e.g. customer, developer or manager, etc.) and ultimately the quality of a software product. The following are just some of *the common reasons* for why these problems occurred during software production: (i) lack of communication among team members that causes collaboration problems, (ii) misunderstood of business requirements and objectives from customer, (iii) late performance evaluation, (iv)lack of reward systems by the management and (v) software practitioners with insufficient technical experiences [19].

With respect to the state of the art as described the idea is to use game elements to engage, motivate, train and monitor all the employees [20] to make them passionate to involve them in the whole development process in the software development landscapes. The gamification broadens a new horizon almost in all areas in the non-game contexts. Its description is with the following statement: "*gamification is the use of game design elements in non-game contexts*." [4, pp. 2]. In gamified contexts, a rewarding mechanism frequently exists to encourage people and excite people's attention to increase the engagement. If gamification applied to the software development it might bring several advantages. From employee perspective, it may help to increase the user motivation, engagement and collaboration. From managers' perspective, it may also have advantages for the performance management to monitor performance of team members in the software development teams.

The application of game elements in the context of software development landscapes is still not an obvious issue and needs research and experiments on it. Therefore, this study investigates the applicability of the use of game elements in the context of software development landscapes and how these game elements affects the quality of software development in software development landscapes in order to overcome the challenges in the software industry.

The purpose of this research is to investigate the use of game elements in the software development landscapes. To summarize, this study aims to address the following research questions:

**i)** What is currently known about people's motivation, engagement and performance issues in software development process?

**ii)** Which game elements and gamification approaches can be applied into software development landscapes to increase the quality and performance of software development?

**iii)** What are the examples of these gamification approaches for the software industry?

**iv)** How these gamification approaches affect the quality of software development in software business context?

This study consists of 6 main chapters (including the *Introduction* chapter) and is structured as follows:

**Chapter 2** presents background information about gamification, game elements, gamification in software development process.

**Chapter 3** details the research methodology that we conducted and gives information about how the systematic literature review was planned.

**Chapter 4** presents the analysis and results that are obtained from the studies.

**Chapter 5** presents the discussion of the studies.

**Chapter 6** discusses the future work, threats to validity, limitations and concludes the study.

# CHAPTER 2

# BACKGROUND

In this section background information about the related topics will be presented. First, we provide brief information about gamification. Then we will provide brief information about the game elements. After that, we will give background information about gamification and software development process.

## 2.1. Gamification

During the recent years, the popularity of gamification has been increased almost in all areas in which people involvement is extensively required [5]. It is used in non-digital service – based contexts such as marketing, education, health care services, etc. to increase the user engagement, motivation, attraction and social interaction [5] [26] while doing the related jobs. The term gamification is defined in [4, pp. 2] as "*using game design elements in non-game contexts to motivate and increase user activity and retention*". As it is clearly stated in its definition gamification uses game design elements to gamify environments and ultimately aims to change people's behavior in positive manner and keep engaged and motivated them in particular tasks in non-gaming environments such as workplaces, schools or in software development organizations. By the increase in the application of real life examples of gamification in different domains, the popularity and usage of it has been growth in the last years. One of the domains where gamification is popular is business innovations. In support, Gartner [29, pp. 1] states that

> "*by 2015, more than 50 percent of organizations that manage innovation processes will gamify those processes, according to Gartner, Inc. By 2014, a gamified service for consumer goods marketing and customer retention will become as important as Facebook, eBay or Amazon, and more than 70 percent of Global 2000 organizations will have at least one gamified application".*

Another domain where gamification is popular is education and mostly for training purposes [7]. The goal is to increase the motivation, engagement and productivity of students. Hamari et al [36] claims that education and learning are the most preferred environments in which gamification is used to enhance the motivation, engagement and make the learning process more enjoyful.

Gamification has also been used to maximize the user engagement and keep user motivated by entrepreneurs, customer oriented web site owners [8]. StackOverflow [9] is a good example where the game elements are intensely used. This web site is a knowledge exchange system for developers where users take badges or performance rates according to their activities in particular actions. Because of its effectiveness and efficiency, game design elements have also been applied in business environments to improve employee's performance while they accomplish their tasks and works [10]. The underlying cause why gamification is in fashion and growing significantly is psychological effects and positive behavioral contributions on the people. Huotari and Hamari state that gamification has both psychological and behavioral outcomes to encourage the people as the games have [30]. The concept of gamification consists of three parts which are [5]:

- **Motivational Contributions:** It implies that the motivational acquisition gained from the gamified environment.
- **Psychological Consequences:** It indicates that the psychological outcomes of gamified contexts.
- **Behavioral Consequences:** It clarifies that future behavioral results.

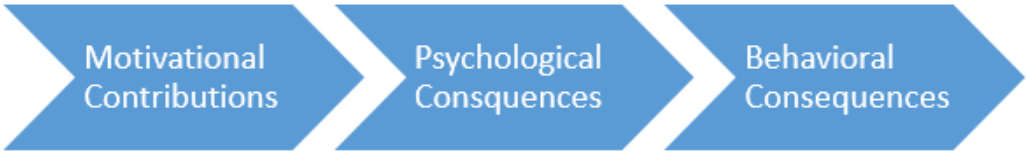


**Figure 1:** The Concept of Gamification (adapted from [5])

### 2.2. Game Elements

Gamification is not a game, it is the implementation of game elements and mechanics into the non - digital contexts to design a new attractive environment to overcome the real - life problems. Lombriser et al. [32, pp. 4] state that "*gamification does not create games, but rather takes certain game elements from classical video games and applies them to real-world (business) problems.*" Most commonly used game elements [42] that are used in gamified contexts are:

- Leaderboards
- Badges
- Points
- Rewards

Each of above elements has different perspectives, which are motivational and behavioral perspectives while building up the gamified contexts for monitoring, evaluation and giving feedback purposes. Leaderboards allow actors to compare them with each other. Badges show the accomplishments of the users and indicate how much s/he is successful among others. Points are used to indicate the grades and success rate of the user to keep engaged in gamified contexts. Rewards are meaningful motivator awards that are given to the players when their tasks are completed successfully. Therefore, the implication is that gamification and game elements is used to motivate and engage people from motivational and behavioral perspectives. In support, Yilmaz et al. [25] state that gamification is used to motivate and engage people by influencing the intrinsic and extrinsic motivational parameters by using game elements such as points, badges and leaderboards.

### 2.3. Gamification and Software Development Process

Software development is a complex process that has interrelating stages to produce a well – qualified software product within a planned time and a well - disciplined plan. Yilmaz et al. [28, pp. 1] state that "*A software process is a dynamic vehicle formed from a group of interrelated activities employed by a project or an organization. These activities not only produce products or services but also provide a road map for the software development within the expected schedule and budget.*" These consecutive stages are called as Software Development Life Cycle. According to the SWEBOK [54] software development life cycle consists of 5 consecutive steps respectively: *planning and requirement elicitation, design, implementation, testing and maintenance.*

***Figure 2*** shows the interrelating steps of software development process.

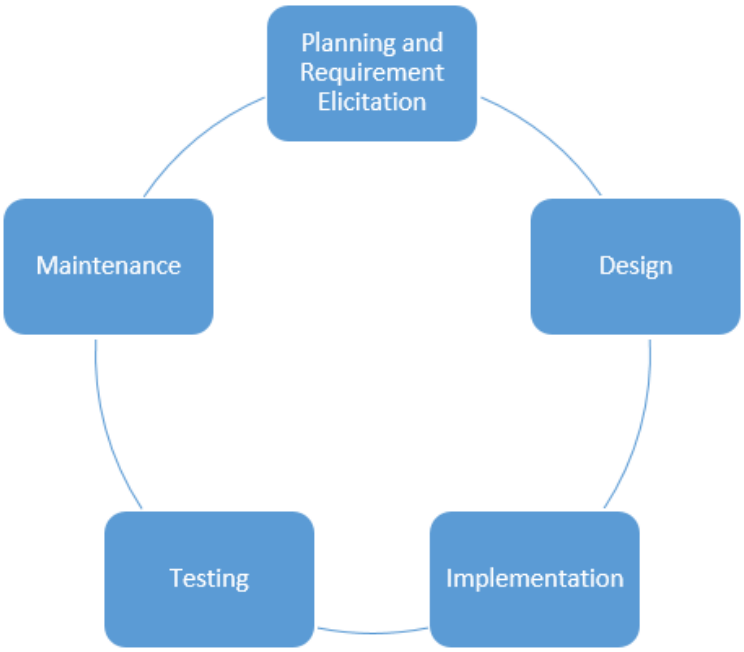


**Figure 2:** Software Development Process (adapted from [53])

### 2.3.1. Steps of Software Development Life Cycle (SDLC)

In this sub-section we will provide information about software development life cycle steps. These steps are [40] [53] [54]:

**Step 1: Planning and Requirement Elicitation**

It is the first step of software development process. In this phase, requirement elicitation phase expectations and business requirements and needs are gathered from the stakeholders and documented in detailed [43]. Therefore, it is the revealing phase what customer wants from the end - product. This phase is the most important part of the software development process because most critical issues are considered and taken into action [38], [43]. According to the requirement analysis, a schedule is created to produce the software within the limited time. Therefore, it is the most important and critical phase of SDLC.

**Step 2: Designing**

It is the next step after planning and requirement elicitation. In this phase, system and software architecture is designed according to the requirements by the system and software architectures [40].

**Step 3: Implementation**

It is the technical phase of software development life cycle. After designing process, implementation process starts. It is coding stage that are conducted by the software developers in the software development teams. It is the phase that prepares the product to the testing stage.

**Step 4: Testing**

Software testing is the activity that aims to find defects, bugs or errors in order to identify the validity and verifiability of the application whether it fulfills the requirements and expectations or not [27]. In this phase, the developed software product is tested by using different testing strategies and methodologies. Gajur et al. [44] state that there are four types of software testing strategies to provide a well – designed testing steps and cases in order to reach a successfully tested software application. These strategies are unit testing, integrating testing (top-down, bottom-

up, etc.), system testing (compatibility, recovery, security, etc.) and acceptance testing (alpha, beta testing). There are also software testing methodologies for an effective and appropriate software testing process to reach successful software testing goal. The major software testing methodologies are black-box testing, white-box testing and grey-box testing [44]. Therefore, using these testing strategies and methodologies software testing phase is conducted to ensure the quality of software product and determine whether the application meets the requirements and expectations according to requirement elicitation and analysis phase.

**Step 5: Maintenance**

It is the last step of software development life cycle starts by deploying software product to the customers' production environment. Then, bug fixing, change requests, new features are operated by obeying the rules.

In summary, the software development process is a complex activity that has interrelating activities. In each step, human is the most important actor and plays a crucial role; therefore, its quality may be affected negatively or even it may fail due to the human factors which are people involvement, motivation, engagement and collaboration. Thus, this kind of obstacles during the process should be solved to get the expected results. Hence, new techniques, tools and approaches are used to get rid of human related issues in software development in the software development landscapes. Therefore, applying game elements and using gamification is seemed as a solution for the human factors in software development to affect the quality in a positive way. Therefore, this study focuses on the applicability of the gamification in software development landscapes and how these game elements affect the quality of software development process in software development landscapes in order to overcome the challenges in the software industry. Thus, this makes gamification a promising field to overcome the challenges related to human factors such as people involvement, engagement, collaboration and motivation throughout software development process. Due to the above stated reasons, some software development process tools have started to integrate to game elements to benefit from gamification principles. Visual Studio Achievements [11], JIRA Hero [12], PropsToYou [13], ScrumKnowsy [14], MasterBranch [15], RedCritter [16] are examples of commercial

tools that are offering gamification in software development landscapes. Therefore, researchers and practitioners recognized that the game elements could be applied to the software development landscapes. However, the applicability of this issue is not obvious so it needs some research on it. Thus, a lot of proposals and academic researches about the topic have been published. In this paper, a research approach is proposed for understanding all following effects to improve the quality of a software development process by conducting a systematic literature review.

### 2.4. Summary

In summary, we provide background information about the related topics. As it is stated that [4, pp. 2] gamification is "*using game design elements in non-game contexts to motivate and increase user activity and retention*". As derived from the definition, gamification uses some game elements such as points, badges, leaderboards, etc. to increase behavioral and psychological outcomes. It is used in non – digital contexts such as software development process – which is a complex activity – to make it funnier and more attractive. However, applicability of gamification in some non – digital contexts (such as software development) needs to be investigated. Therefore, it needs some research to prove the applicability of gamification into software development context. Thus, we conducted a systematic literature review to understand how these gamification approaches affect the software development process. In Chapter 3, we will give information about the SLR methodology that we use while conducting this review.

# CHAPTER 3

# RESEARCH METHODOLOGY

In this chapter, we will provide information about Systematic Literature Review and why we need this methodology. Then, we will give the definitions about our research questions, search strategies and data sources, study selection criteria and procedures and data extraction and evaluation process of our research protocol in detailed.

## 3.1. Systematic Literature Review

We conducted a Systematic Literature Review (SLR) to identify a group of papers that discuss the application of game elements in software development landscapes. A systematic literature review is a research methodology to find out what we know and what we do not know based on the focused research questions. "*As a research area matures there is often a sharp increase in the number of reports and results made available, and it becomes important to summarize and provide overview*." [2, pp. 1] Therefore, systematic literature reviews should certainly contain the question that it tries to answer and should report fully on the methods that have been utilized. According to Kitchenham, "*Systematic Literature Review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest.*" [3, pp. 1] As described, the word systematic stands for planned, methodical acting and review stands for critical appraisal of works.

Systematic → Planned, methodical acting.
Review → Critical Appraisal of Works.
Synthesis → Get together findings.

This planned and methodical literature review is conducted by carrying out these steps. **Figure *3*** shows the Systematic Literature Review process and steps.

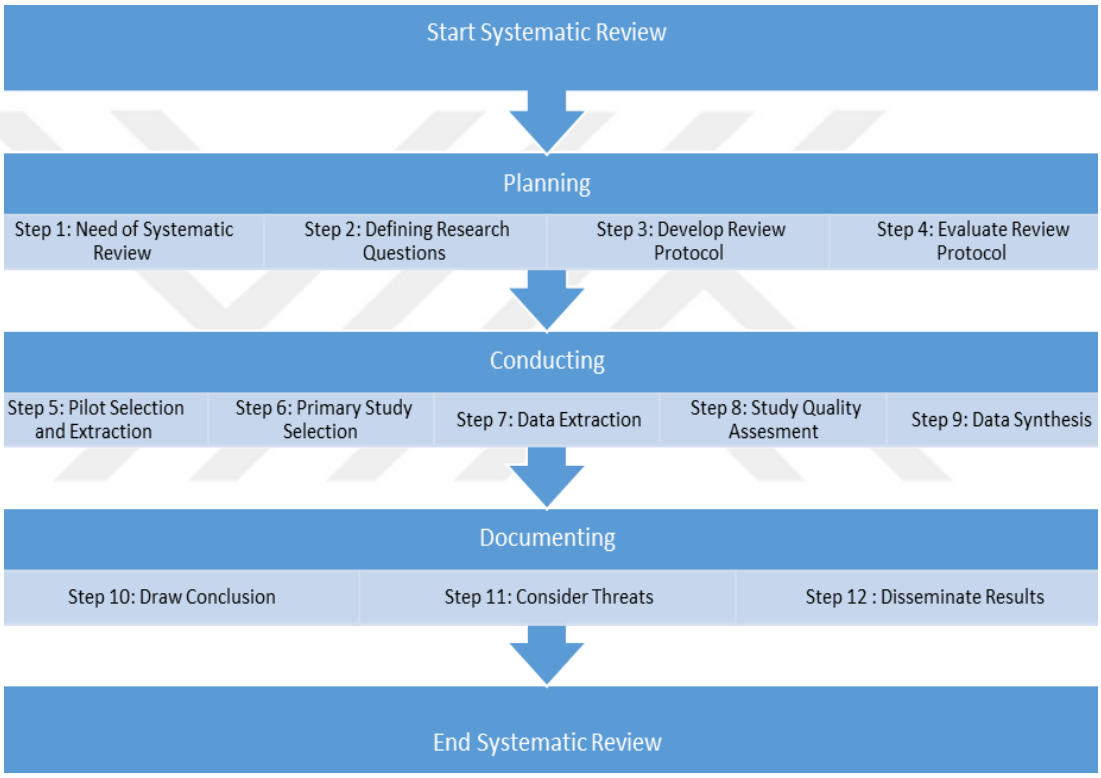


**Figure 3:** Systematic Literature Review Steps (Adapted from [6])

Depending on the **Figure *3***, this methodology consists of three main steps and these steps are briefly explained below:

- **Planning:** It is the first step of the review where the need for review needs to be identified, research questions are identified, and a review protocol is developed and evaluated. According to the **Figure *3***, the first step of planning phase is need for the systematic review. The purpose and why we

need this systematic review is described in the *Chapter 1* of this study. Then we identified some research questions (see **Table *1***) to attain our objective. These research questions are related to people's motivational, engagement, etc. issues, application of game elements and gamification and the affordances of gamification to the software development process. To reach our goal and find the answers to the research questions we need to develop a review protocol. First, we create search strings using major terms (see **Table *2***). These major terms are *Software Development Process* and *Game elements.* To extent the related search results and access the best data sources we used the alternative terms such as *software development lifecycle*, *software engineering, gamifying, gamification,* etc. At the end, to access the related data we built up the search string as follows:

*(Software Development Process OR ((software development) OR (software engineering) OR (software process) OR (software development phases) OR (software development lifecycle) OR (software planning) OR (software testing) OR (software analysis) OR (software maintenance) OR (software design) OR (software quality) OR software configuration management) OR (software validation) OR (software verification))) **AND** (Game Elements OR ((gamification) OR (gamifying) OR (gamify) OR (gamification mechanisms) OR (game mechanics))*

Then we started to conduct the review protocol by combining the major and alternative terms in popular databases such as IEEE Explore, ACM Digital Library, Çankaya University Electronic Database, etc. (see **Table *3***). As the last sub-step of *planning* phase, we set an evaluation protocol to end up first phase to eliminate the irrelevant results depending on the search strings.

- **Conducting**: It is the second step of the review which includes the following sub steps: Primary study selection, the data extraction, study quality assessment, and the data synthesis where the obtained data are synthesized. First, we made a pilot selection and extraction. Then we selected the primary

studies that are related to our research questions. We use the study selection policy for the data extraction process (see **Table *4***). After the study quality assessment, we started to data synthesis process. Then we finished the conducting step.

- **Documenting:** It is the last step of the review to report the dissemination of information by drawing a conclusion and considering threats.

Hence, the systematic literature review is used to review the primary studies to find out the answers about the research questions and observe the results. As Petersen [2] states that an SLR uses existing studies related to research topic describes the context and summarize the results. Kitchenham [6, pp. 3] explains the reasons why systematic literature review is conducted:

- "*To summarize the existing evidence concerning a treatment or technology e.g. to summarize the empirical evidence of the benefits and limitations of a specific agile method.*
- *To identify any gaps in current research in order to suggest areas for further investigation.*
- *To provide a framework/background in order to appropriately position new research activities."*

By this systematic literature review the existing proposals and research works for applying game elements into software development process in software development landscapes are determined, analyzed and classified to attempt to answer the questions and report them clearly for future research. While conducting this review, recommendations in [3, 17] are followed to make the review better and decide the best solution for the investigation.

This chapter continues by giving justification of research method in computer science and software engineering and describing search strategy and definition of research questions, identification of inclusion and exclusion criteria, extracting data and synthesis of the study.

## 3.2. Justification of Research Method

In this section, we will provide information about the systematic literature review in computer science and software engineering. We will describe the importance and the necessity of systematic literature review in computer science and software briefly.

### 3.2.1. Systematic Literature Review in Computer Science and Software Engineering

Systematic Literature Review or systematic review is a research method to identify, assess and evaluate the relevant primary research studies depending on the research questions of the related topics [31]. Thus, it aims to identify the existing evidences, provide background information to evaluate the exact data appropriately. In the earlier 90s, this method has been gained its popularity and started to use in medical research areas to validate the existing results in the scientific literature [46]. Hereby, this approach presented the concept of "Evidence Based Medicine (EBM)" and came into prominence. Kitchenham et al. [35] state that evidence-based research and results were firstly introduced in medicine fields because it needs scientific results and experiments rather than the people opinions. Also, the popularity and the usage of systematic literature review has been started to grow and use almost in all areas in order to validate the results. Kitchenham et al. [35] introduced the Evidence – Based Software Engineering (EBSE) in 2004 and the popularity of systematic literature review in software engineering has been grown to get more reliable results from the literature. Zhang and Babar [31, pp. 2] states that "*there are an increasing number of SLRs being performed in SE since 2004*." The reason for why popularity of SLRs in software engineering (SE) is growing is providing evidence – based results from the literature about the related topics. Kitchenham et al. [35] claim that software engineering field should adopt the evidence – based results due to growing number of studies in the scientific literature. Evidence Based Software Engineering helps to make accurate decisions about software engineering by collecting and evaluating empirical evidences from the studies depending on the research questions.

Therefore, the goal of a systematic literature review is not only providing results from the existing studies, it also aims to provide an evidence based guidance for the practitioners. Thus, to provide appropriate solutions to the software practitioners by supporting those with evidences, emerging EBSE with SLR provide more reliable results from the studies in the literature. Hence, providing appropriate solutions and results from the studies need well qualified systematic literature review process. A well-qualified systematic review should be conducted formal, controlled, planned and rigorous ways depending on the systematic review rules by the researchers. These ways and rules are explicitly identified in the systematic review guidelines in the literature. In support, Kitchenham et al. [47] state that empirical software engineering investigations need guidelines to provide strengthened, rich and reliable results. In software engineering research field there are lot of evidence based SLR studies in the literature [31]. These studies are conducted depending on the guidelines. These guidelines can help to improve the quality of currently conducted empirical studies and the existing studies. Kitchenham et al. [47] collect these comprehensive guidelines under six headings for systematic review in software engineering. These titles are [47, pp. 3]:

- *"Experimental context,*
- *Experimental design,*
- *Conduct of the experiment and data collection*
- *Analysis,*
- *Presentation of results, and*
- *Interpretation of results."*

Experimental Context is the most important part because it is used to make sure whether the goal of the research is appropriately defined or not and the definition of the investigation can provide an adequate information for other researchers. Therefore, it is most critical part in the software engineering research field [47]. In the experimental context there are three elements which are background information about current status of the research topic, discussion of the proposed solution and information about the related research.

In the experimental design, the main goal is ensuring the design is proper for the research goal by considering the products resources and processes. Kitchenham et al. [47] state that experimental design guidelines help to select what should be included (methods, techniques and treatments, etc.) according to needs when developing experimental design.

After making experimental design the conducting review and experimental data collection phase starts. In the conducting and experimental data collection phase, the outcomes of the experimental study should be considered according to the related research topic because of the replication of the proposed solution to the current problem. Thus, this is the main goal of the conducting and data collection for software engineering research. Kitchenham et al. [47] state that some problems occur in the data collection phase in software studies because of unclearly defined software measures. Also, Kitchenham et al. [49] discuss the current problems in data collection in software engineering in term of software measures and suggest several standards such as defining all software measures fully, attributes, units and rules that can others benefit from it when necessary. Therefore, it helps researchers whether to use current measures or define new measures in software measures. After conducting review and collecting experimental data phase, the analysis phase starts to observe the data for the investigation. Analysis guidelines aim to make sure the conforming of collected data to the study design. Kitchenham et al. [47] claim that there are two main approaches which are classical analysis and Bayesian analysis (using prior information) when analyzing data.

After making analysis for the collected data, presentation of results are needed. Everything about the review should be explained explicitly so as to make the reader understand the aim of the study, design, analysis and the results. Thus, it is also important as the analysis phase. Finally, a clearly defined conclusion should be needed for the researchers and practitioners by interpreting the results. Kitchenham et al. [47] explain the interpretation guidelines as defining type of study, specifying limitations of the study and defining statistics. After interpreting the results by

following the guidelines the systematic review process is finalized properly for software engineering research field.

In summary, we present the guidelines for the systematic literature review process and how they can be used in software engineering research. We conducted our systematic literature review by following the guidelines defined in [6, 47, 48, 49] to improve the quality of our study.

### 3.3. Research Questions

Depending on the systematic literature review principles [3], in this step, some research questions are established for this study to find answers from existing research work and proposals related to application of game elements to software development process in the context of software development landscapes. Research questions are related to applicability of the use of game elements in the context of software development landscapes and how these game elements affect the quality of software development in software development landscapes in order to overcome the challenges in the software industry. Different perspectives for applying game elements into software development landscapes are considered and associated with each research question separately. The research questions and their aims for this systematic review are described in **Table *1*.**

**Table 1:** Aim of the Research Questions for the Systematic Review

| ID | Question | Aim |
|---|---|---|
| RQ1 | What is currently known about people's motivation, engagement and performance issues in software development process? | To identify which factors affects the people's motivation, engagement and performance during the software development process. |
| RQ2 | Which game elements and gamification approaches can be applied into software development landscapes to increase the quality and performance of software development? | To determine which game elements and gamification approaches can be applied to software development landscapes to enhance the success stories by increasing the quality and performance of software development. |
| RQ3 | What are the examples of these gamification approaches for the software industry? | To show the applied gamifiaction approaches in the software industry. |
| RQ4 | How these gamification approaches affect the quality of software development in software business context? | To determine which gamification approaches remedy the software development process quality. |

These research questions are going to be guide for the data extraction phase and it helps to avoid reading full text of current research works related to the review. By the help of the research question that are stated above we focused on how game elements and mechanics remedy to increase the quality of software development process. During this review we reviewed the current research works and inspected how they focus on the gamification and software development process. Moreover, we observed other papers' proposals and approaches to find the efficient and effective solutions. For example, Research Question 1 (**RQ1**) focuses on investigation of identification for which factors affects the people's motivation and performance during the software development while Research Question 2 (**RQ2**) focuses on which game elements and gamification approaches or game elements

such as badges, points award mechanisms can be applied to software development landscapes to enhance the success stories and increase the motivation of the people of the software development process. In Research Question 3 (**RQ3**) we identified the real-life examples that adopts the gamification technique to their software development process. And finally, in Research Question 4 (**RQ4**) we observed how these gamification approaches affect the quality of software development process in the industry. To investigate these research questions to get the expected results we developed a review protocol including our search strategy, data source, study selection criteria and procedures. Thus, we will continue with describing these titles in the remaining parts of this section.

### 3.4. Search Strategy and Data Source

To get a clear picture of current research and to find the answers to research questions, popular and reliable data sources are used to carry out this review. The search strategy included the academic electronic databases and Google Search is used for including extra publications and contributions for this study. The following academic electronic databases are used in this review:

- Çankaya University Electronic Database
- IEEE Explore
- Science Direct
- Access Engineering
- ACM Digital Library
- SpringerLink

Some special combinations, keywords and search strings are designed to conduct this review by using these electronic data sources. The following keywords are used as major terms to design the search strings: Software Development Process and Game Elements. These search strings are designed according to steps which are stated in [18] and all these terms related to this review topic is combined using Boolean AND or OR operator to retrieve the articles that are only related to this topic.

The **Table *2*** indicates the search strings that are used in this review study.

**Table 2:** Search strings for this Systematic Literature Review

| **Major Terms** | **Alternative Terms** |
|---|---|
| Software Development Process | ((software development) OR<br>(software engineering) OR<br>(software process) OR<br>(software development phases) OR<br>(software development lifecycle) OR<br>(software planning) OR<br>(software testing) OR<br>(software analysis) OR<br>(software maintenance) OR<br>(software design) OR<br>(software quality) OR<br>(software configuration management) OR<br>(software validation) OR<br>(software verification)) AND |
| Game Elements | ((gamification) OR<br>(gamifying) OR<br>(gamify) OR<br>(gamification mechanisms) OR<br>(game mechanics)) |

These keywords and search strings are built up based on the research questions to get as many papers possible. Also, the alternative keywords and terms are added to major terms (shown in **Table *2***.) to retrieve and cover more papers from the stated electronic databases. These search strings are applied to title, abstract and keywords to get the results.

The **Table *3*** shows the search strategy which is described above.

**Table 3:** Search Strategy

| | | |
|---|---|---|
| Data Source | Academic Search: | • Çankaya University Electronic Database<br>• IEEE Explore<br>• Science Direct<br>• Access Engineering<br>• ACM Digital Library<br>• SpringerLink |
| | Non – Academic Search | • Standard Google Search Engine |
| Items | | • Academic Papers<br>• Academic Publications<br>• Journals, Book Sections<br>• Conference Papers<br>• Online academic publications |
| Search Applied to | | • Title, Abstract, Keywords |
| Language | | • Only written in English |
| Publication between | | • 2010 – 2017 |

### 3.5. Study Selection Criteria and Procedure

Study selection criteria describes the inclusion and exclusion standards whether to include or not the existing research works and proposals depending on the research questions, search string and strategy when conducting a systematic literature review. Therefore, a search result must meet the constraints that are defined in inclusion and exclusion standards. Thus, the inclusion standards include the search result whereas the exclusion standards eliminate the search result in the review.

The inclusion and exclusion criterion need to be defined to increase to more reliable results and decrease the risk to stray away from the topic. In other words, study selection criteria are designed and used for processioning for the review. Thus, this enables other researchers to rework by carrying out review using the same standards. When defining the study selection standards some criterion should be followed. The inclusion criteria should not be too general or too strict. When it is too general, the poor quality studies may be included and it affects the final results. On the other hand, when it is too strict, the entire related studies might not be presented in the

final result. The inclusion standards are defined to get clear and reliable results. In our study, study selection process was executed in two phases. In the first phase, we read the titles and abstracts of the papers that we accessed depending on our search strings, the search results including irrelevant data are excluded for this systematic review Then, in the second phase, we observed the remaining papers by reading its introduction and conclusion parts to get the target data that meets the inclusion and exclusion criteria. Then, studies are eligible for inclusion in the review if they presented empirical data on software development that are used game elements in their business context. In addition, the studies which are related to human factors (e.g. user motivation, engagement and collaboration, etc.) in software development are included to study selection criteria. Therefore, only academic papers and publications, professional forums and contributions such as conferences, online publications and book sections which related to this review domain are included to inclusion standards. Moreover, to get the current information, publications between 2010 and 2017 and only the paper which are written in English is considered as the part of inclusion standards. Also, non–academic results, people opinions, personal blogs and personal web pages from the search results are excluded for this study. The review also excluded the academic results that meet the following criterion: duplicate papers that are retrieved from different databases, the papers available with only its abstracts, papers related to gamification of other domains (marketing, health, etc.)

The **Table *4*** summarizes and presents the study selection criteria that are used in the review process.

**Table 4:** Study Selection Criteria

| | |
|---|---|
| Inclusion Criteria | • Results including the relevant data<br>• Academic Papers<br>• Academic Publications<br>• Journals<br>• Book Sections<br>• Conference Papers<br>• Online academic publications<br>• Papers written in English<br>• Publications between 2010 and 2017. |
| Exclusion Criteria | • Main focus is not related gamification in software development process and human factors in software development process.<br>• Papers available only with its abstracts<br>• Duplicate papers retrieved from different databases.<br>• Non – academic publications<br>• Magazines, Personal web pages and blogs, Personal opinions. |
| Exclusion Criteria<br><br>for full text | • Papers related to gamification of other domains (education, health, etc.)<br>• Papers included the summary of conference notes. |

### 3.6. Data Extraction

This is the 7th step of the Systematic Literature Review process. (shown in **Figure *3***) In this phase the research papers are determined according to the study selection criteria. We have filtered a large amount papers based on the study selection criteria which is described in *Study Selection Criteria and Procedure* section. At the end, we ensured that which paper is eligible for full text reading. The data extraction form is generated to make this review easy to watch and more reliable. Also, the data extraction form is used by authors to conduct this review in an efficient way. In the data extraction form we assigned a property to each research question to reach the

fruition and get the expected results from this review. These properties and research questions are shown in the **Table *5***.

**Table 5:** Properties of Each Research Question

| Property | Research Question |
|---|---|
| **Property 1**<br>Human Factors in software development. | RQ1 |
| **Property 2**<br>Social effects of people in software development | RQ1 |
| **Property 3**<br>Gamification of software development | RQ2 |
| **Property 4**<br>Real World applications of gamification for software development | RQ3 |
| **Property 5**<br>Effects of gamification on software development | RQ4 |

*Property 1* and *Property 2* are assigned to get previous knowledge of peoples' social factors such as motivation, engagement and collaboration. Then, *Property 3* is assigned to get information about game elements and which game elements can be applied to the software development to increase the quality of software development process. *Property 4* is defined to access the studies that are related to real life examples that the gamification concept is currently applied. *Property 5* is created determine how the application of gamification into the software development process affect the process.

### 3.7. Evaluation

After the data extraction period, a data extraction evaluation process was conducted for the quality assessment of the studies. The purpose of this activity is defining the low-quality publications related to the purpose of the paper which is explicitly defined in the Introduction part of the paper. A five-point Likert-scale questionnaire was designed for measuring the quality of the studies that were chosen depending on the study inclusion and exclusion criteria and contributions to the research. These reviewers consist of 4 people in our company with different roles in our software development team. Reviewer 1 is the project manager of the software development team and the Reviewer 2 is the junior software developer and Review 3 is the senior software developer and the Reviewer 4 is the senior software tester in the company. To complete the evaluation of the studies took two weeks and the result are shown in the **Table 7**. Main goal of asking the evaluation questions to the reviewers is that indicating the agreement level to show the study meets the research objectives or not. Each answer of corresponding question has a ranking scale between "Strongly Agree (5)" and "Strongly Disagree (0)". Therefore, each paper evaluation result has a numerical evaluation value as a result of an arithmetic calculation between 0 – 5 that indicates the agreement degree. This degree considered as the quality contribution value to the study. However, if the degree of the evaluation is low, this does not indicate that we should exclude the paper from the study.

The **Table *6*** shows the evaluation questions. In this table, each evaluation question is indicated with an identification number in the ID column. These evaluation questions are defined for the assessment of quality of each study in terms of the motivation of the study whether clearly explained or not, evidences of the results for the contributions to the objective of the research questions.

**Table 6:** Evaluation Questions

| ID | Evaluation Question (EQ) |
|---|---|
| EQ1 | Does the motivation of study clearly explain the problems on the human related issues to adopt the gamification in the software development landscapes? |
| EQ2 | Does the study present the satisfying evidences to attempt to integrate the game elements and gamification concept into the software development landscapes? |
| EQ3 | Does the proposal that they provide have positive effects on the human related social issues in software development process? |
| EQ4 | Does the approach that they present have positive effects on the performance of all stakeholders and quality of work during the software development process? |

**Table 7:** Evalutation Results from the Reviewers

| ID | Reviewer 1 | Reviewer 2 | Reviewer 3 | Reviewer 4 |
|---|---|---|---|---|
| EQ1 | 5 | 5 | 5 | 5 |
| EQ2 | 5 | 5 | 4 | 5 |
| EQ3 | 5 | 5 | 5 | 5 |
| EQ4 | 4 | 4 | 5 | 4 |

We have conducted this evaluation process to assess the quality of the studies to show the study meets the research goals or not as intended at the beginning of this paper. We have asked four questions to the four reviewers (see **Table 7**) to evaluate the studies. As the evaluation assessment indicates that from the **Table 7** all reviewers are strongly agree for the EQ1 and EQ3. For the EQ2 only the Reviewer 3 gave 4 points and the other reviewers are strongly agree for the EQ2. The Reviewer 4 is only strongly agree for the EQ4 and Reviewer 1, 2 and 4 are gave 4 points which indicates the agree according to the five-point Likert-scale questionnaire accumulation. Depending on the evaluation results from the reviewers, this evaluation process provides us to make quality assessment for the studies and indicate the low – quality publications. Therefore, the overall results from the reviewers show that we have observed and analyzed the appropriate studies for our research goals.

### 3.8. Summary

In summary, in this chapter we provide information about systematic literature review in general and justification of research method about systematic literature review in computer science and software engineering. First, we present our research questions to investigate the related studies about our research objectives. Then, depending on the systematic literature review principles [3, 17] we design and describe our and search strings, search strategy and data sources. After that, we define our study selection criteria and execute the study selection procedure for our systematic literature review. Then, we collect some data and conduct a data extraction period. Then, we conduct an evaluation process for quality assessment of the obtained data from the studies to indicate the agreement level to show the study meets the research goals or not. After following these steps, we continue by analyzing and presenting the result of obtained data. In Chapter 4, we present the analysis results of these studies to investigate our research questions and give a brief summary for the results.

# CHAPTER 4

# RESULTS

In this chapter, the obtained data will be presented. The results from the review process will be given according to each research question.

## 4.1. Overview the results

We made our research by using systematic literature review methodology as following the recommendations that are explained in [3, 17]. First, we developed search strings, then identified data sources and finally decided the study selection criteria which are described in CHAPTER 3 in detail. By following these search strategies, we accessed and examined a lot of research work, extracted irrelevant data related to the topic and collect the results to address the research questions. The **Table *8*** shows the search results of each steps of the review process. As a result, number of 32 studies were obtained and these primary studies are listed in Appendix A section.

**Table 8:** Results Obtained

| Step | Process | Number of Papers |
|---|---|---|
| **Step 1** | Search, obtain data | 1093 |
| **Step 2** | Data extraction, reading keywords, title or abstract parts | 84 |
| **Step 3** | Reading full text paper | 59 |

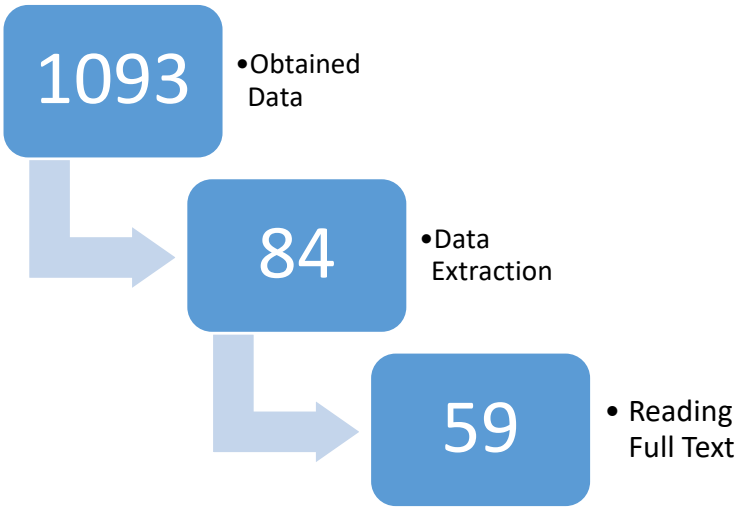


**Figure 4:** Summary of the Obtained Data

***Figure 4*** summarizes the results of the obtained data and the steps that we followed for the elimination process to get studies that are indicated in Appendix A. First, we made a research by using our search strategy, search strings and data sources as we previously explained in the previous sections of the study. We used academic databases as well as Standard Google Search Engine as the data sources. We applied our research to the titles, abstracts and keywords. Then we got 1093 papers according to our search results. Then we applied our study inclusion and exclusion criteria to the obtained search papers. We included only the papers written in English language. We set publication year limit to include the studies that presents current data into our systematic review. The papers published between 2010 – 2017 are included to our study. The ***Figure 5*** shows the distributions of the papers by year.

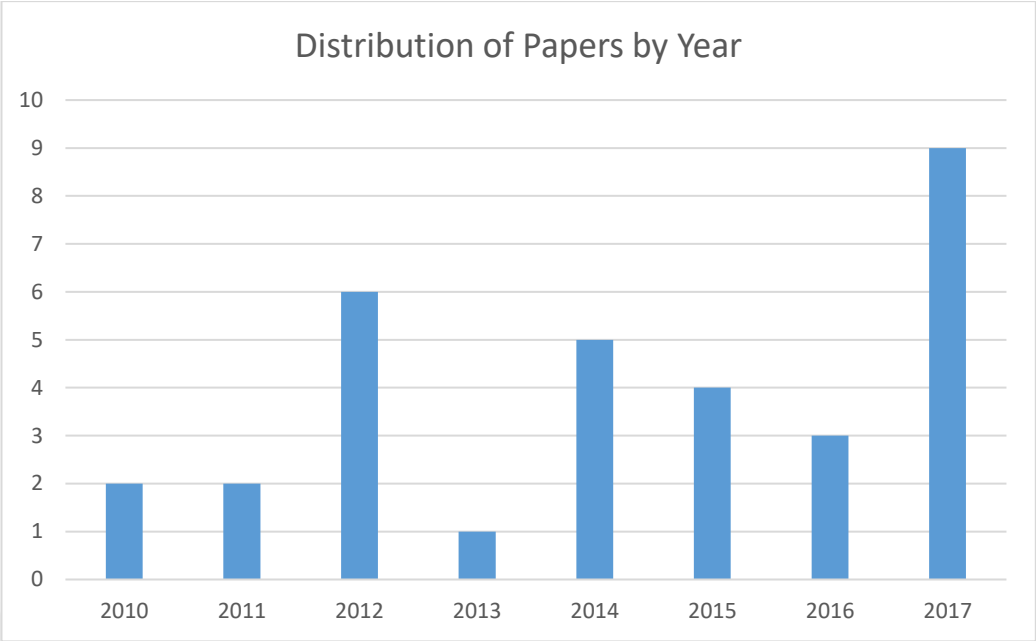


**Figure 5:** Distribution of Papers By Year

We chose academic papers, academic publications, journals, book sections, conference papers and online academic publications as the target items.

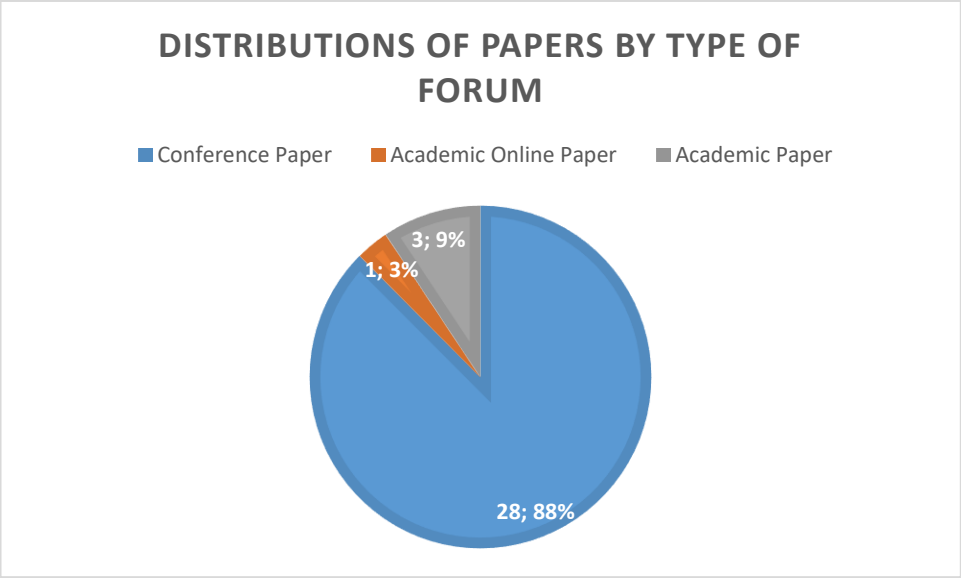


**Figure 6:** Distributions of papers by the type of forum

Then, we eliminated the duplicate papers and the studies that we could access the abstracts of the papers.

The ***Figure 7*** shows the data source distribution of the studies.

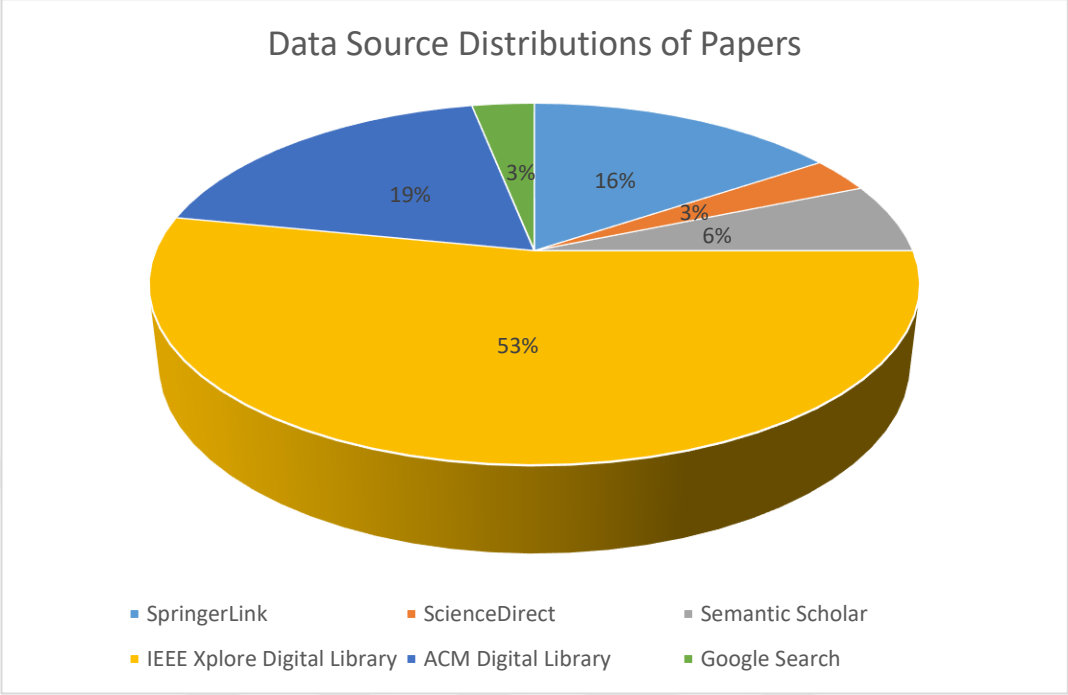


**Figure 7:** Data Source Distribution of Papers

### 4.2. Analysis Results of Research Questions

In this part, we made an analysis using the data that we obtained while conducting this review to answer the research questions by following the study selection criteria.

#### 4.2.1. Evaluation

For the quality assessment of the study we presented the evaluation questions in 3.7 Evaluation section. A five – point Likert scale is used to assess the quality of the questions. Each evaluation question has a ranking scale between “Strongly Agree (5)” and “Strongly Disagree (0)” to indicate the agreement. The ***Table 7*** indicates the evaluation results from the reviewers.

After that, we identified the studies that meets our research questions. The **Table *9*** shows that the appendix mappings in which paper we found the answers to our research questions.

**Table 9:** Appendix Mapping of the Studies to Each Research Questions

| **Research Question ID** | **Appendix Number** |
|---|---|
| **RQ1** | [A3], [A5], [A6], [A11], [A12], [A16], [A22] |
| **RQ2** | [A3], [A4], [A5], [A6], [A9], [A13], [A16], [A28], [A31] |
| **RQ3** | [A3], [A4], [A9], [A11], [A13], [A16], [A31] |
| **RQ4** | [A1], [A2], [A3], [A4], [A5], [A6], [A7], [A8], [A9], [A10], [A11], [A12], [A13], [A14], [A15], [A16], [A17], [A18], [A19], [A20], [A21], [A22], [A23], [A24], [A25], [A26], [A27], [A28], [A29], [A30], [A32] |

We try to investigate the ***Research Question 1 (RQ1)*** depending on our findings. Therefore, we try to cover current states of people's motivation, engagement and performance issues and what causes to problems related to these issues in software development process. As previously mentioned that software development life cycle consists of interrelating activities in which human actively takes part in and plays a crucial role.

The first and the most important step of the software development life cycle is requirement elicitation process. Therefore, human related issues such as motivation, collaboration and communication are the important factors that may affect the elicitation process negatively and so on. Hence, Fernandes et al. [A3] state the currently known common challenges such as lack of user involvement, collaboration and communication problems among stakeholders in software requirement elicitation process and developed a gamified environment which is *iThink* to tackle with the user involvement and collaboration issues. Also, Duarte et al. [A12] state that collaboration, communication and low motivation issues as the currently known problems in requirement elicitation process. Hence, they made their research depending on this point view. They intended to increase the motivation and involve

the users by making a collaborative requirement gathering environment by supporting visualization techniques. Moreover, Kolpandinos and Glinz [A22] pointed out the current motivational issues among stakeholders in actual requirement elicitation process. Therefore, in their study they described three dimensions of motivations (skill acquisition, player type development and need satisfaction) and created an example with respect to gamification to increase the motivation of outside organizational stakeholders during the requirement elicitation process.

As stated above, software development lifecycle consists of interrelating activities and human factors play a crucial role in each step. In the implementation phase, human factors are important also. These factors may also affect the software productivity. Therefore, Melo et al. [A11] observed the collaboration issues to increase the productivity. They proposed a version control system gamification to increase the collaboration and engagement in software projects.

In the software testing phase, people involvement is highly required inevitably. Therefore, motivation and performance issues should be considered in software testing process. Hence, Fraser [A16] conducted their study considering the motivation and performance issues and proposed gamification to solve the currently known human related problems in software testing.

As we consider the software development lifecycle as a whole process, motivation, engagement, collaboration and performance issues are seen as impairing human factors to software development process. Therefore, Sasso et al. [A5] conducted their study by considering the motivational issues in software engineering. They investigated that how gamification can be applied into software engineering to resolve the human related obstacles. Dorling and McCaffery [A6] presented low motivation reasons in software process improvement. Therefore, they offer gamification as a solution to increase the motivation of people in the organizations. Thus, according to the analysis that we made based on the RQ1, we try to identify what currently is currently known about the people's motivation, engagement and

performance issues in software development process. Then, we examined the studies to determine what causes to these issues. Lack of communication, lack of performance evaluation and rewarding affect the people's motivation, engagement and performance issues during the software development process.

We choose the ***Research Question 2 (RQ2)*** to investigate how game elements and gamification approaches increase the quality and performance of software development. First, we try to identify the most commonly used game elements and how these game elements can be applied into the software development process. **Table *12*** and ***Figure 8*** show the distribution of game elements in our studies. Most commonly used game elements that we found in the studies are:

- Points
- Badges
- Leaderboards
- Rewards
- Voting
- Ranking
- Feedback Mechanism
- Quests
- Avatar
- Levels

The aim of these studies is that increasing the quality and performance of software development. Thus, our selected studies are related to tackle with the human related problems. As we realized from the studies, people's extrinsic and intrinsic motivations are affected positively by using these game elements. For example, Fernandes et al. [A3] developed a gamified tool based on giving points by mapping each thinking hat to motivate the users. Lombriser and Valk [A4] observed the real – life examples of gamification which are Visual Studio Achievements, Microsoft Quality Language Game depending on badges and leaderboards to engage, collaborate and motivate the users during the implementation and testing stages within the software development to enhance the quality and performance of the software development process. Passos et al. [A9] conducted a case study in a real-life software development environment to encourage the users by using feedback

mechanism, rewards and experience points to measure the individual and team achievements.

Arai et al. [A13] proposed a gamified tool (which is named *GBC)* to motivate the users by giving points depending on their actions to reduce the bug warnings. They aim to increase the quality and performance of software applications by motivating the users. Unkelos-Shpigel and Hadar [A28] developed a gamified tool based on cognitive principles to collaborate the software practitioners on code review issues. Both programmers and reviewers earn points and rewards in order to encourage, collaborate and guide productive behaviors on code review process. Thus, they aim to increase the quality of work by encouraging users and changing their behaviors. Lotufo et al. [A31] investigated the game elements and mechanisms in bug tracking systems. They investigated that how rewards, points, voting, ranking and reputation privileges affect the motivations, contributions and quality of the solutions to the problems. Also, they observed game mechanisms in *StackOverflow* and their effects to enhance the quality of work.

To summarize, we choose the **RQ2** to observe the effects of game elements and gamification approaches to increase the quality and performance of the software development. We examined the studies depending of this point of view and we classify the results.

We choose the **RQ3** to analyze and observe the examples of the gamification approaches for the software industry. We examined the studies by focusing on the real-life examples and case studies where the gamification approaches are used to overcome the human related problems in the software development landscapes. In our selected studies, researchers discuss the current problems and observe the real – life gamification approaches in the software development landscapes from different perspectives of software development process. For instance, Passos et al [A9] gives RedCitter Tracker system as an example to motivate and encourage the users by giving rewards and badges in task management domain. Melo et al. [A11] observed

the Yahoo Answers and StackOverflow to show how these applications are used to motivate and encourage their customers by using gamification approach. Lombriser and Valk [A4] analyzed the CodeHunt, Visual Studio Achivements and Microsoft Language Quality Game to show the positive effects of the gamification on the software development and testing phases. Thus, by investigating the RQ3 we observed that how gamification approaches are used in real-life software development process. Also, we examined the intended outcomes of gamification approaches in the real – life software development processes.

We choose the **RQ4** to determine which gamification approaches increase the software quality in the software development context. Therefore, we included the papers that present the research papers, real – life examples, experiments and case studies for our review. The aim of the selected studies is resolving human related issues in software development process. For instance, Yilmaz et al. [A1] proposed a 3D gamification approach to explore personality types of software practitioners to team up with well-adjusted people. Kosa and Yilmaz [A2] suggest that building game - based environments in software development is an effective solution to overcome the human related social issues in software development process. Yilmaz and O'Conner [A8] [A10] have made a research to understand the effects of the social factors for software productivity and increase software productivity through economic mechanism design. Dorling and McCaffery [A6] observes the applicability of gamification to the software development process and its positive psychological effects. Herranz et al. [A7] consider the critical success factors for the software process improvement and present the gamification as a solution. Rojas et al. [A30] conducted an experimental study to validate the positive behavioral effects of gamification and reported the expected results in software development context. Steffens et al. [A20] conducted a qualitative study to change the behavior of software actors by making collaborative environment and affecting their motivation.

We also analyzed the applicability of gamification into software engineering education. Akpolat and Slany [A14] conducted a case study in a university environment to enhance the engagement of software engineering students. Uskov and

Sekar [A15] proposed a gamification approach to improve the skills of the students by increasing the engagement, collaboration and motivation through gamifying software engineering curriculum in universities. Singer and Schneider [A26] also built a gamified tool in a university environment for version control systems to make students more frequent check – in to the repositories. Thus, we choose RQ4 to determine how these gamification approaches remedy the quality of software development process from different points of views such as behavioral and psychological contributions to the software development process by resolving the human related factors.

Based on this evaluation we have a chance to characterize and classify the papers to observe them detailed to investigate our research questions. Therefore, in the remaining part of this section, we will describe the results for each question in detailed. **Table *10*** shows the type of software development process areas that gamification approaches are applied in the selected studies.

**Table 10:** Type of Software Development Process Areas

| Software Development Process Areas | Appendix |
|---|---|
| Requirement Elicitation and Analysis | [A3] [A12] [A18] [A22] |
| Software Development | [A1] [A2] [A4] [A8] [A9] [A10] [A23] [A24] [A20] [A32] |
| Agile Software Development | [A17] [A27] |
| Software Testing and Bug Tracking | [A4] [A5] [A13] [A16] [A21] [A30] [A31] |
| Software Project Management | [A9] |
| SCRUM | [A19] |
| Software Process Improvement | [A6] [A7] [A29] |
| Code Review | [A5] [A28] |
| Version Control Systems | [A11] |
| Software Engineering Education | [A14] [A15] [A25] [A26] |

#### *4.2.1.1.* Research Question 1 (RQ1): *What is currently known about people's motivation, engagement and performance issues in software development process?*

Groh [33] states that the aim of using gamification in the various domains is that increasing users' motivation, engagement and collaboration to get the higher quality productivity and performance. People's motivation is a key factor of the software development process. If there is a low motivation, lack of user involvement engagement and collaboration occur during the development process. Also, lack of communication between stakeholders affects the user involvement in the software projects. Viskovic et al. [41] state that communication is highly important in IT projects and the lack of communication is the reason of failure of a project. Since user involvement and communication plays a crucial role in software development process, the lack of user involvement affects the software development process in a negative manner [37]. Efficient and effective user involvement and user engagement requires high motivational aspects to increase the performance issues related to human factors. Hence, it can be said that the main focus is human related factors and its effects on the software development process steps while conducting this review.

Fernandes et al. [A3] proposed a gamified tool which is called *iThink* for the requirement gathering phase of the software development lifecycle. They proposed this tool to increase the user involvement by enhancing the user motivation and engagement. This tool was developed based on the game elements which are points, rankings and rewards. Users can earn points for each contribution (defining a new requirement, asking, discussing, etc.) and these points are shown in the scoreboard so as to increase the user involvement. Moreover, they tested this tool by conducting two case studies and observed that user motivation, engagement and performance are increased prominently. Duarte et al. [A12] claim that an explicit analysis is a key factor when developing a software product. Therefore, they also proposed a collaborative environment for team members and stakeholders to produce a well-understood software product. They used *badges* and *ranking systems* as the game elements in their collaborative environment. Both in [A3] and [A12] researchers

indicate that rewarding mechanism affects the software development process due to improving motivation and engagement of users.

Kolpondinos and Glinz [A22] proposed and developed another gamified tool for the stakeholder engagement to involve the outside stakeholders in the requirement elicitation process efficiently. They applied this gamification tool to the Garuso (Game – Based Requirement Elicitation) project. They developed for this system to motivate and engage the users by using rewarding element of gamification. They claim that by giving rewards to users based on their contributions, gamification might be a solution to increase the user involvement in the requirement elicitation process of software development process .

Sasso et al. [A5] built a gamification framework (based the Taje's layered game design principle) to develop two gamified environments to increase the user motivation, performance and collaboration. They exemplified this framework in bug tracking and code review issues. They use levelling mechanism to increase the performance and motivation of the users. After evaluating two examples they observed that the performance and motivation of users are affected positively. Arai et al. [A13] proposed a gamified tool which is named *GBC* to motivate the users to detect the bugs in software applications. Each user gets the points regarding the *rank* of each bug issue. They carried out an experiment to validate their proposed tool and they satisfied from the overall results. They claim that gamification of bug removing tool may be effective for motivating the users.

As previously mentioned, user engagement in software development is highly important to finalize the process successfully. Passos et al. [A9] present a case study by developing a gamified tool to make the software development process as a funny activity. In this proposal, they aimed to give real – time feedbacks to the users to increase the user engagement by competing with others in software development process. They claim that their tools increase the user engagement and performance. Dorling and McCaffery [A6] state that gamification may be a solution for software

process improvement by encouraging engagement. Melo et al. [A11] proposed the version control system gamification to encourage the engagement of software developers in software projects. In this system, the user engagement is calculated based on the metrics and sent to gamification server. Users' performance is announced on the leaderboard to encourage the engagement. They claim that their proposed theory increases the user engagement. Fraser [A16] examined the existing and potential gamified software testing tools such as, Stash Badgr, Visual Studio, CodeFight, The Jenkins CI Game, etc. to engage the users based on the game elements such as points, rewards and badges. Fraser [A16] reports that gamification of software testing may be beneficial for increasing the motivation and engagement.

To summarize, we chose this research question to identify what is currently known about people's motivation, engagement and performance issues and which factors affect these issues during the software development process. We observe that lack of communication, lack of performance evaluation, lack of rewarding mechanism and real – life feedbacks affect the people's motivation, engagement and performance during the software development process. The obtained data from the studies that we examined shows that using gamification as solution offers to solve the human related factors by improving the user motivation, engagement and increasing the performance of software development process.

#### *4.2.1.2.* Research Question 2 (RQ2): *Which game elements and gamification approaches can be applied into software development landscapes to increase the quality and performance of software development?*

We use this research question to determine which game elements and gamification approaches can be applied into the software development process in software development landscapes to enhance the success stories by increasing the quality and performance of what kind of software development.

There are several software development types depending on their domains. The examples of software development types are: application development, system development, embedded systems development, web development, software tools development, scientific development, etc. These types of software developments are chosen to produce software products according to their goals. For example, web development is chosen for developing software products which run on web browser or embedded system development is chosen for designing and developing software product which run on non – computer devices. Therefore, in this research question we also try to investigate the applicability of gamification in any kinds of software development by considering the gamification types. According to the Kapp [7] there are two gamification types which are structural and content gamification. In structural gamification type the content of the structure does not change. It is a way of adding game elements and mechanics to gamify content. Structural gamification uses points, badges, rewards and leaderboards to increase the motivation, engagement and collaboration of the users [7]. Content gamification is a type which is turning into a game without designing a game. It uses challenges, feedback mechanisms to increase engagement of the users. Thus, we also analyzed the studies according to these gamification types to determine which gamification type is applicable to what kind of software development.

Kosa and Yilmaz [A2] observed game-based approaches for software engineering education and software development process. They presented the studies which are related to structural and content gamification types for the different software development types such as systems development, applications development, scientific development, web development. Fernandes et al. [A3] proposed a game-based approach for the requirement elicitation for *systems development*. In this tool points and progress bar are used. Therefore, structural gamification type is used in this tool to improve the collaboration and participation of users. Lombriser and Valk [A4] gives Visual Studio Achievements and Code Hunt as areal – life examples for the software development. Visual Studio Achievements uses feedback mechanisms to teach coding so it uses content gamification type for the *application* and *web development* type. Also, Lombriser and Valk [A4] gives Microsoft Language Quality Game example which uses points, leaderboards, levels to motivate users for the testing activity.

Schäfer [A19] presents lessons learned after two teaching periods with training Scrum course. It is a content gamification of Scrum course for teaching. Rojas et al. [A30] proposed a gamified tool which is named CodeDefenders for producing software tests. It is the content gamification type for *software tools development*. Rojas et al. [A30] claim that users can produce stronger test by using CodeDefenders because it increases users' motivation and engagement. Thus, after analyzing these studies, we investigated that what kind of gamification is beneficial for what kind of software development.

The **Table *11*** presents some of the findings according to our analysis to show what kind of gamification is used in different software development types.

**Table 11:** Gamification Types and Software Development Types

| Gamification Type | Software Development Type | Appendix |
|---|---|---|
| Structural Gamification | Systems Development | [A2] [A3] [A11] [A22] [A25] [A26] [A27] [A28] [A29] [A32] |
| | Application Development | [A2] [A11] [A18] [A22] [A24] [A25] [A26] [A27] [A28] [A29] [A32] |
| | Web Development | [A2] [A11] [A18] [A22] [A25] [A26] [A27] [A28] [A29] [A32] |
| | Software Tools Development | [A11] [A13] [A16] [A21] [A22] [A26] [A27] [A28] [A29] [A31] [A32] |
| Content Gamification | Systems Development | [A2] [A9] [A15] [A25] [A32] |
| | Application Development | [A2] [A4] [A9] [A15] [A18] [A24] [A25] [A32] |
| | Web Development | [A2] [A4] [A9] [A15] [A25] [A32] |
| | Software Tools Development | [A9] [A15] [A16] [A19] [A21] [A30] [A32] |
| | Scientific Development | [A2] [A9] [A15] |

We observed that structural and content gamification have positive impacts on different types of software development in terms of increasing motivations, engagements and collaboration in software development landscapes.

We also classified and examined the primary studies that are about using game elements and mechanics. High-level motivation of team members causes to increase the quality of software development. Herranz et al. [A7] suggest that using game elements (rankings, badges, etc.) strengthens motivation of people; thus, it fosters to increase the quality and performance of software development. Human related issues affect the quality and performance of development process. Yilmaz and O'Connor [A10] proposed that using game elements as a solution to fix the human related obstacles maximizes the quality of development process. Moreover, according to the analysis [5, 21, 22] that we made, we identify the following game elements:

[A1], [A2] [A4], [A8], [A11] that may be applied into the software development process to increase the quality and performance of software development process.

- Point
- Badges
- Leaderboards
- Rewards
- Voting
- Ranking
- Feedback Mechanism
- Quests
- Avatar
- Levels

The **Table *12*** and ***Figure 8*** show the distribution of game element in our studies.

**Table 12:** Game Elements in the Papers

| **Game Element** | **Appendix Number** |
|---|---|
| **Points** | [A1], [A2], [A3], [A4], [A6], [A9], [A13], [A16], [A17], [A18], [A21], [A22], [A23], [A24], [A25], [A26], [A27], [A28], [A29], [A30], [A31], [A32] |
| **Badges** | [A1], [A2], [A3], [A4], [A6], [A9], [A11], [A13], [A14], [A16], [A18], [A20], [A21], [A22], [A23], [A24], [A25], [A26], [A27], [A29], [A31], [A32] |
| **Leaderboards** | [A6], [A9], [A11], [A16], [A18], [A20], [A21], [A22], [A25], [A26], [A28], [A29], [A31], [A32] |
| **Rewards** | [A3], [A4], [A6], [A9], [A11], [A12], [A16], [A18], [A20], [A21], [A22], [A23], [A24], [A25], [A26], [A27], [A28], [A29], [A30], [A31] |
| **Voting** | [A3], [A12], [A31] |
| **Ranking** | [A3], [A12], [A22], [A26], [A30], [A31] |
| **Feedback Mechanism** | [A4], [A9], [A11], [A16], [A20], [A21], [A25], [A27], [A29], [A32] |
| **Quests** | [A4], [A9], [A16], [A23], [A25], [A27], [A30], [A32] |
| **Avatar** | [A4], [A26] |
| **Levels** | [A1], [A2], [A3], [A4], [A6], [A9], [A18], [A20], [A22], [A23], [A24], [A25], [A27], [A29], [A30], [A31], [A32] |

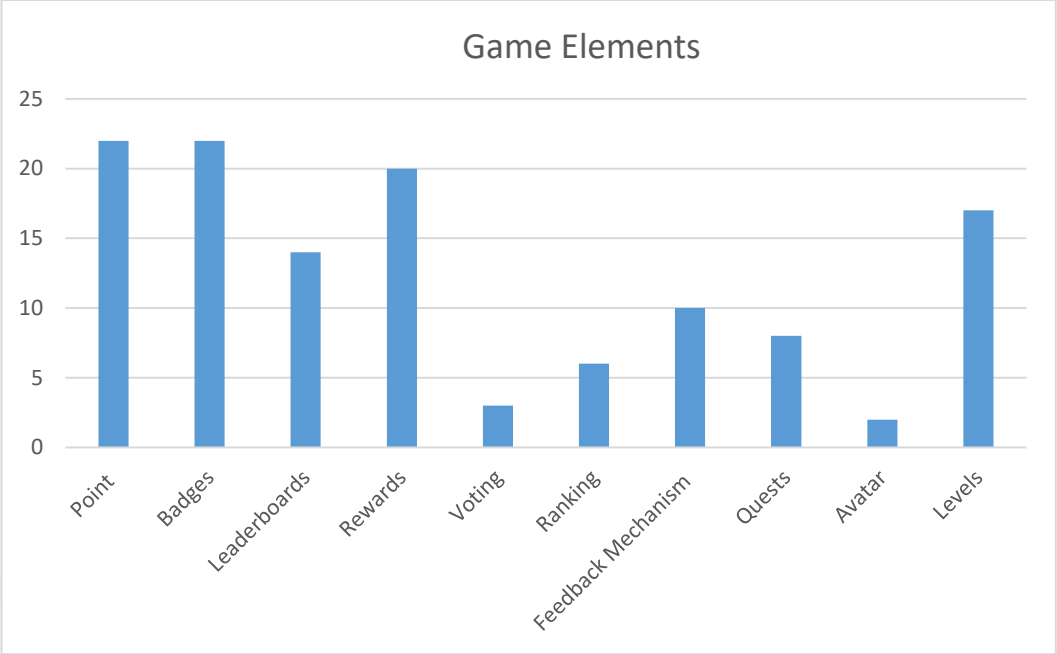


**Figure 8:** Number of Game Elements in the papers

#### 4.2.1.3. Research Question 3 (RQ3): *What are the examples of these gamification approaches for the software industry?*

We choose this research question to show and exemplify the applied approaches in the software industry. In this research question, we focused on the real-life examples in which gamification approach is used.

Passos et al. [A9] give RedCitter Tracker system as a real-life example that gamification is used in the software development domain. This is a gamified agile project management tool that uses rewards and badges as game elements. In this tool, users get badges according to completion of their tasks. These badges are displayed on the user profile. Reward mechanism is another game element in this tool. Users are rewarded with the points depending on task completions. These points can be spent on the reward store of the company by the users. This tool aims to motivate and engage users by using gamification mechanisms. Melo et al. [A11] state that SAP Company has started to use gamification concept in their software development team

to increase the quality and performance of people. Melo et al. continue by giving real – life gamification approaches in the software industry. They state that Yahoo Answers and StackOverflow uses the gamification concept to motivate and engage their customers. Both in Yahoo Answers and StackOverflow, ranking and rewarding mechanism is used according to specific user actions.

Fernandes et al. [A3] presented an online gamified tool which is called *iThink* to conduct the requirement elicitation process. This tool uses reward mechanism, points and ranking mechanism as game elements. This tool is the combination of these gamification elements and creative parallel thinking technique which is called "The Six Thinking Hats" [39]. *iThink* is developed to avoid the conflicts among team members and stakeholders. Also, it aims to increase the motivation, user involvement and collaboration. The *iThink* users get points based on their actions in the requirement elicitation process. These actions can be a declaration of a new requirement or resolving conflicts in requirement definitions, asking questions or adding new comments on the requirement definitions or discussing/rating the existing requirements. Based on these actions users get points and these points are displayed on the progress bar to show the accomplishments of the users. Fernandes et al. [A3] conducted two case studies to evaluate this gamification approach and presented the results of the case studies. Results show that *iThink* is very beneficial for the requirement elicitation process and has positive impacts on people such as increasing motivation and collaboration. Project managers and project owners are satisfied from the results since the quality of work is increased by using this gamified approach.

Lombriser and Valk [A4] give the real – life examples by analyzing the CodeHunt and Visual Studio Achievements (for software development) and Microsoft Language Quality Game (for testing). CodeHunt which is released by Microsoft Research for the educational purpose. It aims to teach coding by encouraging the users. The CodeHunt uses feedback mechanism to make the users more motivated and engaged. It uses puzzles which must be solved by an appropriate written code. Therefore, when the piece of code is correct the puzzle is solved successfully and

user get quests. The user continues the challenge by playing next level to improve her coding skills. In each upper level the difficulty of solving puzzles increases and it becomes more challenging for the users. Hence, it is observed that using CodeHunt as a gamified concept for the software development educational purpose has positive impacts on the improving coding skills by motivating and engaging participants. Another real – life example where the gamification concept is used in software development is Visual Studio Platform [A4]. Visual Studio Platform released a plug in to give badges to developers regarding their actions. By this way it aims to solve real life problems in software development process in software development landscapes. In Visual Studio Achievements users get badges in different categories (customizing the Visual Studio, Power Coder, etc.) and these badges are displayed on the user profile. As the earned badges increase the user ranking is recalculated and her status can be monitored easily on the leaderboard. These status changes of users can also be shared on Facebook to announce their progression. The researchers state that CodeHunt and Visual Studio Achievements have positive impacts on developing coding skills of users and improving the development process [A4]. The usage of gamification elements in these tools increases motivation and performance of the team members and tries to solve human related obstacles.

As previously mentioned, software testing is the important phase of software development lifecycle to produce a high-quality software product. Therefore, some gamification approaches were applied to attempt using gamification to improve the motivation of software testers. [A13] [A16]. Lombriser and Valk [A4] observed the Microsoft Language Quality Game (MLQG). The company employs the professional language translators for its multi – lingual applications. Firstly, these translators make the translations and second their translations are controlled and verified by the software quality teams by employing the native people in other countries. However, sometimes there is not enough people to employ in the company. Therefore, Microsoft has developed this gamified tool for the verification of mistranslation. Volunteers around the world can participate the testing the translations. Users earn points and badges for each mistranslation finding and these scores are displayed on the leaderboard.

Lotufo et al. [A31] gave information about StackOverflow which is a web-based question and answer site that uses game element to attract the users about the software development issues. In this web site, users ask questions about their problems and get proper solutions from other users, start discussions about the software engineering topics. They state that using game elements and mechanisms are the reasons for why it is successful. Users answer the questions or propose solutions to the problems to win points according to their contributions. Also by increasing/decreasing voting mechanism, users gain reputation points from other users. Moreover, it uses leveling mechanism to perform the specific actions. For example, newbies can only ask or answer the questions whereas experienced users can vote, rank or edit the questions. These experienced privileges can be earned by the specific action points (reputation points or levels). These indicate that it has giving the accomplishment sense to the users and causes the competition among the users. By rewarding mechanism, users try to get more reputation points and levels. Therefore, it causes to improve intrinsic and extrinsic motivations of the users to provide the best and correct solutions to the problems.

We have also observed that some glorious version tools and source controls as JIRA [11], RedCitter [12], and Visual Studio Achievements [16] in the industry. These examples use some basic game elements such as badges, levels and award mechanisms so as to monitor and evaluate the performance of each team member. Hereby the most productive team member could be determined and may also rewarded.

All of above approaches in the have positive impacts on development process and increases the quality of the software development process.

#### *4.2.1.4.* Research Question 4 (RQ4): *How these gamification approaches affect the quality of software development in software development context?*

We choose this research question to determine which approaches remedy the software development process quality. Software development process requires people involvement, user engagement, collaboration and motivation. Therefore, these are the human related factor that affects the quality of software development process. Regarding from that we conducted our review to classify and characterize the state of the art as regarding applying game elements into software development process to overcome the human related issues such as motivation, engagement, communication and collaboration.

Yilmaz et al. [A1] proposed a 3D gamification approach to explore personality types of software development team members when group formation to improve the software development process. They state that due to the human factors, software development team formations are not improper because of the incompatible personality types. Hence, they developed a 3D interactive assessment environment to overcome the human related issues. They claimed that although their proposed solution works for improving the software development as intended at the beginning of the proposal, it needs some optimizations. Kosa and Yilmaz [A2] observed game-based approaches in the literature to improve the software development process. They analyzed and reported the related researches and approaches in the literature in the software engineering and software engineering education domain. They suggest that building game – based environments for the software development process has various advantages and positive impacts on the social aspects of the software development process.

Yilmaz et al. [A8], Yilmaz and O'Connor [A10] have made a research to understand the social factors that effects the software productivity and how to increase the productivity using gamification through economic mechanism design. They claim that due to the productivity, software development is an economic activity so it has an economic mechanism. Therefore, its value should be maximized by enhancing the

human related social issues such as motivation, engagement and collaboration. Authors claim that, game theoretical approaches or gamification can be seen as the solution to overcome the social obstacles. In their research, they propose an economic model to enhance the software development process and productivity. They suggest that the proposed model should be used for the resource allocation (for instance) to increase the social interactions, overcome the human related issues and maximize the value of the software productivity.

Herranz et al. [A7] present the gamification as a solution for the disruption of software process improvement initiatives. They claim that human factors are the crucial factors that influence the software processes directly. They show the human related factors as the failure reasons for the software projects due to the low motivation, engagement and commitment. Therefore, they offer developing gamified environments to resolve the human related social factors in organization change management and software process improvement. They consider the positive psychological outcomes of gamification on software process improvement. Then, they claim that combining two concepts by validating critical success factors (organization change management and gamification) fosters the motivation, commitment, performance and productivity of the participants in the software process improvement area. Dorling and McCaffery [A6], argue that the because of its positive psychological effects, gamification may be a solution to increase the user engagement and the quality of software process improvement.

Lack of user involvement is another social aspect and human related factor (related to motivation of people) of the software development process. Fernandes et al. [A3] suggest that gamification in software development overcome the user involvement problem due to the low motivation. They proposed a gamified tool to increase the user involvement in the requirement elicitation and analysis phase of the software development lifecycle [A3]. It mainly uses the game elements like points, rankings and rewards. Their results from the case studies indicates that using this gamified tool in requirement elicitation makes it funnier and may increase the motivation, collaboration and involvement of the process actors (analysts, project managers and

project owners). Duarte et al. [A12] conducted a case study to provide an online collaborative environment for the requirement gathering phase to increase motivation and the user involvement. They use visualization techniques to show the progress of the stakeholders according to their contributions. The results indicate that using rewards facilitates improving the user motivation and participation. They also report that using social visualization technique to display user ratings increases the user involvement and collaboration. Also, they report that since it is proposal it requires a future work and needs more case studies and experiments on it to increase the validity of work.

Piras et al, [A18] conducted a comparative study for the gamifying software acceptance requirements. They use two framework which are *Agon* and *MAF* which are related cognitive principles and human and behaviors respectively. Their study is related about combining these two frameworks into one framework for designing gamification solutions for acceptance requirements. They applied their approach to meeting scheduler system. They presented comparative results for their case study. They claim that *MAF* is considered with user and systems and their behavioral issues and *Agon* is related with cognitive principles about the effects of gamified environments on people. Therefore, this shows that these comparisons and its results can help to use which one should be chosen while building a gamification solution for software acceptance requirements.

Passos et al. [A9] present how a gamified tool affects development team's motivation and engagement by conducting a case study. They conduct their case study on a medium-scaled real-world software company. They integrate their proposal which is called *DevRPG* into a task management tool. They aimed turning software development into a game with this proposal. Main goal of this proposed tool is giving real –time feedbacks to the developers to engage them. By this way, it gives a chance users to compete with each other in their teams and also other teams in their company. From the results, they claim that applying game elements into a software development process increases the performance and motivation of software development actors so as to enrich the quality of development process.

Software testing is the important phase of software development lifecycle to produce high-quality software products. Therefore, some gamification approaches were applied to attempt using gamification to increase the motivation of software testers. [A13], [A16]. Rojas et al. [A30] present a mutation testing tool that gamification and crowdsourcing is applied. *The CodeDefenders* is a mutation testing tool that has *attackers* and *defenders*. *Attackers* have role to produce mutants to fight with *defenders* whose role is protecting the code and destroying the mutants. Programmers are developing codes according to test scenarios for software testing. They decided to conduct an empirical study to apply gamification for validating the positive behavioral effects of gamification. At the end of their experiment, researchers get feedbacks from the participants and try to answer their research questions. The results are satisfying about the applying gamification and game elements to CodeDefenders. Participants claim that they are enjoying and their motivation is increasing. Also, participants claim that they can write stronger tests than the automated tests. Therefore, authors claim that, applying gamification into software testing is very useful and has many positive effects on motivation.

Sasso et al. [A5] built a proposed gamification framework to develop gamified environments to overcome the human related challenges in software development. They illustrate this framework in two examples. *The Myth and De-bug* and *The Empire of Gemstones* are the examples which is built with their framework. *The Myth and De-bug* is related to bug tracking. In this tool, users get new badges depending on their activities. For example, *Tomb Raider* badges can be earned by a user who explores old reports or setting status of the report from inactive to active. It has leaderboard for the important contributors for collaborating the team members. In *Empire of Gemstones* is related to code reviewing process in the software development process. It uses the levelling mechanism to increase the performance and motivation of the users. After evaluating two examples they observed that the performance and motivation of users are affected positively by using this proposed framework; therefore, they claim that it may help to increase quality of software development process by motivating and collaborating them.

Lombriser and Valk [A4] gives a brief information about a gamified requirement elicitation tool. This tool uses some gamification elements and mechanisms such as points, rewarding and ranking to make the requirement elicitation process more attractive and proper. This tool aims to increase the collaboration, user involvement and reduce the possible conflicts in the requirement elicitation teams and other stakeholders by combining with gamification elements and creative parallel thinking technique which is "The Six Thinking Hats" [39]. In this tool, users get points depending on their actions such as offering new requirements or suggestions to the existing requirements or adding new comments to the requirements in order to resolve conflicts. These points and achievements are shown on the progress bar to see the progress of the current state of the users and their advancements. Lombriser and Valk [A4] gave this example in their study in order to show how game elements affect the user involvement and collaboration in a non – game context and they claim that it has potential to gamify requirement process.

Melo et al. [A11] proposed a gamified version control system that uses game elements such as points and rewards to increase the engagement and collaboration during the development process. In this gamified system, they developed a proposed score system according to cyclomatic complexity of the committed code. Therefore, by this calculation developers who work on critical parts of the application can get more points and rewards comparing to others who work on less critical parts. The calculation is done based on some rules. It concerns with modified line of codes, their complexities and their effects to overall to decide whether it increases or decreases to overall complexity. Depending of this calculation developers earn points so as to show the developers' success when developing a program. These scores are shown on the developers' profile to see their performance. Melo et al. [A11] claim that this gamified environment makes development process more collaborative since it causes the balanced performance evaluation. Also, they claim that this proposal should be applied to the real-life development projects to validate the expected results in terms of engagement and collaboration.

Akpolat and Slany [A14] conducted a case study in a university environment to enhance engagement of software engineering students. In this experiment, there were 50 students who took high level extreme programming course. The authors' purpose is improving engagement of the students through gamification. In their experiment, they set the rules and declared challenges for the students. They used feedback mechanism to attract students' engagement. Every week, they announced high scores of the participants so they were able to see their progress. The participants said that using gamification in high – level extreme programming course makes it more attractive and their engagement to the course is increased. Researchers claim that, using gamification in the software engineering may increase the motivation and engagement of the students in the university and it need future work for better gaining from the adoption of gamification into software development.

Uskov and Sekar [A15] also present a proposed approach to gamify the software engineering curriculum based on the purposes such as active usage of gamification mechanisms and game elements in the software engineering course, increasing students' collaboration and engagements, improving their software engineering skills through motivating them based on the gamification, etc. To achieve these goals, first they made an analysis by observing the industrial companies that have already adopted the gamification into their activities, reviewing academic works related to gamification in software engineering field and software engineering curriculum and comparing serious games and gamification curricula. Then, they passed on the design phase to implement the gamification into software engineering curriculum. In the design phase, they use the gamification techniques and game elements such as goal settings, point, badges, leaderboards, quests, challenges, bonus, transparency, feedback mechanism, etc. Then they present the pilot study results and student feedbacks. Their conclusion about gamifying software engineering curriculum is positive and these approaches have positive impacts on the students for motivating and collaborating them and also it may help to improve their software engineering skills by educating them with gamified software engineering curriculum.

As another work in software engineering education, Singer and Schneider [A26] proposed a gamification approach for version control. Their study is related to increasing encouragement among software engineering students by doing more frequent check – in to the source control repositories. They built a web - based application which is named *Teamfeed* that is integrated with a version control system to read the students' committed data. They choose the commit counts as a metric for per participant. In this proposal, they used some game elements and gamification mechanisms such as newsfeeds, leaderboards, etc. In *Teamfeed* application, the utilized data of each students are shown on the leaderboard to see the progress of each participants and the project. It sends notifications about milestones to the team members after committing their work to the repository to motivate and encourage them. At the end of their study, authors reported the comments of the participants. Although, there some negative feedbacks are given, the overall results are positive for adoption of the gamification into the software development process. According to the positive comments, authors claim that tailoring software engineering with gamification is very useful and may contribute to improve the motivation, engagement and collaboration issues to tackle with the social related human issues in software engineering.

Matsubara and Silva [A25] conducted a case study in software engineering education. They use progression, levels, leaderboards, badges, rewards, quests and experience points as the game elements in their gamified approach. They collected qualitative and quantitative data to measure the performance of the students. The experimental results show that using game – like software engineering education increases competition and collaboration. Moreover, the qualitative and quantitative results show that their participations, attractions, scores and experiences are increased by means of adopting gamification into engineering study. Additionally, their learning curves are increased through gamified environment. It has also positive impacts on behavioral changes like increasing motivation and engagement. Therefore, we can claim that adopting gamified approach into the software development may be beneficial to tackle with the social related human issues.

Marques et al. [A27] present a proposal to gamify the Scrum projects to make it funnier and more engaging for the software development actors. They designed and developed a tool as *JIRA ad – on* that aims to gamify the Scrum projects iteratively. They explained their main goal as the increasing motivations of software practitioners in Scrum development teams. In their proposal, they use score boards, levels, badges, gems, rewards and feedback mechanism for displaying and monitoring of the user achievements. They demonstrated their tool in a company as a prototype. At the end of their work, they made an analysis of their work to determine how their proposal affects the real-world organizations. In the results, they see the positive changes in the behaviors of the practitioners and they claim that it should be used in real world organizations to affect the software development process in better way.

Sharma et al. [A17] present a gamification approach in agile software development process. They use their gamified approach in their organization which is called *Agile Workbench* as a pilot study. They built a gamification engine that has some rules and calculation methods based on *Total Net Revenue* formula. As the pilot information, this engine gets the output data from *JIRA, RTC* and *Rally* environments for the calculation of total net revenue of each team. Then this tool visualizes the results on the dashboards for the performance screening. In their approach, they show the behavioral impacts on agile software development and team members through gamification by piloting the approach in their workbench. They claim that their results from the experiments are encouraging to use the gamification in agile software delivery. It increases the collaboration with team members and with other teams. Therefore, it shows that this approach has positive effects on the development process.

Schäfer [A19] experiments the gamification on the Scrum (it is an agile software development and project management method) projects. Schäfer presents the lessons learned after two teaching periods with training Scrum through gamification using the Minecraft game. Student who took this teaching methods think that using gamification in software development education and training Scrum with

gamification is motivating than the traditional one. It also helps to bringing developers who have different point of views and backgrounds into the teams. The author claims that this is the need in the real-life software development contexts. Although author claims that the experiment has some drawbacks and needs to work on it, it may contribute to improve the software development with Scrum.

Liechti et al. [A21] conducted a case study with a test analytic platform to support the agile delivery. They bring the agile methods which are continuous integration, automated testing and feedback mechanism together and called it as *test analytic*. They use *Behavior Driven Development (BDD)* collaborate the participants who have different roles in the teams. They add the gamification features to add fun to the process. They use feedback mechanism, points, badges and leaderboards for motivating users. At the end, they get the positive results from their case study. It has positive impacts on collaborating and motivating the Scrum team members.

Unkelos-Shpigel and Hadar [A28] presented a conceptual solution to overcome the challenges in software development depending on code reviewing issue with cognitive principles. Their purpose is collaborating team members by knowledge sharing among the colleagues and increasing the productivity during the development process. To reach this goal, they developed *SCRUT (Social Code Review Unifying Tool)* based on the game elements and cognitive principles to make the code review process more effective and efficient. In this conceptual solution, there are two types of user which are programmer and reviewer and each user has an initial score. If the programmer's code is accepted by the reviewer s/he (in some situations her team gets points also) get the points and also s/he get the points by asking a review. Also, reviewers get points based on their contributions to encourage the reviewers. In this study, authors' goal is that increasing collaboration and productivity in software development process based on code reviewing perspective. They claim that this conceptual solution has positive outcomes on software development process and contributions to improve the productivity and increase the quality of work.

Herranz et al. [A29] presented *Gamiware* as a solution to increase the motivation in software projects. They conducted a pilot study to see the effects of *Gamiware* in software process improvement. While conducting this pilot study they have two purposes: observing the effect of the tool on motivation changes of people and the second is that testing of applicability of this tool in software development process. As the results indicate that, in overall, participants who uses this tool is satisfied with the positive feelings about their motivation and engagement. However, there some negative feedbacks are presented such as lack of fun in the upper levels and raising stress when their doing tasks. Although they claim that their proposal needs more evidences using this tool could be helpful for improving motivation and engagement of participants.

Nicholson [A23] and Dubois and Tamburelli [A24] proposed the gamification framework to enhance the quality of work, improve the motivation and engagement considering human related issues. Dubois and Tamburelli [A24] outline gamification mechanisms and game elements to adopt it into the software development process to overcome the human related obstacles. They made a deep research analysis about software engineering gamification and propose a research approach for the applicability of gamification in different contexts. And they approach it on software engineering education. They claim that, applying gamification into software development is an easy task but predicting its possible effects needs more future work and evidences.

Steffens et al [A32] [A20] proposed a preliminary gamification framework to apply the game elements to reduce the human related factors in software engineering fields and collaborate software teams by using gamification. They state that motivation of the software actors is the key and most important factor for producing a well quality software product. Therefore, they decided to build a gamified framework to overcome the human related issues. Steffens et al. [A20] conducted a qualitative study to change the behavior of software practitioners in terms of increasing collaboration and motivation by adopting gamification framework into software development teams. They report that the participants thought gamifying software

development may be useful for the improvement of the motivation and collaboration issues. Therefore, they decided to conduct a case study for their next study.

In summary, we choose this research question to determine which approaches remedy the software development process quality. We analyze the case studies, experiments, research works and real-life examples to determine how gamification approaches contributes to enhance the software development process to overcome the human factors. Studies that we analyze show that using game elements and gamification approaches in software development process have positive impacts on increasing productivity and quality of work by resolving human factors.

#### *4.2.1.5. Summary*

In summary, we choose these research questions to investigate the currently known issues about people's motivation, engagement and collaboration during the software development process. Likewise, we investigate which game elements and gamification approaches can be used during the software development process to increase the quality and performance of software development process. Moreover, we observe real – life examples that try to attempt to adopt gamification into their software development process and its intended results. In addition, we try to determine which approaches remedy the software development process quality by observing the related studies in the literature. Therefore, for all our research questions, we analyze related studies almost all phases of software development lifecycle of software development process. Studies that we analyze show that using these different gamification approaches in software development process have positive effects on people's motivation, collaboration and engagement. These are critical human related social issues that affect the software quality and productivity during the software development process in the landscapes such as real – world organizations, software engineering educational areas. Therefore, although most aforementioned approaches are the proposal and case studies, they propose and present that using game elements and applying gamification concept into software

development process fosters the productivity of software development, increases the quality of work by tackling with the human related social issues.

# CHAPTER 5

# DISCUSSION

In this chapter, we will provide a discussion of the results obtained from the literature review.

## 5.1. Discussion

As the software development process is a human centric and brain intensive activity human factors have effects on the development process for its success or failure. Therefore, we have decided to conduct this systematic literature review to classify and characterize the state of the art as regarding applying game elements into software development process from different perspectives such as software development, requirement elicitation, testing and software engineering education, etc. in which human plays a crucial role. We considered these perspectives as the main categories of this study while covering 32 papers (*see* APPENDICES A *section*) which are published between 2010 – 2017.

According to Kitchenham [6] the reasons for conducting a systematic literature review are:

- Summarizing the existing research works that presents the evidential data to summarize the benefits of related topic.
- Identification and clarification of the gaps in the current works so as to give suggestions for future works.
- Providing frameworks or backgrounds for new research related to research field.

Therefore, we defined the following research questions (*as defined in the INTRODUCTION part*) to investigate this study.

- What is currently known about people's motivation, engagement and performance issues in software development process?
- Which game elements and gamification approaches can be applied into software development landscapes to increase the quality and performance of software development?
- What are the examples of these gamification approaches for the software industry?
- How these gamification approaches affect the quality of software development in software business context?

These were the main purpose of this SLR to investigate these research questions. Thus, we analyzed 32 papers (*see* APPENDICES A *section*) that present the data to adopt the gamification into software development process in the software development landscapes. As a result of this analysis, people's motivations, engagements and collaborations are negatively affected due to lack of communication, misunderstanding requirements and objectives, late performance evaluation and lack of reward systems by the management. They show these reasons for the human related problems in the software development landscapes.

Game elements and mechanisms are one of the focal point of our analysis to investigate our research questions. We analyzed which game elements are mostly used to build up gamified environments for the software development process. According to **Table *12*** and ***Figure 8***, points, badges and leaderboards are mostly used game elements to design gamified tools and environments for the software development process. Therefore, we derived from our included studies that these game elements are more suitable game elements to emerge gamification with software development process in order to motivate, engage and collaborate the users in the software development landscapes.

Considering with game elements and mechanisms, we conducted our analysis to apply gamification into the software development process. According to this

analysis, we realized that applying game elements and gamification into software development process have potential to overcome the human related issues but still there are insufficient evidences about its positive outcomes. Thus, it needs more evidences and empirical data to prove its contributions to the software development process. According to the results that we presented in *Results* chapter, some of the included studies presented case studies and research works. Therefore, it indicates that application of gamification into software development process needs more empirical data to show its effective applicability. However, we also observed the studies that present the empirical data and real-life examples in which gamification approaches have already been adopted into the software development process. These studies reveal the benefits of gamification in the software development process and show the positive outcomes of these gamification approach. Therefore, these real-life results show that gamification technique can successfully be applied into the process areas which are software project management, requirement elicitation, software development and testing to increase the users' motivation, engagement and collaboration in the software development landscapes.

In summary, based on our analysis gamification seems a way to resolve the human related problems (e.g. motivation, engagement and collaboration) during the software development process in the software development landscapes. Gamification can be applied into the process areas which are requirement elicitation, software development and testing, software management, software engineering education, and software process improvements to make behavioral changes and increase user motivation, collaboration, engagement and performance issues.

# CHAPTER 6

# CONCLUSION

In conclusion, this study shows the applicability of game elements in the context of software development landscapes. At the beginning of the study, we intended to show the application of game elements both in research and industrial levels of software development and as well as in software development landscapes. As we mentioned before previously, human plays a crucial role during the software development process intensively. Therefore, our aim is to identify the human related factors such as motivation, engagement, collaboration and performance and what causes these factors during the software development. Then, we investigated gamification as a solution which is a new trend that has been used in almost all areas in real life such as marketing, education, health and also business life.

We realized that, numerous studies have been conducted in the literature to examine the benefits of gamification and how game elements affect the software development process. Pedreira et al. [52] conducted a systematic mapping study for the gamification of software engineering. They conducted their mapping study by including the studies which are published between 2011 and 2014. We conducted our systematic literature review by including studies between 2010 and 2017. Therefore, we have a chance to observe the studies before 2011 and after 2014. Depending on our research questions we have observed, case studies, experiments and real life examples to investigate the possible effects of applying gamification into the software development process.

Thus, with the state of the art as described, we decided to conduct a systematic literature review to classify and characterize the state of the art as regarding applying game elements into software development process. To conduct our systematic literature review we followed the instructions and principles which are described in [3, 17, 47, 48, 49] for better understanding and concluding our investigation with better solutions. Depending on the recommendations [3, 17, 47, 48, 49] we followed the systematic literature steps which consist of three main steps which are *Planning, Conducting* and *Documenting.*

In our study, we focused on investigation of gamification in software development process to solve the underlying human related issues. In order to achieve this goal, firstly, we determine what currently known is about people's motivation engagement and collaboration and performance issues in software development process. Then we give brief information about gamification, software development process and life cycle to create an infrastructure of the study. Then we started to investigate our research questions to find solutions to the current problems in software development landscapes. Then we made a preliminary investigation [45] and share the results. The results that we obtained from our preliminary investigation [45] on applying game elements into software development landscapes provides a basic guidance for the software development organizations which aims to benefit from gamification. In our previous work we included and observed 12 studies in order to investigate the solutions to overcome human related obstacles and improve software development process. This was also a systematic literature review to the 17th International Conference on Process Improvement and Capability dEtermination (SPICE 2017). This paper [45] has been accepted as a full technical research paper and has four reviewer comments to make us to address the comments and feedbacks in our revision before publication stage. In this process, we got positive comments and made some revisions according to their feedbacks and sent our paper for the publication.

We revised this study by considering the comments and feedbacks of preliminary investigation [45] and we included 20 more studies to strengthen our findings and to

get more evidence for our research. At the beginning of this literature review, we intended that using gamification in software development process should reduce the problems which are related to human factors (e.g. user motivation, engagement and collaboration, etc.). We included and examined 32 studies which are research and conference papers, proposals and case studies about software process improvement, software engineering education and software engineering domains to increase the motivation, engagement and collaboration. In addition to our preliminary investigation, we also included new studies which are related to gamifying the steps of software development life cycle in software development process. We analyzed the papers which are related to gamification of requirement elicitation process, software implementation, teaching coding and testing, version control systems, code reviewing, software testing and bug tracking systems, software process improvements and SCRUM projects to make behavioral changes and increase user motivation, collaboration, engagement and performance issues. Thus, the results that we obtained from our extended research confirmed that using game element in these areas supports our expectations. In addition, our results suggest that using gamification in software development increases the user motivation, engagement and collaboration. It also proves that it improves the software development process in terms of quality and performance, which creates a potential to resolve some obstacles related to human factors [5, 21, 22]. Based on the real-life examples [23, 24, 25], [A19], [A27] the adoption of gamification approach into software development process provides some evidences to confirm our research goal. Therefore, it shows that it can be adopted into software development process in the software development landscapes. We also examined the case studies that are conducted in software engineering education and software organizations. Their proposed solutions suggest that gamification of software development have positive impacts on enhancing the software development process by reducing the human related issues.

In conclusion, this research provides a guideline to assess the applicability of gamification in software development landscapes to increase the quality of software development process by eliminating the human related issues. This research contributes to software engineering research field by proposing classification approach to categorize the related gamification studies in software development

process. Therefore, a systematic literature review was conducted to assess the association between gamification and software development process in software development landscapes. These empirical studies and proposals meets our expectations as we intended at the beginning of this study. The results of this research suggest that there has been a significant increase in empirical and theoretical studies that were recently published and many positive steps being taken towards better integration of gamification in software development organizations.

### 6.1. Threats to Validity

As Yilmaz [1] states that there are some threats that may affect the validity and reliability of the study when conducting scientific research. Therefore, we described our threats to validity depending on the publication bias and identification and data extraction of the primary studies. We conducted this systematic review using the academic databases (see **Figure *7***) which are Çankaya University Electronic Database, IEEE Explore, Science Direct, Access Engineering, ACM Digital Library, SpringerLink.to observe and analyze the relevant data in the literature. We also use standard Google Search as the non - academic search to increase the number of papers to access the relevant data. However, although we have extended our data sources to get more studies in the literature, some other important relevant studies may have been missed out. Also, as our study selection criteria and procedure, we tried to keep type of forum as wide as possible. (conference papers, academic papers, academic online papers, etc., see **Table *3*** and **Figure 6**) However, we accessed 28 conference papers according to our search operation, so it may affect our validity of review study.

We designed a search strategy to conduct this systematic literature review to reach as many papers related to our research topic. As part of our search strategy we used popular databases as data sources to access the studies in the literature. These data source are Çankaya University Electronic Database, IEEE Explore, Science Direct, Access Engineering, ACM Digital Library, SpringerLink as listed in section 3.4. To investigate our research question, we developed search strings related to gamification

of software development process by combining major terms with alternative terms. These search strings are designed by following the steps which are explained in [18]. We combined these search strings using Boolean AND or OR operator to retrieve the studies to get the relevant data about our topic. For example, we used *(Software Development Process OR ((software development))* ***AND*** *(Game Elements OR ((gamification))* as the search string to access the studies which are related to game elements or gamification of software development process. Also, another (defined in section 3.4) different combinations of search strings are used to access the similar studies to retrieve different studies which are related to our research topic. According to Kitchenham and Charters [6] a review protocol including study selection criteria and procedure is required for the data extraction phase to conduct the systematic review more efficient and effective. Therefore, we execute our study inclusion and exclusion criteria to include more reliable results and decrease the risk to stray away from the topic. Our search is applied to titles, abstracts and keywords to eliminate the studies which presents irrelevant data or mismatches our study selection criteria. For example, we excluded the studies [50] and [51] because they were not written in English language although they are related to our research topic. In our study, we executed our study selection procedure in two phases. As the first phase, we read the titles and abstracts of the papers from the first search results and we excluded some papers which are not conform to our study inclusion and exclusion criteria. As the second phase, we observed the remaining papers that meet our inclusion and exclusion criteria. These papers are eligible to read for full text reading for the data extraction phase. For the data extraction phase, we generated a form and assign properties (see **Table *5***) to match these papers with our research questions to increase the reliability and consistency to get the expected results from this review. To summarize, for the validity threats, we designed a search strategy to define our search strings, data sources and review protocol for study selection criteria and procedure to reduce the risk to stray away from topic and get more reliable outcomes from our investigation.

## 6.2. Limitations

There are some limitations about the software development types to prevent generalization of the results while conducting this SLR study. For example, we did not consider *Extreme Programming, Rapid Application Development (RAD)* and *Joint Application Development (JAD)* for our study to prevent the conflicts about our analysis results due to their software development techniques.

## 6.3. Future Work

The concluding remarks of this study outlines the important areas for further work. First, further research regarding the future of gamification in process improvement activities would be interesting outlet. The outcomes of this study might also help us to understand the social mechanisms of software development. A future study investigating how gamification can be integrated to DevOps would be a fruitful area for further work. As another work, a systematic review need to be carried out in order to assess the effects of gamification for the *Extreme Programming, Rapid Application Development (RAD)* and *Joint Application Development (JAD).*

## REFERENCES


1. **Yilmaz M., (2013),** "A Software Process Engineering Approach to Understanding Software Productivity and Team Personality Characteristics: An Empirical Investigation", PhD Thesis, Dublin City University

2. **Petersen K., Feldt R., Mujtaba S., Mattsson M., (2008),** "Systematic mapping studies in software engineering", in: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE'12), pp. 68–77

3. **Kitchenham B., (2004),** "Procedures for Performing Systematic Reviews", Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia.

4. **Deterding S., Khaled, R., (2011),** "Gamification: Toward a Definition." In CHI '11 Gamification Workshop. ACM.

5. **Deterding S., Dixon D., Khaled R., Nacke L., (2011),** "From game design elements to gamefulness: defining gamification'', in: Proceedings of the 15th International Academic MindTrek Conference, Envisioning Future Media Environments (MindTrek'11), pp. 9–15

6. **Kitchenham B., Charters S., (2007),** "Guidelines for performing systematic literature reviews in software engineering," Software Engineering Group, Keele University and Department of Computer Science, University of Durham, United Kingdom, Technical Report EBSE-2007-01.

7. **Kapp K. M., (2012),** The Gamification of Learning and Instruction, Pfeiffer. ISBN: 978-1-118-09634-5, May 2012

8. **Currier J., (2008),** Gamification: Game Mechanics is the New Marketing, OogaLabs Blog.

9. StackOverflow. http://stackoverflow.com.

10. **Hugos, M., (2012),** "Enterprise Games: Using Game Mechanics to Build a Better Business", O'Reilly, 2012

11. Visual Studio Achievements, Retrieved July 19, 2017, from https://channel9.msdn.com/achievements/visualstudio

12. Atlassian, JIRA Hero. Retrieved July 19, 2017 https://marketplace.atlassian.com/plugins/com.madgnome.jira.plugins.jirachievements.

13. PropsToYou, Retrieved July 19, 2017, from https://www.propstoyou.com.au/

14. ScrumKnowsy, Retrieved July 19, 2017, from http://www.scrumknowsy.com/

15. Masterbranch. Retrieved July 19, 2017, from http://masterbranch.com/

16. RedCritter, Retrieved July 19, 2017, from http://www.redcritter.com/

17. **Kitchenham B.A., Budgen D, Pearl Brereton, O., (2011),** "Using mapping studies as the basis for further research – a participant–observer case study", Inf. Softw. Technol. 53. Pages: 638–651

18. **Brereton P., Kitchenham B., Budgen D., Turner M., Khalil M., (2007),** "Lessons from applying the systematic literature review process within the software engineering domain", J. Syst. Softw. Pages: 571-583

19. **Tsumaki T., Tamai T., (2006),** "Framework for matching requirements elicitation techniques to project characteristics," Softw. Process Improv. Pract., vol. 11, no. 5, pp. 505–519.

20. **Herger M., (2014),** Enterprise Gamification: Engaging people by letting them have fun (Volume 1), 1st ed. CreateSpace Independent Publishing Platform, 2014

21. **Paharia R., (2013),** “Loyalty 3.0: How to Revolutionize Customer and Employee Engagement with Big Data and Gamification”, McGraw-Hill.

22. **Werbach K., Hunter D., (2012),** “For the Win: How Game Thinking Can Revolutionize Your Business”, Wharton Digital Press.

23. **Aydan U., Yilmaz M., Clarke P., O'Connor R.V.,** “Teaching ISO/IEC 12207 software lifecycle processes: A serious game approach.” Computer Standards and Interfaces journal, 54(3), pp. 129-138

24. **Kosa M., Yilmaz M., O'Connor R.V., Clarke P.** “Software Engineering Education and Games: A Systematic Literature Review. In: Journal of Universal Computer Science, 22(11), pp.1558-1574 (available online).

25. **Yilmaz M., Yilmaz M., O'Connor R.V., Clarke P., (2016),** “A Gamification Approach to Improve the Software Development Process by Exploring the Personality of Software Practitioners.” In: Proceedings of the 16th International Conference on Software Process Improvement and Capability dEtermination (SPICE 2016)

26. **Hamari J., Lehdonvirta V., (2010),** “Game design as marketing: How game mechanics create demand for virtual goods”, International journal of business science and applied management, 5(1), pp. 14-29.

27. **Jamil M. A., Arif M., Abubakar N. S. A., Ahmad A., (2016),** “Software Testing Techniques: A Literature Review”, Information and Communication Technology for The Muslim World (ICT4M), 2016 6th International Conference, pages 177 – 122, 2016.

28. **Yilmaz M., O'Connor R., Collins J., (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

29. **Gartner (2011)**, “Gartner says by 2015, more than 50 percent of organizations that manage innovation processes will gamify those processes”, http://www.gartner.com/newsroom/id/1629214.

30. **Huotari K., Hamari J., (2012),** "Defining gamification: a service marketing perspective", In Proceedings of the 16th International Academic MindTrek Conference, October 3-5, 2012, Tampere, Finland, ACM, pp. 17-22.

31. **Zhang H., Babar M. A., (2012),** "Systematic reviews in software engineering: An empirical investigation", Information and Software Technology 55 (2013) 1341–1354.

32. **Lombriser P., Valk R.,** "Improving the Quality of the Software Development Lifecycle with Gamification." Department of Information and Computing Sciences, Utrecht University.

33. **Groh F., (2012),** "Gamification: State of the art definition and utilization," Institute of Media Informatics Ulm University, vol. 39.

34. **Beecham S., Baddoo N., Hall T., Robinson H, Sharp H., (2008),** "Motivation in software engineering: A systematic literature review." Information and Software Technology, 50(9-10):860 - 878.

35. **Kitchenham B.A., Dybå T., Jørgensen M., (2004),** Evidence-based software engineering, in: Proceedings of the 26th International Conference on Software Engineering, (ICSE'04), IEEE Computer Society, Washington DC, USA, pp. 273–281.

36. **Hamari J., Koivisto J., Sarsa H., (2014),** "Does Gamification Work? — A Literature Review of Empirical Studies on Gamification", 2014 47th Hawaii International Conference on System Science.

37. **Bano M., Zowghi D., (2013),** "User Involvement in Software Development and System Success: A Systematic Literature Review", Proceeding EASE '13 Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering, pages 125-130.

38. **Zowghi D, Coulin C., (2005),** "Requirements Elicitation: A Survey of Techniques, Approaches, and Tools." In: Aurum A., Wohlin C. (eds) Engineering and Managing Software Requirements. Springer, Berlin, Heidelberg

39. **De Bono E., (1995),** “Parallel thinking: from Socratic thinking to de Bono thinking, Penguin: Language, linguistics, Penguin.

40. **Sommerville I., (2010),** Software Engineering, 9th ed, International Edition, Boston, Published by Addison-Wesley, March 2010, ISBN-10: 0-13-703515-2, ISBN-13: 978-0-13-703515-1, Chapter 2, Software Processes; pages 27-55

41. **Viskovic D., Varga. M., Curko K. (2008),** “Bad practices in complex IT projects”, ITI 2008 - 30th International Conference on Information Technology Interfaces, pages 301 – 306.

42. **Kusumasari T. F. et al. (2011),** “Collaboration model of software development.” In Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, pages 1 - 6.

43. **Rafiq U. et al. (2017),** “Requirements Elicitation Techniques Applied in Software Startups”, 2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pages 141 – 144.

44. **Gaur J. et al. (2016),** “A walk through of software testing techniques”, 2016 International Conference System Modeling & Advancement in Research Trends (SMART), pages 103 – 108.

45. **Olgun S. et al., (2017),** “A Systematic Investigation into the Use of Game Elements in the Context of Software s: A Systematic Literature Review.” In: Mas A., Mesquida A., O'Connor R., Rout T., Dorling A. (eds) Software Process Improvement and Capability Determination. SPICE 2017. Communications in Computer and Information Science, vol 770. Springer, Cham.

46. **Silva V.M. et al. (2014),** “The Importance of Systematic Review as a Scientific Research Methodfor Computer Science: A Quantitative Systematic Review”, doi: 10.7763/IPEDR. 2014. V 78. 9.

47. **Kitchenham B. A. et al. (2002),** “Preliminary guidelines for empirical research in software engineering,” IEEE Trans. Softw. Eng.,vol. 28, no. 8, pp. 721–734.

48. **Biolchini J. et al. (2005),** "Systematic Review in Software Engineering, Technical Report, Universidade Federal do Rio de Janeiro.

49. **Kitchenham B.A., Hughes R.T., Linkman S.G., (2001),** "Modeling Software Measurement Data" IEEE Trans. Software Eng., vol. 27, no. 9, pages. 788-804.

50. **Hernández L. et al.** (**2017),** "Application of gamification elements in software engineering teamwork", 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), pages 1-7.

51**. Gómez-Álvarez M. C., Gasca-Hurtado G. P., Jesús Andrés Hincapié, J.A. (2017),** "Gamification as strategy for software process improvement: A systematic mapping", 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), pages 1 – 7.

52. **Oscar Pedreira O., García F., Brisaboa N., Piattini M., (2015),** "Gamification in software engineering –A systematic mapping." Inf. Software Technol. 57, 157–168, https://doi.org/10.1016/j.infsof.2014.08.007

53. **Bassil Y., (2012),** "A Simulation Model for the Waterfall Software Development Life Cycle"**,** International Journal of Engineering & Technology (iJET), ISSN: 2049-3444, Vol. 2, No. 5, 2012, http://iet-journals.org/archive/2012/may_vol_2_no_5/255895133318216.pdf

54. **Abran A., Moore J. W., Bourque P., Dupuis R., (2004),** "Guide to the Software Engineering Body of Knowledge", SWEBOK, A project of the IEEE Computer Society Professional Practices Committee, 2004

## APPENDICES A

**[A1]** Yilmaz, M., Yilmaz, M., O'Connor, R., Clarke, P. 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

**[A2]** Kosa, M., Yilmaz, M. Designing Games for Improving the Software Development Process. Systems, Software and Services Process Improvement, 22nd European Conference, EuroSPI 2015, vol. 543, pp. 303–310, 2010.

**[A3]** Fernandes J., Duarte, D., Ribeiro, C., Farinha, C., Pereira, J. Madeiras, and Mira da Silva, M. ‘‘iThink: A Game-Based Approach Toward Improving Collaboration and Participation in Requirement Elicitation,’’ Procedia Computer Science, vol. 15, pp. 66–77, 2012 2012.

**[A4]** Lombriser, P. and Valk, R. “Improving the Quality of the Software Development Lifecycle with Gamification. Department of Information and Computing Sciences, Utrecht University.

**[A5]** Sasso T. D., Mocci A., Lanza M., Mastrodicasa, E. “How to Gamify Software Engineering”. 2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), 261 – 271.

**[A6]** Dorling, A., McCaffery, F., ‘‘The gamification of SPICE,’’ Communications in Computer and Information Science, vol. 290, pp. 295–301, 2012.

**[A7]** Herranz, E., Palacios, R. C., de Amescua Seco, A., Yilmaz, M. (2014). Gamification as a Disruptive Factor in Software Process Improvement Initiatives. J. UCS, 20(6), 885-906.

**[A8]** Yilmaz, M., O'Connor, R., Collins, J. 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

**[A9]** Passos, Erick B, Danilo B Medeiros, Pedro AS Neto and Esteban WG Clua. Turning Real-World Software Development into a Game. Games and Digital Entertainment (SBGAMES), 2011 Brazilian Symposium on, IEEE. 2011.

**[A10]** Yilmaz, M. and O'Connor, R. (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

**[A11]** A. A. de Melo, M. Hinz, G. Scheibel, C. D. M. Berkenbrock, I. Gasparini, and F. Baldo, ‘‘Version Control System Gamification: A Proposal to Encourage the Engagement of Developers to Collaborate in Software Projects,’’ presented at the Proceedings of the 6th Int. Conf. on Social Computing and Social Media (SCSM’2014).

**[A12]** Duarte, D. Farinha, Silva, C. M. and Silva, A. R. ‘‘Collaborative Requirements Elicitation with Visualization Techniques,’’ presented at the Proceedings of the IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE’12), 2012.

**[A13]** Arai, S., Sakamoto, K., Washizaki, H. and Fukazawa, Y. “A Gamified Tool for Motivating Developers to Remove Warnings of Bug Pattern Tools, 2014, 6th International Workshop on, Nov 2014, pp. 37–42.

**[A14]** Akpolat, B. S., and Slany W., “Enhancing software engineering student team engagement in a high-intensity extreme programming course using gamification”, 2014, 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T), pp. 149 – 153.

**[A15]** Uskov, V. and Sekar, B. "Gamification of software engineering curriculum," Frontiers in Education Conference (FIE), IEEE, pp. 22–25, 2014.

**[A16]** Fraser, G. Gamification of Sofware Testing. In: 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing, 2017, pp. 2-7. Department of Computer Science, The University of Sheffield, Sheffield, United Kingdom.

**[A17]** Sharma, V.S., Kaulgud, V., Duraisamy, P. “A Gamification Approach for Distributed Agile Delivery.” In: 2016 IEEE/ACM 5th International Workshop on Games and Software Engineering (GAS), 2016, pp. 42 – 45, Accenture Technology Labs, Accenture, Bangalore, India.

**[A18]** Piras L., Paja E., Giorgini, P., Mylopoulos, J., Cuel, R., Ponte, D. “Gamification Solutions for Software Acceptance: A Comparative Study of Requirements Engineering and Organizational Behavior Techniques.” 2017, pp. 255 – 265, Department of Information Engineering and Computer Science, University of Trento.

**[A19]** Schäfer, U. Training scrum with gamification: Lessons learned after two teaching periods. In: 2017 IEEE Global Engineering Education Conference (EDUCON), 2017, pp. 754 – 761, East Bavarian Technical University of Applied Sciences (OTH-AW) Media Informatics and Mobile Computing Faculty of Electrical Engineering, Media and Computer Science.

**[A20]** Steffens, F., Marczak, S., Filho, F. F., Treude, C., Singer, L. and Souza, C. R. B., “A preliminary evaluation of a gamification framework to jump start collaboration behavior change.” in Proceeding CHASE '17 Proceedings of the 10th International Workshop on Cooperative and Human Aspects of Software Engineering Pages 90-91, 2017.

**[A21]** Liechti, O., Pasquier, J., Reis, R. “Supporting Agile Teams with a Test Analytics Platform: A Case Study”. In: 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST), 2016, pp. 9 – 15.

**[A22]** Kolpondinos, M. Z. H., Glinz, M. “Tailoring Gamification to Requirements Elicitation: A Stakeholder-Centric Motivation Concept”. 2017 IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), 2017, pp. 9 – 15.

**[A23]** Nicholson, S. "A user-centered theoretical framework for meaningful gamification." Games+ Learning+ Society 8.1 (2012): 223-230.

**[A24]** Dubois, D. J., Tamburrelli, G. “Understanding Gamification Mechanisms for Software Development” Proceeding ESEC/FSE 2013 Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, pages 659-662, 2013

**[A25]** Matsubara, P.G.F., and Silva, C. “Game Elements in a Software Engineering Study Group: A Case Study”, 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET), pages 160-169, 2017.

**[A26]** Singer L., and Schneider, K. “It was a bit of a race: Gamification of version control.” In Int. Workshop on Games and Software Engineering (GAS), pages 5–8. IEEE, 2012

**[A27]** Marques, R., Costa, M., Silva, M.M., Gonçalves. P., “Gamifying software development scrum projects.” 2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games) pages 141–144. IEEE, 2017

**[A28]** Unkelos-Shpigel, N. and Hadar, I. “Gamifying software engineering tasks based on cognitive principles: The case of code review,” Proceeding CHASE '15 Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering, Pages 119-120, 2015

**[A29]** Herranz, E., Colomo-Palacios, R., de Amescua Seco, A.: Gamiware: a gamification platform for software process improvement. In: O’Connor, R.V., Umay Akkaya, M., Kemaneci, K., Yilmaz, M., Poth, A., Messnarz, R. (eds.) EuroSPI 2015. CCIS, vol. 543, pp. 127–139. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24647-5_11

**[A30]** Rojas, J. M., White, T. D., Clegg, B. S., Fraser, G., “Code Defenders: Crowdsourcing Effective Tests and Subtle Mutants with a Mutation Testing Game.” 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pages 677 - 688, 2017

**[A31]** Lotufo, R., Passos, L. T. and Czarnecki K. “Towards improving bug tracking systems with game mechanisms.” In 9th IEEE Working Conference of Mining Software Repositories, MSR 2012, June 2-3, 2012, Zurich, Switzerland, pages 2 - 11, 2012.

**[A32]** Steffens, F., S. Marczak, F. Figueira Filho, C. Treude, L. Singer, D. Redmiles, and B. Al-Ani “Using gamification as a collaboration motivator for software development teams: A preliminary framework,” in Proc. of the Brazilian Collaborative Systems Symposium, Salvador, Brazil, ACM, 2015.