Deanship of Graduate Studies

Al-Quds University

Gamification of Agile Project Management for Software Development Projects and Tasks

Areej Ibrahim Salman Aldaghamin

M.Sc. Thesis

Jerusalem-Palestine

1441/ 2020

Gamification of Agile Project Management for Software Development Projects and Tasks

Prepared By:

Areej Ibrahim Salman Aldaghamin

B.Sc. Information Technology from – Al-Quds University - Palestine.

Supervisor: Dr. Rashid Jayousi

This thesis has been submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Computer Science /Department of Computer Science Faculty of Science &Technology /Deanship of Graduate Studies /Al-Quds University.

1441/2020

Deanship of Graduate Studies

Al-Quds University

Computer Science

Thesis Approval

Gamification of Agile Project Management for Software Development Projects and Tasks

This thesis is approved for recommendation to the Graduate Council.

Prepared by: Areej Ibrahim Aldaghamin

Student ID No: 21510410

Supervisor: Dr. Rashid Jayousi

Master thesis submission and acceptance date: 15/07/2020

The name and signatures of examining committee members are as follows:

1. Head of Committee: Dr. Rashid Jayousi	**Signature**	R. Jayousi
2. Internal Examiner: Dr. Radwan Kasrawi	**Signature**	
3. External Examiner: Dr. Derar Eleyan	**Signature**	D.Eleyan

Dedication

I dedicate my thesis to my family. A special feeling of gratitude to my loving parents, I dedicate this to my supervisors for their guidance throughout my dissertation journey. I also dedicate this to everyone who encouraged and supported me for this research.

Thank you all

Areej Ibrahim Aldaghamin

Acknowledgments

This project would never have been possible without the support and guidance of various professionals at the University of Al-Quds. It was mainly supported by the Computer Science Department.

I would like to thank Dr. Rashid Jayousi and Dr. Radwan Kasrawi for their assistance with my thesis, and for their continued invaluable feedback that greatly improved the quality of my research.

I would also like to express my gratitude to Dortmund University in Germany for their support, and for sharing pearls of wisdom? throughout the course of this research.

Finally, I would like to thank my friends and family who supported me throughout my time working on my thesis project. Moreover, I would like to thank my parents for their unwavering encouragement, and support.

Abstract

This thesis focuses on the enhancement of team members' performance working in Agile software projects. Agile projects are complex, contain rapid changes, and require less documentation. These projects depend upon team performance and evaluating it. Gamification is a tool used to motivate users to behave as desired, which can be merged ideally with software development.

The approach of this research is to gamify project management systems of Agile software development to ensure high performance of team members while recognizing individuals' differences, and to encourage each team member to work towards a common goal. A proposed prototype was created by combining gamification elements to a task management system in software projects, particularly for the development tasks to study their impact on the team members' performance. The implications of embedding gamification in the work environment were also studied. Its application was examined as an added value to the management and development of teams of software projects in IT companies. This was achieved through a six-step approach to gamify software development tasks that include presenting, applying, and reflecting on its advantages from the perspective of both management and employees.

This research presents test case results. The tests were performed with a group of twenty developers, they were interviewed after completing usability testing in the context of gamification. The results presented not only support how gamification shows an increase in the developers' performance and productivity but also display its benefits on the management team evaluation.

Keywords

Software Project Management, Gamification, Task Management System

Table of Contents

List of Figures

List of Tables

List of Equations

List of Appendices

List of Abbreviations

Abbreviation	Full Name
MDA	Mechanic Dynamic Aesthetics
MDC	Mechanic Dynamic Components
PMI	Project Management Institute
D6	Six Step Framework
GTMS	Gamification task Management System
HQ	Hedonic Quality
PQ	Pragmatic Quality
WVP	Minimally Viable Product

Chapter 1 Introduction

Introduction

“Games are the most elevated form of investigation” - Albert Einstein.

One of keys to success in any project lies in the high motivation of the project team members. Unfortunately, traditional administrative mechanisms are often tiring and demotivating to users. Gamification is a new trend in user interface design, and human interaction studies. Gamification aims to examine the motivation that users express in order to achieve desired goal and add game elements to non-game systems to keep up motivation as well as engagement. Gamification is currently popularly used and it shows excellent results within the context of user engagement and motivation. It has shown great outcomes in the fields of education, software and health.

Essential objectives of a successful project must include having an engaged, productive, and motivated team. This is specifically important in software projects due to the dynamic nature of the field, and the work. Gamification shows successful outcomes in education and self-assessment; recently, its application has also proved its practicality to businesses. The primary goals of applying gamification are to add interactive fun elements to routine team tasks in order to improve the sense of motivation, enjoyment, engagement, and commitment [1]. Such concepts guarantee a long-term motivation base to keep using the software, and to enhance performance.

Project Management is defined according to the Project Management Institute [2] as the application of knowledge, skills, tools, and techniques to a broad range of activities in order to meet the requirements of a particular project. Every project is constrained by time, cost, and quality [2]. One of the crucial aspects of project success is having an engaged and motivated team [2]. Modern leading organizations use game elements and point systems to create a positive and competitive environment between their employees along with understanding the performance of teams in order to find out ways to enhance it [3]. Adding fun, game elements to a new system or an existing one, which in turn enhances the user experience, is called ‘gamification’. The user implied in this context is the employee. It is specifically needed for software development projects where people management has been considered a key concern [4]. This concern has also been cited in various reference models such as the People Capability Maturity Model Curtis, 2001, and Team Software Process [5].

Adding gamification can inspire a positive influence on the motivation of users. Game elements results can also help in monitoring and improving the personal differences of system users.

Gamification was defined by the academics Kevin Werbach and Dan Hunter as "the use of game elements and game-design techniques in non-game contexts." In order to engage and motivate people to achieve certain goals in projects their research should focus on the use of gamification elements in non-game related contexts, which in turn influences both personal-goals and team-goals [6]. Recently gamification has been added to many various aspects of life. A major successful application of gamification is in education, such as Duolingo, which is an application that motivates people to learn new languages using fun interactive methods. Other applications can be found in health and business [7]. Research shows that gamification has positive effects on people, taking into consideration the individual differences [6]. Tasks in software development can often get tedious and demotivating; moreover, the use of administrative tools is tiresome for developers to complete. Unmotivated developers can put projects in jeopardy, as the human resource is very important to the success of projects.

In the Agile environment, where development tasks are tedious and do not contain new activities, there is a great need for a method to inspire the development team to stay productive and creative. The Agile iterative and incremental nature make it a popular methodology in software development. The Agile methodology has been used to create higher quality software in a short period of time. Therefore, the focus of the thesis was to fit the Agile projects. [8]

It is not easy work for managers to continuously maintain the team's principles and performances. There have already been made several attempts of increasing motivation in Agile teams [9]. It is feasible to integrate gamification into Agile projects because of its iterative nature. Thus, driving team members to achieve the tasks of every iteration seeking self-improvement and engagement.

This research aims to provide insights on how gamification of the development process benefits the administration in management and performance of team members in a software development project, and how to implement it within the software project's needs. The research is based on a critical review of available resources, and adaptation of game-thinking models to the development process of the software projects. Furthermore, this investigation will also demonstrate related work, followed by an overview of the project and the

contributions that the proposed system adds to the gamification research topic supported with a case study. The game design concepts and mechanics will then be explained, using the 6D framework by following the six steps approach [6] then finally ending with a conclusion.

1.1 Problem Statement and Motivation

Software engineers' work of line is monotonous, and provides little social interaction. Gamification is a tool used to change and increase the motivation rate of team members to achieve optimum productivity and creativity in projects.

As mentioned in the previous section, the Agile framework is based on iterations, more user involvement and quick change of software requirements. However, the flexibility of Agile methodology has reflected complexity in the management of the projects. Managers of Agile projects need more time to manage these rapid changes more than traditional projects for many challenges. These challenges include the undocumented and unagreed upon requirements, the non-fixed variables of the project time, cost, and deliverables. Project Management means not only managing the tasks and the product but also managing the people. Everyone response and behave differently at work. Most people do not want to do anything that they consider unnecessary unless there is a profit from doing it. It follows that the main problem lies in people's motivation and performance.

In software projects, one of the software engineering lifecycle stages is "implementation", which is usually the coding part. Developers have had many issues regarding working with development tasks; the nature of development tasks usually do not include any exciting activities or social activities. Gamification is not a new method, it was used in education and health in a number of researches and applications, and the results of implementing gamification in such fields gave great results and feedback in terms of performance and productivity. However, considering the use of gamification in real-life work situations is a new idea that is full of challenges [10]. One of these real-life work environments is software projects. Due to the nature of the tasks in these projects, gamification could be applied to motivate developers and support evaluation of their work.

1.2 Research Questions

- At what level integration of gamification enhance the performance and engagement of team members?
- How gamification method may support the management of teams in Agile software projects to evaluate the performance of the team members?

1.3 Research Methodology

This research relies on a thorough review of literature resources on the topics of gamification and project management, specifically the gamification of software development projects. The review identifies gaps in literature when it comes to software development as an application field; the demonstration of the focus in current publications is on measuring the benefits of gamification through quantitative measurements that reflect user performance and productivity. However, the available researches fail to address many qualitative aspects such as tasks details and complexity, work environment, and the less obvious benefits of gamification such as those benefits to team behavior, engagement, and motivation. To shed light on the lack of comprehensive investigation in gamification and how to benefit from its results to evolve the developer's skills and engagement. Therefore, in this research, the six-step approach to gamification proposed by Werbach and Hunter [6] is applied to the development process of software projects. The approach is justified using a case study of developing a library system in which we measured the enhancement in the developer's performance and engagement after adding the gamification elements. For this, a group consist of twenty developers was formulated and a list of tasks was generated. The case study was done into two iterations, traditional and gamified.

In order to calculate the results, the data of both iterations were recorded. Six parameters from the proposed metrics of achievements were calculated and analyzed by calculating the Mean and the Standard Deviation of both iterations.

Lastly, a comparison of the data was compiled that demonstrated the increase of the performance rate of developers using the proposed research project.

Therefore, the contribution of this paper is a systematic approach to gamifying the development tasks in a software project, highlighting the benefits gained for employees and management alike.

1.4 Research Goals

- Applying the six-step approach to gamification (6D Framework) proposed by Werbach and Hunter [6] to Software Development Projects.
- Investigating the motivation elements and game elements and evaluate the effect of adding gamification in increasing the developer's performance in software projects and enhance their productivity.
- Fill in the gap between the management team and developers to enhance and support the evaluation process.

1.5 Thesis Outline

The thesis in chapter one gives an overview of the research and declares the research's problems, motivations, goals, and contributions. Chapter two gives reviews for some studies in the same context. In chapter three Game Design concept and elements are highlighted, as they are used in the research. In chapter four the design methodology and implementation of gamification are described. Chapter five discusses and analyzes the results of the case study and the Usability Testing results. Chapter six summarizes the research and the research conclusion.

Chapter 2 Related Work

Introduction

In this chapter the main topics that the research related to will be explained and discussed. The topics include managing the information technology projects and gamification literature review.

2.1 Project Management

According to the Project Management Institute (PMI) [2] the main challenge in creating successful project management is the execution of the project goals under some restrictions, for example, limited time, quality, budget, etc. [2] There are many factors, which lead to a successful project at different important levels. One of the fundamental factors of success is having an engaged, motivated team, which is our focus of this study and project [2]. Project scheduling is the part of the project that shows the project plan and the list of team responsibilities (tasks). The project plan contains all the processes in the project and the resources needed for every process. Those resources can be human resources, time, or cost. Many projects use online project management software services to keep the project on track, and to organize the project deliverables. It also proves to be more advantageous for real-time updates of tasks. Companies use task management software to support the evaluation process of the project and the team members. Project scheduling is considered a critical step that leads to a successful project and better outcomes. Furthermore, for team members, this step is the work plan in the project, which shows how important it is for their overall success. The team atmosphere and the team environment are the starting points to enhancing project performance [11].

Motivation is the key to productivity in team members and one of the well-known motivation techniques is rewarding. There are two types of rewarding, formally and informally. Management of teams using rewarding systems is an immense responsibility**.** “Most project team members are motivated by an opportunity to grow, accomplish, be appreciated, and apply their professional skills to meet new challenges.” [2]. In software projects new opportunities and challenges help expand work experience and are the motivation points for developers. They view seeking new skills as one of the most important goals for them on a professional and personal level. Moreover, for every team member to be valued in the organization they work is a factor of increasing their loyalty to their work, also translucence

evaluation satisfaction. Providing up to date results of the project gives more opportunities to compete and be creative.

2.2 Agile Methodology

Agility is the new trend for business organizations to manage software development projects. Software development process includes many activities, in order to deliver a project from an idea to a working product delivered to the customer. The software development process is divided into mainly four stages: Software Specification, Software Design and Implementation, Software Validation, and Software Evolution [12].

Agile methodology introduces an iterative model, where the output of each iteration is a release of the software. This output is aimed to be a minimally viable product (MVP) meaning the minimum work possible of the product which can be functional. In every iteration of the software life cycle, more functionalities and features are then added to the MVP. This iterative process will enable the user to evaluate the current deliverable and provide feedback. The feedback can be new requirements or changes on the existing requirements.

Scrum is one of the used Agile methodologies nowadays. Scrum is used to manage the project development, which is based on iterative cycles called sprints (each two to four weeks). Scrum has dedicated meetings, like daily scrum, sprint planning, and sprint review meetings, in order to keep the team focused on achieving the desired goals. The development team in Scrum decides on the practices to be used in the development phase, and how the development will be done. [12]

In general, the development team of Agile projects coordinates work through frequent informal interaction among themselves and with customers. Scrum teams use visual board to manage the tasks, like in Kanban [13]. The simplicity and flexibility of the Scrum methodology make it uncomplicated to adopt by teams, also for a new addition in the process [12].

Strode et al in his research proposed a model of coordination at the team level in Agile teams [14], the study presents a strategy to coordinate an Agile software project containing three categories of coordination, mechanism – synchronization, structure, and boundary spanning. Strode et al research addresses that managing a large-scale project is not as simple as its in

small projects, managing large-scale projects requires more time and effort, in this case, there is a need to manage interfaces between teams by doing “Scrum of Scrums”.

2.3 Gamification

There has been growing interest in gamification research within the context of software development. These studies do not only focus on gamification in general, but also look at gamification for specific applications. However, gamification studies are still a new trend, and according to Kumar, J. M., and Herger it is not that easy to find relevant publications or articles on this topic. It is also difficult to find contributions that highlight the application aspect of gamification in a software development environment, e.g. in the shape of a real task management system that manages engineers using a gamification technic [15].

The first step to design a gamified system is to understand gamification; Hamari defined gamification as enhancing user interaction with any system and providing the user with achievable and daily challenges to make them feel more engaged [16]. Zichermann, G. and Cunningham, C. defined gamification: “The process of game-thinking and game mechanics to engage users and solve problems” [17].

Werbach, K. and Hunter, D., 2012 categorized gamification into two types, Internal Gamification like company development, Human Resources improvement and work efficiency, and External Gamification, for education, health and propagation [6].

A systematic literature review of empirical studies in the field of gamification found that gamification produces effects on users on three levels: the “motivational affordances, psychological outcomes, and behavioral outcomes.” [18]. Another research by Paul Denny, studies the virtual achievements and game elements effect, in his research a badge-based achievement system within an online learning system. Denny showed a highly significant positive effect of badges on the number of students’ contributions without reduction in their quality. Students enjoyed earning the virtual rewards [19].

Another study that highlighted the benefits of applying game elements is in the work of Daniel and Giordano [20]; their work is in the educational field and their focus is on students as creators of content. They explain the gamification process as a three-step approach. First, analysis, which is defined as searching for the best gamification technique in the available context/application. Second, is known as integration, which is selecting the best mechanism

to introduce and integrate the selected gamification technique. Finally, evaluation, which is selecting a mechanism and a set of metrics to evaluate the results of different teams after the introduction of the game elements. The analysis step is deemed the easiest one according to the study. However, the integration of the game elements within the context and applying it to achieve its purposes is a more complex task. Similar to the work of Hamari [16], the work done by Daniel and Giordano [20] shows that gamification has positive effects on the creators, students in this case, and on the outcomes. The productivity of the student teams who benefited from gamification elements in their educational experience was higher, and their engagement and participation were improved. They explain this by pointing out that students track other students' achievements, and this makes them more competitive and committed.

An Additional relevant study "Educational Gamification vs. Game Based Learning: Comparative Study" showed that gamification was an innovative approach in the learning field [21]. They also described advantages of gamification lied in the fact that it is low-cost to implement. Furthermore, they found that gamification made learning content more interesting, and engaging.

Recently, the popularity of using gamification in development project increased, the increase came as a result of the positive outcomes in the education field, many researchers have been dedicating special attention to it. Unkelos-Shpigel offers a specific solution through the scientific work in "Gamifying Software Development Environments Using Cognitive Principles" [22]. This scientific work was based on categorizing the users into predefined classes: the creators (architect, programmer, customizer), and the reviewers (architecture reviewer, code reviewer, tester). Creators are responsible of their work, and the reviewers give points. This research is modifying the process of software projects, giving expert developers the power of the game elements and not automatically calculating the points. Users in the context of software development projects are mainly parallel to the role of "creators," however, it is emphasized that the effects are highly dependent on the context of the gamification and also the effects on the users.

Gamifying Software Development Scrum Projects research also show another use of gamification by presenting a software with gamification (add-on in JIRA software) to gamify the Scrum techniques to make it more engaging and interesting [23]. They created a prototype and began using it in teams to evaluate it and planned to collect more data. In

addition, they showed the lack of motivation in Agile practices, and explained that the studies still lack empirical validation of applying gamification and test the solution on actual software project environment. A similar application was done entitled “Gamification in the SCRUM Software Development Framework” [24], which presented the same concept, and results. A prototype was made to add gamification to the SCRUM software development, the results of the research show that the most efficient motivation elements are: money, vacations, personal success, and customer satisfaction. However, the prototype of the proposed system is a separated system from the existing used software in SCRUM projects. Adding a new component into the SCRUM project can burden the whole project and project members. Whereas, incorporating instead of adding gamification into projects guarantees better integration of the software development with gamification.

With regards to the research “Gamification in Software Development Projects,” the popularity of game elements was presented, which was applied to assess game elements, Achievements, Feedback and Leaderboard. These were determined to be the highest in the study due to the reason that they were easily imported and produced successful results. The research focused on the game elements and did not elaborate further in the game implementation and design [25].

In contrast to what was proposed by Unkelos-Shpigel [22], the gamification process should be easy, straightforward, and should not create additional effort for either the creators of the game or reviewers (players). It should also not change the current project management processes entirely as this could lead to project teams refraining from applying game elements. The focus should be on adding the elements as an additional layer and ensuring smooth integration.

Trotzek presents an overview in “Gamification Fundamentals and State of Research and Development” [26] in which the concept of gamification is thoroughly explained from different aspects, including more application-wise explanations. The research shows that several gamified projects of business to help employees improve their knowledge and share it with colleagues. The research presented as an example Microsoft experience; Microsoft used a game called Language Quality Game to enhance the quality of its product localizations. Microsoft employees participating in the game can gain points and compete with other employees and teams. The game had almost four thousand participants “and a

large number of them described the process as enjoyable and even addicting" according to Werbach and Hunter [6].

Another related work is the "Turning Real-World Software Development into a Game" [27] it presents the concept of adding a gamification model to the software development process by investigating the number of tasks individually and per team. In this research the metrics of achievement was in covering how much work the developer does in the project, which dismissed more important details. The existing research in the area of gamification of the software development process focuses on quantitative measurements, which do not necessarily reflect the complexity of the task performed or the quality of the work performed to complete the task [27]. Therefore, more research is needed to cover this gap of implementing and designing the game in advance measures of the quality considering the nature of development projects. The focus is more on simply measured quantitative metrics, for example, lines of code, timing and etc. Indifference to those factors usually appears as an insignificant metric for the programmer with regards to the number of achievements lacking to cover the complex and subtle nature of the development projects. On behalf of software development related literature, gamification is highlighted in the work of Erick Passos [27]. He proposed a game design arranged in a challenge-punishment-reward loop that was decomposable, chainable and combinable; meaning the challenges needed to be breakable into smaller, less complex tasks, progressively rely on each other and lead to a learning curve. Therefore, they see that the game elements need to be organized as a set of ordered, hierarchical challenges often requiring several different skills from developers, and lots of teamwork effort to be overcome. From a game designer point of view, the authors suggest easily measured quantitative metrics, such as lines of code, number of tasks completed, duration of each iteration, etc. However, several shortcomings can be observed in the related work, for example, the lack of covering all aspects and data sources of the tasks and projects. In addition, deficiencies arose in integrating gamification to the projects. Every user in the task management systems needs to read feedback about his experience in the system. There must be a method to provide every user with data that demonstrated what he achieved today, and what areas of improvement may still be needed. Undeniably rewarding is important, even if it is virtual. There is a need to add challenges to the system to motivate the users with each other.

Brian Cugelman explained that the only thing that matters in gamification behavior is "behavioral medicine," and he presents mapping between gamification and users' behavior

from the point of health and medicine. When there is a goal in the system, the user starts to change his goals to fit the game goals and that then creates the desired behavior and motivation. Another example is that, when a challenge list is in the system the user starts to manage his time and plan to win the challenges by competing with his colleagues or himself. These points make the users' lifestyle better and interactive [28].

The “MDA: A Formal Approach to Game Design and Game Research” offers the general concepts for game design and mechanics, on which a solution is based [29]. In order to support management teams, it is important to understand the preferences of every developer in the team especially in diverse team members with diverse skills and personalities. This is a considerable responsibility to handle. Moreover, supporting decision making in this aspect is necessary not only for evaluation purposes, but also to provide more team members with a professional environment of understandability. All these facets are needed to propel better team performances.

The research presented in this thesis focuses on two concepts. First, application of the 6-step approach to gamification proposed by Werbach and Hunter [6] and adapting it to the development process of software projects to contribute to the available body of knowledge. That will act to address some of the aforementioned shortcomings of the lack in covering all data sources and analyze it to improve the measurements of achievements and productivity. Second, study the effect of gamification on software projects tasks of the developers’ productivity and the motivation to work.

Chapter 3 Gamification Design and Game Elements

Introduction

This chapter discusses the components of the gamification system and explains how we can use game elements to integrate gamification into software projects. This chapter will go through game frameworks analysis, game elements, and motivational affordance.

3.1 Game Frameworks Analysis

There is an explicit distinction between game and gamification design and their features. The purpose of the game design is enjoyment, while the gamification design points towards business goals.

Gamification design is complicated and connected to many factors. The designer of the game needs to study the player's behavior, daily routine, and the user interfaces that fit them respectively. In the gamification design and elements section, we will walk through the main components of the game design and the analysis of it with regards to our research purpose.

Gamification adds game elements outside the game industry, which means the game is created for a specific purpose, and this arises in the designing phase of the game. The design phase includes designing the gamification elements, designing the game mechanics, and defining the purpose of the game design.

Following the literature review of gamification design frameworks [30], the general process of creating a gamified system and creating a game will always be different as a general process, although there is a thin connection (not well defined yet) between game and gamification design.

One can understand a Game, by looking at two main groups. One group being the designers or developers that created the game and the other is the group of people who will purchase and use the game, or in other words, Players. Developers are the addressed players in the game of this study. It is of great importance to fill the gap between the designers of the game and its players because a game without players wanting to play is irrelevant. Thus, filling the gap between designers and players is a crucial aspect. The gap that the management teams and supervisors, also clients in some cases, are not fully aware of developers' achievement and work and in many cases fail to evaluate their work. This is failure is due to the many responsibilities that managers have leading to lack of time. Also, fully evaluating

employees is time costly. Thus, managers will participate and get involved in designing their project gamification layer specifications. However, this should be regarded by the game's designers. For example, considering both sides' perspective, seeing the game from the player's perspective and understanding what makes them want to use it. As a result, a comparison of using two of the game frameworks was made: MDA and DMC, these frameworks are mostly used in applying gamification.

3.1.1 MDA Framework

MDA framework is an outline that was developed by Hunicke, LeBlanc, and Zubek [29]. It is a formal approach to understand games, and build a link between game design and development, as well as game criticism and technical game research.

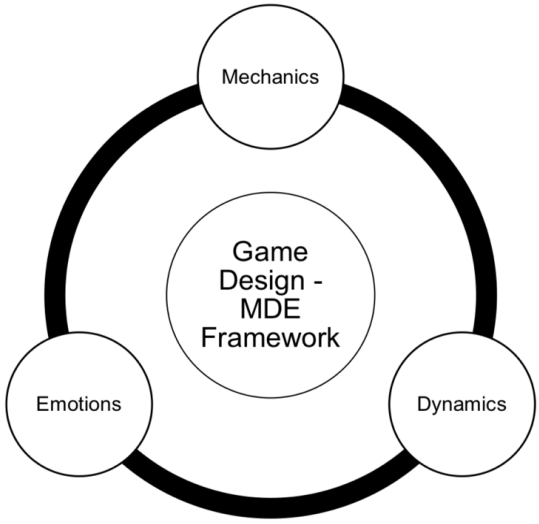

Figure 3.1: MDA Framework [29]

Figure (3.1) shows that the MDA framework contains Mechanics, Dynamics, and Aesthetics framework, which correspond to Rules, Systems, and Fun consecutively.

- Mechanics:

It represents the basic components of the game; it defines the rules and the actions that can be done in a game based on some algorithms. Applying that concept on the research

project, some initial rules were specified, for example, the system will be based on the challenge aspect in which developers will be able to gather points and move from one level to another. There will be different categories that will measure how well the developer is doing, depending on different metrics, these metrics reflect the developers' skill improvements, the ability to work in teams, delivering the tasks on time, number of tasks that were finished, and many others.

- Dynamics:

Dynamics discusses the system run-time behavior, operating upon the inputs and outputs of the system. In the software development process, the developers contribute to the work they do inside the project. Our approach is to use the data from the components used in software development as a source of input and develop an algorithm to keep track of their achievements, to continuously motivate them to perform better. As described in the Mechanics section, there will be categories that rule the developers' accomplishments in different fields. For each category, the developers will get badges, which will prove the level they are. Lagging developers will be given assistance, support, and direction in which they can develop themselves. For that purpose, a dash-boarding system is proposed to keep the developers updated about their performance and to give them instant feedback. This is accomplished by showing them their accomplishments, points, and levels in addition to an overview of the performance of the whole group.

This is the game design- the way the players observe the game when they are interacting and using it. It describes the desired emotional characters of the experience of playing the game. In this case, an enjoyable part, that is acceptable by the players, should be added to make the game more attractive and entertaining. For that, periodical challenges will be presented between developers to add a source of competing for fun away from the seriousness of the practical working life. Developers with extraordinary achievements will be shown on the dashboard. Incentives will be offered, in addition to advice and tips for self-improvement.

One of the essential aspects of inspiring motivation is to distinguish between intrinsic and extrinsic motivations. In other words, the difference between wanting to do something and needing to do it. Extrinsic motivation comes from outside the activity when a person is obliged or has to do a task. It can use the traditional way of reward and punishment to get people to do their work. However, intrinsic motivation starts from the people themselves,

and their need for self-improvement, which is on the top of Maslow's hierarchy of needs [11]. In Gamification, it is important to keep the motivations intrinsic, to let people improve because they want to do so, and not because they must do so. This is also included in the Self Determination Theory (SDT), which is based on human needs, by including the needs for competence, autonomy, and psychological relatedness [29].

3.1.2 DMC Framework

Werbach and Hunter developed the DMC framework in 2015 [31], it is also known as the DMC Pyramid. It presents a game as having three distinct parts: dynamics, mechanics, and components [32].

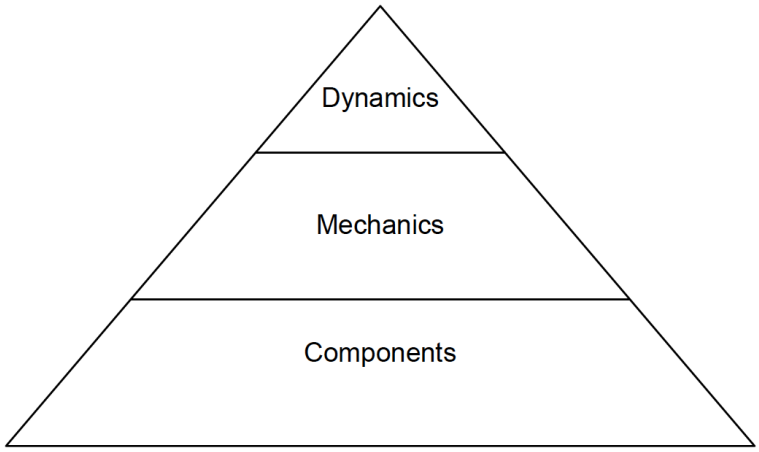

Figure 3.2: DMC Framework [6]

- Dynamics

  It is at the top of the pyramid, which is the highest level of game abstraction. It includes the most important game dynamics: constraints (limitations), emotions (competitiveness and curiosity, happiness and frustration), narrative, progression, and relationships. Basically, it is the game culture.

- Mechanics

  This part of the game refers to the techniques needed to move on in the game for the next steps, and it increases the players' engagement. It includes challenges, chance, competitions, and feedback.

- Components

This is where the list of components that represent the game dynamics and mechanics is addressed. It also represents the game results and visualizes the game goals, for example, points, and badges.

3.1.3 6D Framework

Hunter and Werbach presented in their book: "For the Win: How Game Thinking Can Revolutionize Your Business" the 6D framework [6]. The 6D framework includes the DMC framework but also includes concrete steps of designing a gamification system; starting from the definition of business objectives, continues to target the desired behaviors, describes the players; formulate the activity loops without forgetting the fun, and at the end, implementing the gamification system with the suitable tools.

This research tries to deal with the research problem with a gamified software tool to make software development tasks more fun and engaging for practitioners. Both frameworks were overlapping and although not in an exact way, it takes a slight impact of Hunicke et. al's MDA game design framework [29]. This can be noticed in the Pyramid of Gamification Elements, which is the basis for several other gamification design frameworks.

The 6D Framework is one of the most mentioned and more comprehensive frameworks to constitute the gamification design process [30] thus, the gamification solution of this research was designed by following the 6D Framework steps (6 step approach).

3.2 Game Elements

Many studies show the effect of gamification on increasing the productivity and engagement of students in the learning and education sectors. The reason is that games separate reality so it removes the extraneous factors that would be the outlet for employees from their traditional daily work-life [2] by creating a space for developers to do different types of activities. Game thinking suggests considering every activity as part of a game. If we want to define the variations between game and play, first, "gamification" relates to games, not to play (or playfulness), where "play" can be considered as the more widespread looser category, but is critically different from "games" [22]. In-game studies, the difference between games and play is tied back to the Caillois concept of "paidia and ludus" which are

two poles of play activities. These concepts broadly indicate the experiential and behavioral qualities of playing (paidia) “gamefulness” indicates the qualities of gaming (ludus) [4] leading to the main difference between game and play is the purpose and the outcome of each.

In research of Game Design Elements to Gamefulness Defining Gamification research [1] gamification was defined “as the use of game design elements in non-game contexts.” Play is defined as doing an activity within a game, and game is defined as the stage that happens after the start of an activity with a goal and a rule. The purpose or the goal of the game creates the essential motivation of the player. The research described the differences in the recent theoretical and empirical studies of the differences between playing and gaming, which leads to the various game designs depending on the purpose of the game. Gamification is a notion that combines both play and game. This means it can be entertaining and achieve as noted in the Figure (3.3) below:

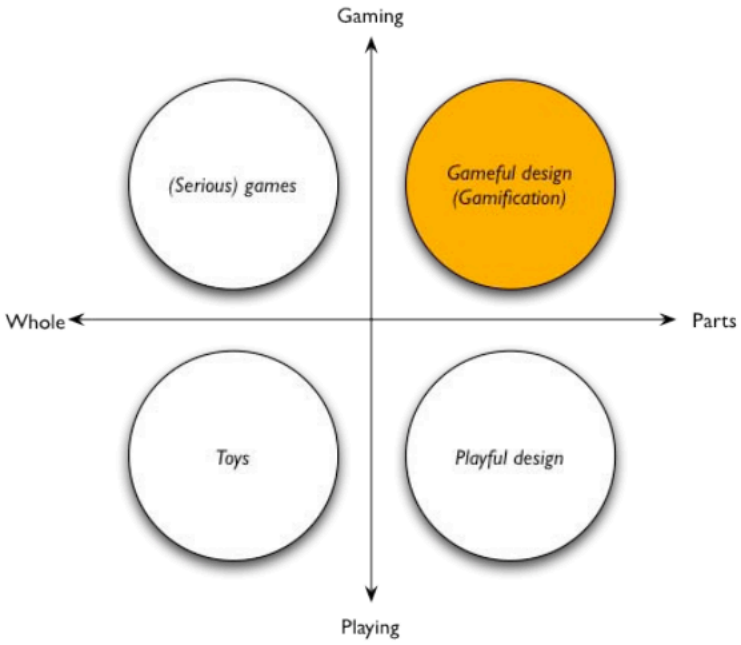

Figure 3.3 :“Gamification” between game and play, whole and parts [1]

Understanding the player is very important, the purpose is to establish the rules that support the gamification components. This stage of designing the game compound and elements because it is not easy to develop compound elements that are entertainment elements and at the same time support professional purposes leading to more enjoyable, engaging, and productive.

3.3 Motivational Affordance

Gamification adds fun to real work, game components are the fun-elements that can incentivize developers to work and to engage more in the workspace. According to the "Educational Gamification vs. Game Based Learning: Comparative Study" the most used elements in gamified applications are points, feedbacks, levels, and leaderboards. Here is a shortlist and description of some of the game elements:

- Points and Leaderboards:

User points within the system refers to what they achieved for the whole project time with their tasks. The points section could be variant between systems, but it is one of the main elements inside gamification and one of the most important elements in the feedback where the developers see that they earn those points. Erick, Danilo, Pedro, and Esteban [27] explained that every system needs points to act as the main elements of any game framework. The points section could be variant between systems, but still, it is one of the main elements of gamification. People like to be rewarded for what they do, and therefore people like games a lot. Finding the right way to calculate points will increase the user engagement with the system and enhance their daily work. Leaderboards are considered the most critical area of evaluation. Engineers view the evaluation presented on leaderboards as a challenge of a value that represented their work and achievements compared to their colleagues in the team or the company [27]. Results of the point evaluation are visualized on leaderboards, where every team member, category, effort and points achieved may be viewed. This allows managers to gain quick insights into the progress of the tasks, and to easily get an evaluation of the achievements of every team member.

- Achievements:

Achievements also sometimes known as a trophy, badge, award, stamp, medal, challenge, represents the user's accomplishments and performance. This allows the user to see his work immediately, and to view his achievements and results instantly.

- Feedback:

Feedback of every action and performance is needed in the gamified system. Every user should be provided with information on their achievements. Usually feedback is presented

as a list of details and analysis. It is an important factor because it allows a space to better understand the user's abilities and areas of improvement and displays the level of improvements.

- Rewards:

Usually, companies have their own rewarding systems and procedures set in place. It is not our purpose to try and change any of the existing processes of projects. We would stay with the existing rewarding system because gamification works to support those decisions and procedures. However, gamification adds more virtual reward, and that would create a positive feeling for developers.

- Challenges and Levels:

Adding the concept of challenges to the elements of the game in our project posed a challenge. To motivate the users to be more competitive is a double-edged sword. It is better to be used in specific events and not really for serious things inside the project to avoid any problems and frustrations.

Chapter 4 Implementation of Gamification in Software Projects

Introduction

This chapter includes all the specifications for creating the research solution proposed. It also covers the methodology used to add gamification to software development projects using a step-by-step approach. Finally, following with implementation design and specifications.

4.1 Applying the Six-Step approach to gamification of Software Development Projects

Gamification design is complicated and connected to many factors. In gamification, the focus is to make the players' experience more enjoyable and engaging. It is important for the designer of the game to study the player's behavior and nature of work and daily work routines. Gamification offers a way to add game elements outside the game industry, which means to make the game for a purpose, and this comes in the design stage of the game, which includes designing the gamification elements and mechanics and the purpose. Werbach and Hunter [6] proposed a six-step approach to gamify any project as could be seen in Figure (4.1)

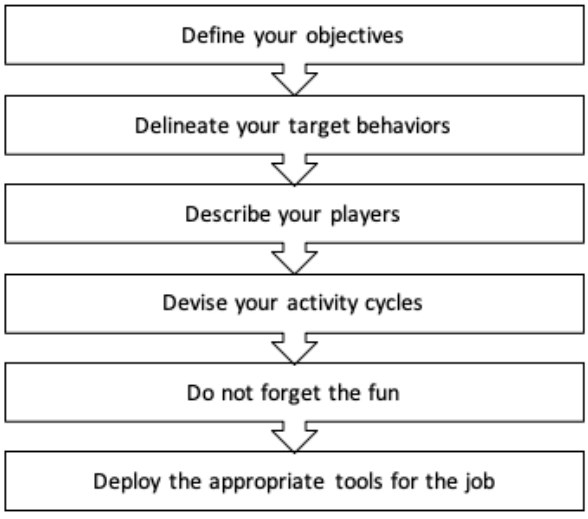

Figure 4.1: Six-Step Approach to Gamification

The author uses this approach to develop a step-by-step approach to adding gamification elements to the development process of a software project, in the proposed system we have followed the following steps:

1. **Define Objectives**

In this context, defining the objectives of the gamification comes first. The author see that it is important to derive goals from two perspectives, from the developers / “Creators” perspective gamification should improve the following:

- The engagement and sense of “fun” during the implementation of coding tasks. Which is achieved through creating gamified challenges in order to increase the productivity of the developers. This is done by incorporating the rewards system to be integrated into real social life.
- The gamification results will help the developers to recognize the weaknesses and strengths in order to improve self-improvement rate. The proposed system offers developers the chance to create focus groups from different teams by matching developers who share similar achievement level, and highlight the weak points in order to create a suitable training (workshops) or rewards. This all reflects on improving the productivity of the developers. Moreover, these activities support social interaction.
- The gamification challenges contain social challenges and events to improve the socialization between developers to complete the challenge. The rewards of the proposed system should not only be virtual rewards, but also include activities that different developers can communicate.

All of these steps are made to improve the sense of satisfaction with developers and themselves and the team environment. On the other hand, from Management Team “Reviewer's” Perspective: Gamification results support the administration and monitoring purposes by evaluation of employees' and teams’ performance in a project or a sprint. The results also support the evaluation the overall project, and this is done by observing the gamification results that reflect achievements of every team member. Gamification provides a real-time view of the results of every developer for every task allowing the managers and team leaders to monitor their projects in an advanced way. The gamification analysis and results support the managements’ decisions regarding the use of resources to achieve the

most optimal task distribution depending on the gamification recommendation results. It is also important for the developers to follow and adapt with new technologies and the rapid changes, the proposed system helps to the understanding of the individual differences, skills, competences, and training needs by presenting the level of the developer in every different aspect, for example, general level, timing, coding skills, self-improvement rate. This all can help managers to control and predict the amount of time and resources needed for future tasks and deliverables and therefore, improved project and sprint planning.

Those goals can be achieved over time and through several stages and gamification elements. The conflicts between goals, if any, should be resolved by setting priorities and starting with the highest priority goals.

2. Delineate target behaviors

This outlines the desired behaviors and deciding how to evaluate them.

With regards to the creators, the desired behaviors could be in the form of:

- Increased willingness to learn, e.g. by attending more training courses.
- Increased willingness to cooperate with other team members, e.g. by the number of tasks performed by more than an individual.
- Increased productivity, which can be measured by the number of tasks performed, lines written, duration vs. complexity of tasks achieved.
- Increased satisfaction of users, which can be measured by surveys and feedback of the developers and managers, in this research the focus is on the feedback of the developers.

With regards to the reviewer, part of the players, this could be in the form of reward systems. Adjusting motivational levels and psychological outcomes should be considered within those target behaviors. Almost all of the team project members aim to be motivated by taking an opportunity to improve skills, be appreciated, accomplish and use their professional skills to meet new challenges. In software projects, the new opportunities and challenges to expand work experience lie in the motivation of developers. Seeking new skills is one of the most important goals for them on a professional and personal level. Moreover, the point system achieved through gamification increases the transparency and accuracy of the team members' evaluations by giving a systematic evaluation, which is supported by the analysis

of the work they do and not a feedback from the manager or the team leader. This increases team members' creativity and sense of competition. After all, for every team member, it is important that their efforts are visible and valued in the organization.

3. Describe players

The focus for software projects is the list of players that are going to be affected by gamification: Developers/software engineers, Team leaders, Management team, and Human resource management. Every single player of the system will be affected. The developers are the main players of the system; their work will be analyzed and presented to the rest of the players and to themselves. Team leaders will be supporting the system by choosing the level and the priority of every task since they are more experienced. The management teams will have another view of the results; they will receive the developers, sprint, and task results with more details. Human Recourses are managing the rewards and support the achievement of the desired gamification results.

4. Devise the activity cycles

This step contains two steps, engagement loops, and progression stairs. As the discussion is about software projects there is a possibility to make the activity cycle by sprint which is not a long-term round and can give developers chances to improve on every new sprint which is between (3-4 weeks). Every new sprint will show how engaged the developers are in their work and represent their progress development in each sprint as a team and offer a comparison between them.

5. Don’t forget about the fun

One of the essential steps in the research is to provide immediate results for the user. The gamification layer should consider the capabilities of employees and the differences in experiences and abilities. Achieving this analysis of every developer is by inserting the priority of every task that shows what the most prioritized element of task is. For example, the time of delivering the task or the quality of the code (bugs free). Every user in the system needs to see feedback about their experience in the system; the game must give every user

an overview of what they achieved today and what areas of improvements are needed. Rewarding is essential even if it is virtual. Also adding challenges to the research project to motivate the users within each other. The challenges of the system will focus on developing the skills that have the potential to be improved. Also challenging other developers with a task of the same level and priority.

In this research, gamification adds fun to real work. Game components are the fun-elements that can increase developers' motivation to work and to engage more in the workspace. These components present the developers productivity as a self-improvement rate; also compare their productivity with the rest of their team members. This is descried more in the game elements chapter.

6. Deploy the appropriate tools for the job

The tools are needed to achieve the goals. They use the task management system with an integration with the code of the version control system used in the company. Additionally, to other systems for example, a code analysis system, gamification elements will be added as a layer (add-on) to be shown in the same task management system that is used in the project. Tasks management systems and code version systems are used in most software companies who are using Agile and these services are very important to manage the software projects.

4.2 The Technical Approach and Tools

Following the previous steps, every task is an entity that updates the developer's performance per project or sprint. The results are automatically calculated from the different sources, task management system, and version control system.

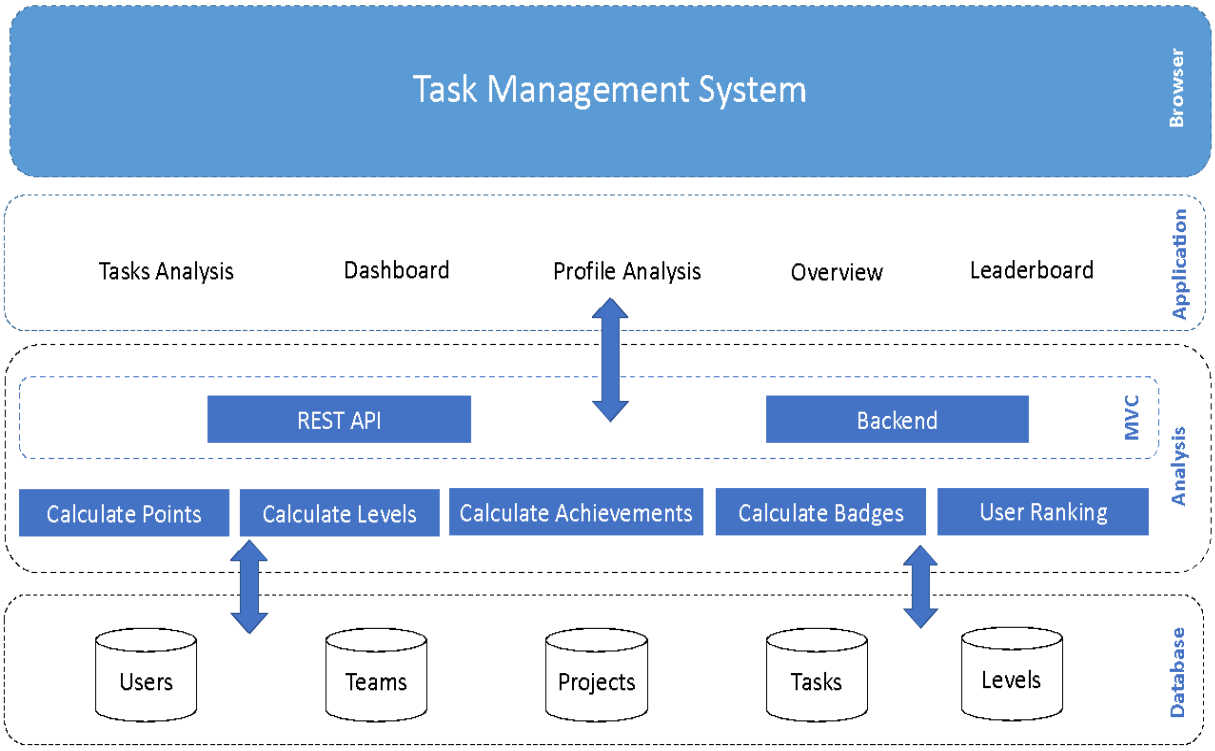

Figure 4.2: Technical Suggested Approach

Figure (4.2) describes the overview system structure, starting from user interface and ending with a database layer.

The proposed system is a web application that also works as an add-on to the task management system (ASANA) that was built using a PHP web server. The main component of the system is the Home Page, containing five modules (micro-services), each module is responsible for a specific request and each one was built using JavaScript for web interaction, and the PHP module used in the backend is responsible for serving the frontend modules. For example, the task analysis module is responsible for showing the user task achievements.

The next layer is the Server layer; it is responsible for listening to users' requests and handling them. The requests are handled by using a restful protocol for communication between frontend and backend. The Backend layer is chunked in multiple microservices, and each microservice is responsible to serve one thing related to task management. Each microservice contains its own database and they can talk to each other using a bus.

In Agile software development the work is performed in numerous iterations and there are continuous updates in the whole system project cycle. Every iteration of the project contains

a list of tasks; these tasks are divided between the members of the team. Gamification was defined as using Game Design Elements in Non-Gaming Contexts [33]. They presented a gamification layer that was to be merged to the Agile project management and development activities in the software industry. Our potential was created due to the work type and tasks in this industry. As we know most of the tasks are programming (coding) and these types of tasks do not require active activities so most of the programmers do not feel motivated or excited. The social activities are also limited, the research proposes adding a gamification layer to those tasks as an effective idea to change the typical way of doing projects. It could also serve as a way to see self-improvement aspects and work as a motivation tool to create a more productive environment to follow up with the game. The purpose of this research is to create gamification layers to follow up with the development tasks. We will study all the aspects of the development tasks in order to create our game design. Usually in Agile projects for software development main tasks are created in the project plan. In every sprint, the number of tasks to be done would be decided. The production team provides developers with new requirements, and a list of tasks is created and assigned. This happens in iterations usually every one or two weeks depending on the size of the project. The senior team (architecture, team lead) and the developers define some stays in the backlog, tasks description, assignee developer, deadline, and resources also. This list of tasks is listed in the tasks management system, so the developer can see tasks assigned to him and can comment and add his data to any task within the project.

On the other hand, the project manager and team lead can view all the information regarding every task for example, Task-list, Task timing, Task deadlines; Task assignee; Task level. Developers write their source code and add it to the version control software, for example, GitHub. Where developers can commit and push their code into it. Also covering the software for testing and bug management is a quality assurance team. In software projects usually many types of software are used, for this reason our study tried to cover the most important ones, coding, time tracking and test management.

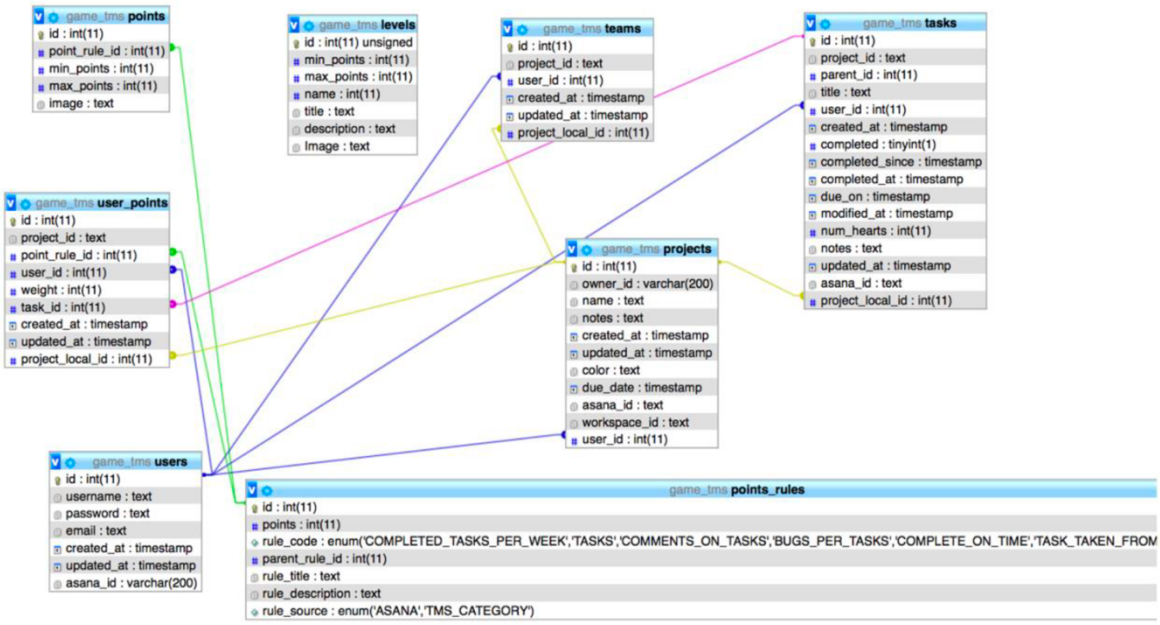

Figure 4.3: System Class Diagram

Figure (4.3) explains the class diagram of the proposed system. Moreover, many aspects of the performance of the work should be taken into consideration; the collected data should concentrate not only on the quantity of the work but also on the quality of it, and investigating whether it is effective or not, therefore, some data sources of the development tasks were initially chosen:

1. Task Management System

This service has been used in companies recently; its purpose is managing the distribution of the tasks and visualizing live-information of every task in the project. Usually, it provides details about the tasks. In this research, this service is used as a data source, also as a main base that will be used to implement the gamification layer. The data is extracted from the task management system, which will include a significant amount of important data about the project: Task-list, Task time, Task comments, Task descriptions, Task sub-tasks, Task deadlines, Task assignee and Task type. The data from the task management system in our project ASANA was collected from the APIs of the system as they are giving the possibility to use the APIs of the website. After this, the data was analyzed using the proposed Metrics of Achievements.

2. Code Version Control:

This is service is mostly used in software companies specifically. It functions to provide the system with more important information and data of coding tasks to analyze and to use in the suggested system. In this component the code and bugs are given. It is an important service of any software project. Code version control systems that manage the code source and bugs using repositories and having the developers as its main users, for example, GitHub and BitBucket. This would be covering the part of the quality of the code source, which will be providing a data source of more details about the tasks, the solutions, time of submitting, and bugs reported.

3. Code Analysis Systems:

This is a system used to analyze the code based on many factors like complexity, best practices, etc.

These services are used as data sources, the Code Version Control System, Task Management Tools, there is a need in software companies, and to facilitate managing software projects and teams due to the nature of these projects, also to manage outsourced projects, all of these factors created the need to use these services.

After analyzing the naturality of managing software projects, the implementation of the gamification layer will be in the task management system as an add-on due to the fact that this system is the service that manages all the tasks regardless of their type and contains all team members of every project and management teams.

One of the essential aspects of the research is to add gamification to projects without overhead. The project process and tasks will stay the same as usual with no disruption of any changes that may confuse the developers. The offered extension (add-on) will rather enhance the involvement of the users, mainly developers, by adding gamification elements, which will be used for proposing ways to enhance their performance by motivation, challenges, etc.

In the proposed research system, all data of the used services of the software project will support the implementation of the gamification layer.

4.3 Implementation of Gamification Elements in the Proposed System

Software development in Agile methodology contains iterations (sprints), the life cycle of the software starts from requirement specification ending with delivering a release of the product, the focus of this thesis is the development phase of the software cycle.

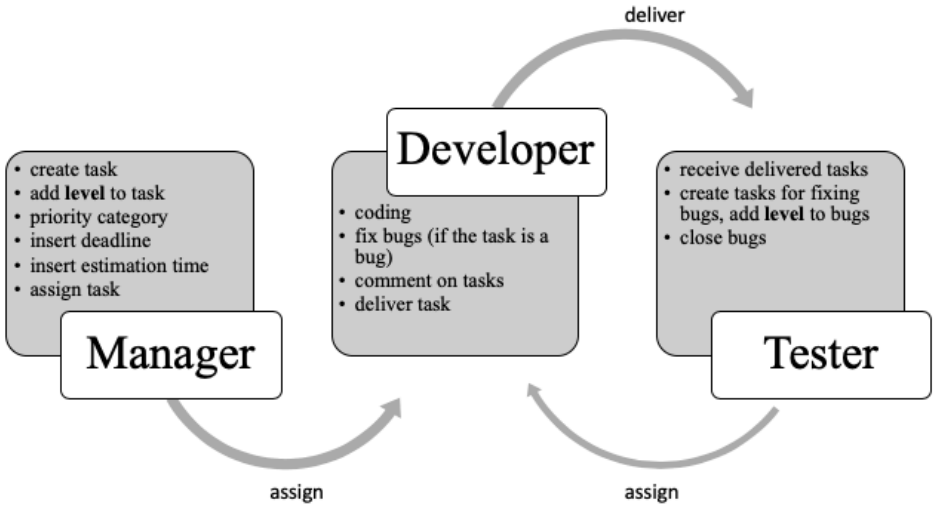

Figure 4.4: Adding game points and levels

In Figure (4.4), every project sprint goes through three stages, starting from creating the task list and the backlog of every sprint by the management team or product team. Every single task contains three categories; time, quality, and complexity. Each of these categories is rated in accordance to its priority on a scale from one to five, five being the highest prioritized and one being the least. The categories and priorities are decided by managers, team leaders, or agreed within the team. There are three main categories; the level of the priority represents how critical the priority is. The first category is timing, which represents importance of the noted time limit and the deadline of the task. Second, complexity, which shows how complex the task is. Lastly, quality which means that the tasks main priority is to be delivered without or with minimum possible bugs (critical functionalities), for example, tasks that is related to privacy or security.

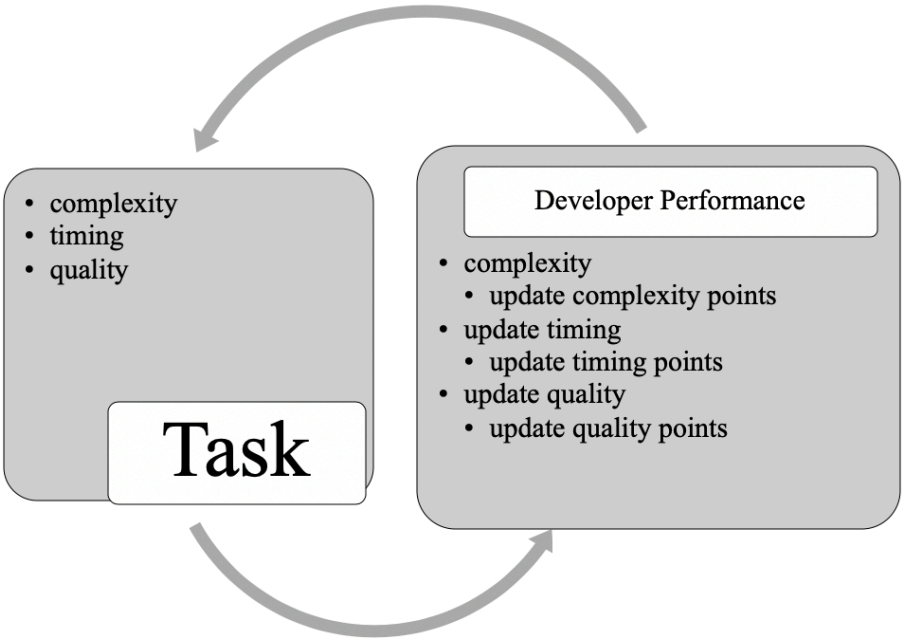

Figure 4.5: Update Developer Points

In Figure (4.5) explains the second stage in the implementation of the task, and in some cases, the task is fixing a bug. At the development stage, the developer will work on the coding part, and additionally, they will view the task categories, priorities, details, and how many points the task contains. Challenges and dashboards will be shown to developers.

After delivering the task, the task is received and tested by quality engineers. Then, if a bug is produced a new task is created then added to the task list. As discussed previously, this task will have listing of each category with responding priority levels.

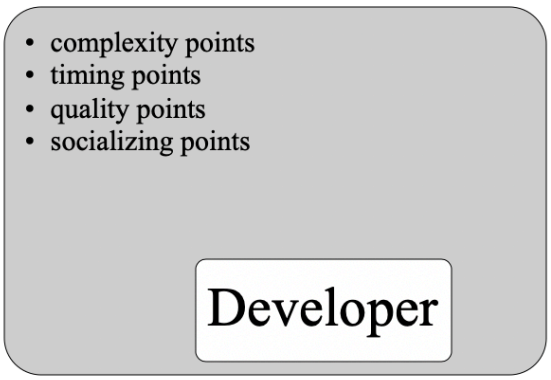

Figure 4.6: Developers Points

In Figure (4.6) every developer will get from each task new points depending on the priority level of every task they do. This calculation will be compared on every cycle with the previous cycles, in order to calculate the Level of Improvements of the developer, it is very important that the developers evaluate and see their performance giving them the chance for more improvements. However, this is not the only source for points; an additional source of points will be added from socializing the developer makes in the task management system. These points can be added via comments and mentions also questions to others and these will be called “Socializing Points” and has also levels that is connected to the badge “Socializer developer”.

4.4 Matrix of Achievements

One important aspect in measuring the achievement of the developer is not only measured upon the number of tasks done, but also to concentrate on the quality of the code completed, for example, some code analysis algorithms, or by comparison with best practices. Thus, different aspects were counted when determining achievements divided into many categories. Accordingly, the focus was distributed on both quality and quantity, which was presented in previous research [34]. The following table (4.1) shows the proposed metrics of achievements some of the rules that were used to measure the performance of the developers, also the category (aspect) of every measurement, and the data source:

Rules	**Category**
The task is done;	Time
Time of the task depending on the average time of same type tasks within the project;	Time
Number of completed tasks per period of time;	Achievement
Number of tasks developers work on / comment on/ involve in;	Socializing
Number of bugs in code/ after the testing stage;	Quality
The task is done before / after the deadline;	Time
Code Complexity;	Quality
Code practice, Code reuse;	Quality
Improvement per period of time;	Achievement

Table 4.1: Metrics of Achievements

The Metrics of Achievements is the key of evaluating the developer's performance, taking into consideration, the tasks, the project, and the team. The metrics of achievement parameters were chosen after developing the existing researches parameters. Each of these metrics have specific meanings in the software engineering context [27]. In chapter 6 the proposed metrics of achievement was tested and results were noted.

4.5 Gamification and Team management

Gamification influences not only the employees but also the managerial teams. Managers run several projects in parallel and have to consider aspects of cost, schedule, quality and customer satisfaction to ensure the success of their projects. As their work entails a lot of responsibilities, their ability to accurately controls the progress of tasks and notice the efforts, achievements and professional development of each team member in the development team is limited. In addition to the limitation in time that can be dedicated to planning social activities that increase the bonding and interaction among team members. Gamification saves, therefore, time and efforts for the management team and provides results with higher accuracy, the system provides the management teams with analyzed data per developer, team, sprint, and project. The game components, on the other hand, allow for the fun and social interaction between members of development teams. Games motivate people to organize and manage their time and tasks, also create an environment for social life activities and developers will tend to communicate more personally or virtually.

At the same time supervisors track every employee' performance per project, usually using reporting systems or simple reporting sheets. The proposed solution is covering and focusing on this area to support evaluation and rewarding of developers. That not only games make employees more active, motivated, social and competitive, but it can also affect management teams for a better and more clear understanding of the personal differences of the employees. It is from a technical point of view even the abilities, the weakness and strength aspects of every developer. Giving access to all the results data and the ability to view all details from a manager perspective can provide many benefits from the side of the status and distribution of the task, which reflects on the success of the project with fewer risks. From an employee perspective, many employees always complain about the evaluation and unfairness in the work. However, adding gamification can be a benefit to help avoid such problems, by providing the gamification results as a main reference of the evaluation process, and give more transparency between the employees and managers. For example, every developer has

a view of his work results, and his performance, quality, and time efficiency. The results are also shared with the management team. Managers' decisions and evaluations depend mainly on the outcome of gamification.

4.6 Prototype of the Proposed System

The system was created as an external web service that support the Task Management System. The proposed system gives the user the possibility to view personal information and game results also to view the project team members results. The provided mockups were used in the testing and were used to design the game elements. It includes User Profile, Dashboard, and some more details of the achievements' badges and numbers.

- User Profile

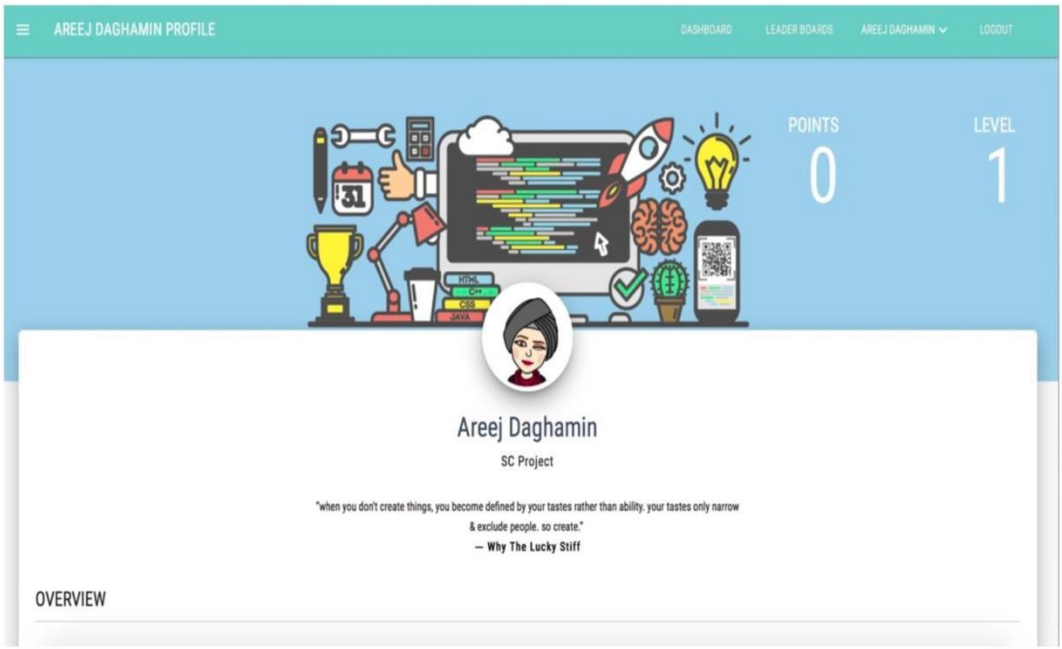

Figure 4.7: User Profile

Figure (4.7) shows the profile page in the proposed system. Profile is a good way to establish user identification. It also functions as a platform to share feelings and professional statuses of the day or in the sprint. One may use it see the points and levels as well.

- Dashboard

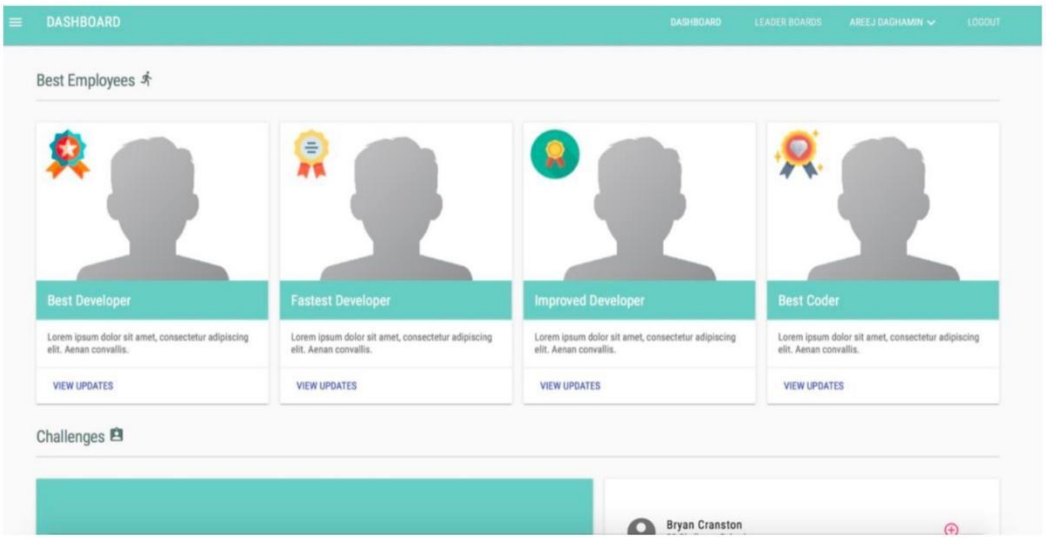

Figure 4.8: Home Page and Dashboard

In Figure (4.8) the Dashboard is the shared component between the developers in order to view in every sprint the badges and achievements. It is the main motivation and competition component; the developer's work may be observed by other team members and is updated in every sprint.

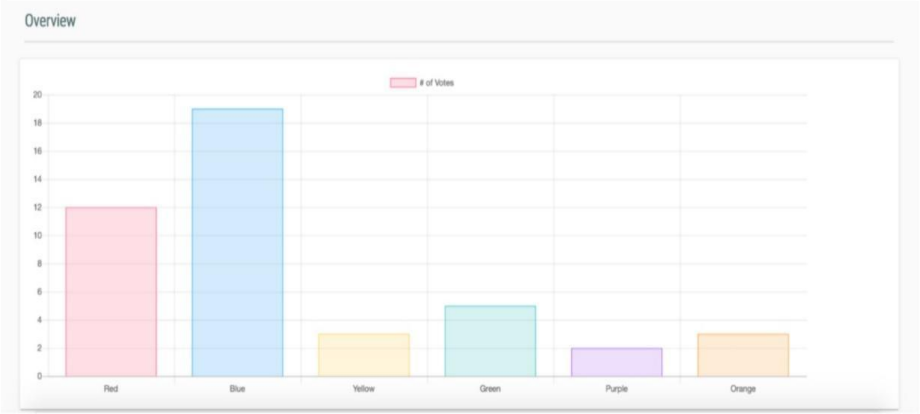

Figure 4.9: General Tasks Overview

Figure (4.9) shows the general overview, the overview is depending on the management team's decision if they choose to see the performance of every category (timing, quality) to be on every task of a project or a sprint with every task and every assignee.

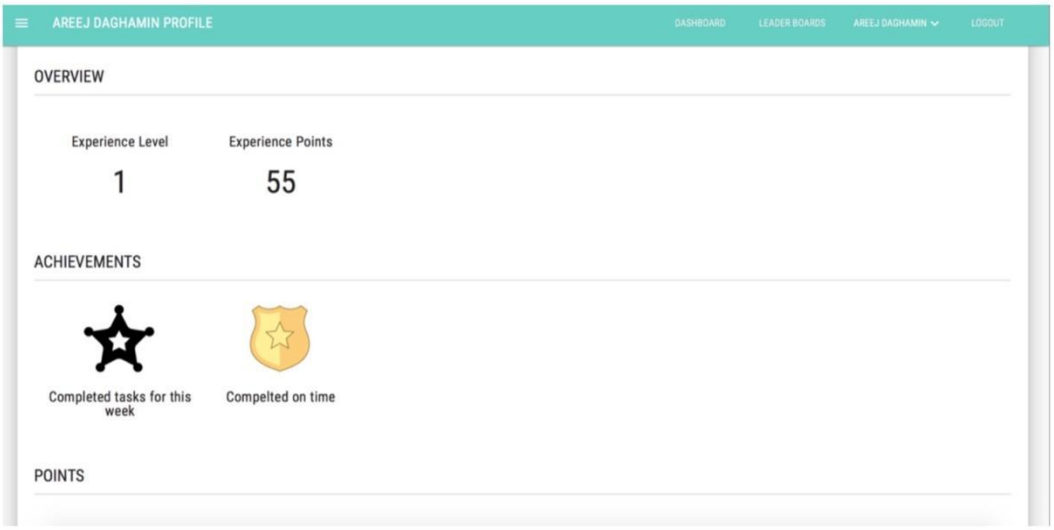

Figure 4.10: User Overview and Achievements

Figure (4.10) shows the User Overview with levels and points, every number of points is a level in different fields, this should be supported by management team and team leaders. This is usually done manually in software companies and every six to twelve months. Achievements of the user and badges are also collected and saved in the user overview.

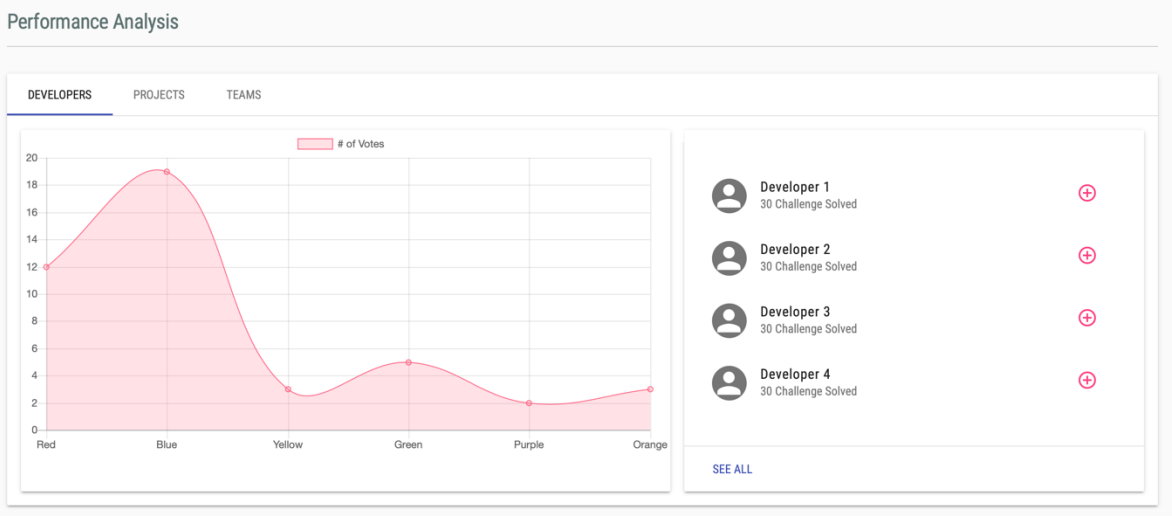

Figure 4.11: Performance Analysis

Figure (4.11) represent the performance analysis tool, this tool gives the management team the possibility to view the performance of every developer within a team and project by

clicking on the developer name, for example Developer 1 and then view his performance analysis. The performance analysis of every developer is shown per time.

Chapter 5 Case Study and Evaluation

Introduction

In this chapter, we will start by presenting the case study used to evaluate proposed gamified system. Then the results and evaluation of the proposed system.

5.1 Case Study: Library System

The case study is a web based online library system. The project and the technical environments were prepared in advance. The system objective is to ease the process of managing, borrowing, and reviewing books. The project implementation structure and plan were prepared before the session and was later introduced to the participants.

The library portal is an online web-based project, which presents services provided by the library administration. The main objective of the system is to enable the managers of the system to access the online website to add, delete, and update books. On the other hand, the site should enable customers to search, view, and borrow books. Each book will have a title, ID number, description, and price.

The system is prepared to contain the following main functions:

- Registration System: enable admin to create and confirm profiles created by other users on the system.
- User profile: enable users to modify their profile according to their characteristics and authorization.
- Searching: enable the user to search for books.
- Add, delete, update a book
- Display existing books description and information.

The user's classes and characteristics are as follows:

- Administrator: represent users who have the privilege of creating, editing, and deleting books.
- Registered user: has the ability to apply for any service available on the portal.

- Visitor end-user: be able to view the home page of the library and browse through different services available.

The projects tasks were outlined and reviewed by an expert Senior Engineer for technical consultation.

5.1.1 Participants Profile

The case study included twenty developers. The developers who participated in the case study are working in IT companies in Palestine. The developer's specialization and level of expertise were various, to fully fit the requirements of the tasks and made the study comprehensive.

Technical Position	Expertise Level	Ids	Count
Frontend Developer	Senior	1, 2	2
	Mid-level	3, 4	2
	Junior	5, 6	2
Backend Developer	Senior	7	1
	Mid-level	8	1
	Junior	9, 10	2
Software Engineer	Senior	11, 12, 18	3
	Mid-level	13, 14, 19, 20	4
	Junior	15, 16, 17	3

Table 5.1: Participants Information

Table (5.1) shows the participants' information; the participants were ten software engineers, six Front-end developers, four Backend developers. Seven of them were juniors, six seniors, and seven middle levels. The participants in total were twenty developers.

5.1.2 Tasks Distribution

Every task was analyzed and prepared. Twenty tasks were prepared and distributed to ten groups. Each group needs to implement two tasks of the same level of complexity, time, and quality. (see Implementation chapter 4)

#	Group	Task description	Quality level	Time level	Complexity level	Duration (in minutes)	Phase Type
1	Group 1	Create user registration	5	3	3	120	Gamified
2		Create user profile	5	3	3	120	Traditional
3	Group 2	Create UI for Registration	2	4	2	60	Gamified
4		Create the UI for Users Profile	2	4	2	60	Traditional
5	Group 3	Create user login function	5	5	2	80	Gamified
6		Create a function: “add new book”	5	5	2	80	Traditional
7	Group 4	Create a function: “search for a book”	3	4	3	45	Gamified
8		Create a function: “update a book”	3	4	3	45	Traditional
9	Group 5	Create a function: “show book description:	3	3	3	30	Gamified
10		Create a function: “print book summary”	3	3	3	30	Traditional

Table 5.2-A: Tasks Distribution and Details

#	Group	Task description	Quality level	Time level	Complexity level	Duration (in minutes)	Phase Type
11	Group 6	Create a function: “delete user”	5	2	4	30	Gamified
12		Create a function: “delete a book”	5	2	4	30	Traditional
13	Group 7	Create a function: “print a book”	4	2	2	40	Gamified
14		Create a function: “print book summary”	4	2	2	40	Traditional
15	Group 8	Create a function: “borrow a book”	3	2	4	50	Gamified
16		Create function: “show list of books”	3	3	3	50	Traditional
17	Group 9	improve performance 18of adding new book	4	2	5	60	Gamified
18		improve performance of searching for a book	4	2	5	60	Traditional
19	Group 10	Create a log-out function	4	5	2	20	Gamified
20		Duplicate a book	3	3	3	20	Traditional

Table 5.3-B: Tasks Distribution and Details

Table (5.2) shows the tasks list that the developers implemented in the session. Every task has a descriptions and estimated duration. The table also shows if the task was done in the traditional phase or the gamified proposed system. Every single task contains three categories; time, quality, and complexity. Each of these categories is rated in accordance to its priority on a scale from one to five, five being the highest prioritized and one being the least.

5.1.3 Case Study Methodology

The prepared list of tasks contains twenty tasks, ten of them were done in the first iteration ‘Traditional phase’ and ten tasks were done in the ‘gamified phase’. Each group were assigned two tasks, one to be implemented in the traditional phase while the second one to

be implemented in the gamified phase. These two tasks were chosen so they are of similar with respect to level of complexity, time, and quality.

In the first session the traditional phase, the developers were asked to implement the tasks and were given the deadline of submissions. The ten tasks were implemented, the time and results were recorded. The developers were allowed to communicate as they are working in a project. The deadline and the estimated task duration were given to developers.

In the second session, the gamified phase, another ten tasks were introduced to the developers. Additionally, in this phase, the list of tasks was added to Asana [35] and the developers got access to the proposed gamified system. The gamification, game elements, and how the system works was explained and introduced. The developers in the gamified phase were also received points and badges immediately that were presented on the dashboard. During this iteration, the developers can observe their results on the dashboard.

In order to calculate the results, the data of both phases were recorded. Six parameters from the proposed Metrics of achievements were calculated and analyzed.

The following six parameters have been examined:

- x1: number of delivered tasks;
- x2: task duration in minutes;
- x3: delivered tasks per 1hour;
- x4: number of tasks developers work on / comment on/ involve in;
- x5: number of bugs in code/ after the testing stage;
- x6: number of tasks finished before the deadline;

5.1.4 Case Study Results

In the following sections, we explore the results each of these parameters.

1. **Number of delivered tasks (x1)**

The first parameter was the number of delivered tasks in the whole project (the online library), the developers have done ten tasks traditionally and ten similar tasks but gamified.

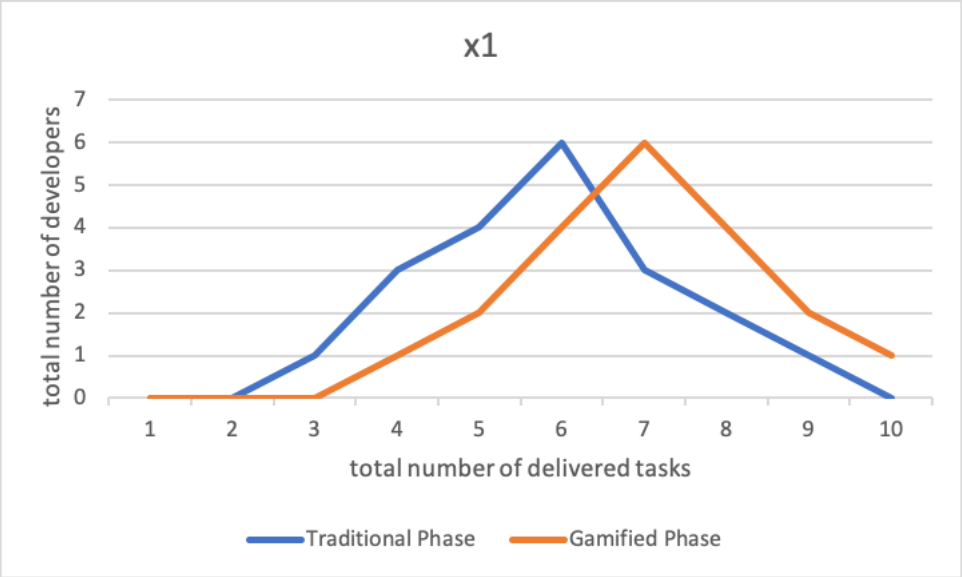

Figure 5.1 x1: Number of Delivered Tasks

Figure (5.1) shows the x dimension represents the total number of the delivered tasks and the y dimension shows the number of developers who delivered the tasks. The figure shows the analysis of the number of delivered tasks and the difference between the two phases. The blue chart shows the results of the traditional phase. In order to measure the enhancement of the performance we have calculated the mean in the traditional and the gamified phase for this parameter as shown in the Equation 1. The mean in the traditional phase was 5.8 and in the gamified system become 7. As we can see the performance is improved as the mean increased which refers to the number of tasks finished by each developer. It is also noticed that the developers who delivered only 3 tasks at the traditional phase were all improved. It is shown in the gamified phase that no developers have delivered only 3 tasks.

$$m = \frac{sum\ of\ delivered\ tasks}{number\ of\ tasks}$$

Equation 1: Mean

The improvement was not only of the performance of the developer but also, the developer's achievements was close to each other, which is reflected in the standard deviation. The Standard deviation in the gamified phase was 1,5310. The standard deviation was calculated following the Equation 2 below.

$$\sigma = \sqrt{\frac{\sum(\chi_\iota - \mu)^2}{N}}$$

$\mathcal{N}$: $stand\ for\ the\ number\ of\ developers$ 20

χ_ι: $stand\ for\ each\ mesured\ task$

$\mu = the\ mean$

Equation 2: Standard Deviation

x1 Traditional		**x1 with gamification**	
Mean	5,85	Mean	7,0675
Median	6	Median	7,38
Standard Deviation	1,531253357	Standard Deviation	1,53108793
Minimum	3	Minimum	4
Maximum	9	Maximum	10
Sum	117	Sum	141,35
Count	20	Count	20

Table 5.4: x1 Parameter Data Comparison of Traditional and Gamified Phases

Table (5.3) shows the results of all the developer's performance during the sessions and the two traditional and gamified phases. The maximum number of delivered tasks in the traditional phase was 9 and the minimum was 3. On the other hand, in the gamified phase, the maximum number of delivered tasks was 10 and the minimum was 4. The data shows the normally distributed developers' performance and it was checked using the normal distribution formula in Equation 3 below; refer to appendix A for more details.

$$\int(\varkappa) = \frac{1}{\sigma\sqrt{2\pi}} \mathcal{C}^{\frac{1}{2}\left(\frac{\pi-\mu}{\sigma}\right)^2}$$

Equation 3 Normal Distribution

2. Task duration in minute (x2)

Developers have worked in a group of ten tasks in the traditional phase and another group of ten tasks with the gamified system and elements. Every duration of time the developers take to finish every task was recorded; refer to appendix B, C for more details.

The first group of tasks was, 'create user registration' in the traditional phase. The gamified task was 'create user profile'. The estimated duration of time of both tasks is 120 minutes. The complexity level of the task is 5 the quality level is 3, and the time priority is level 3. The two tasks that shared the same attributes and were implemented by all developers.

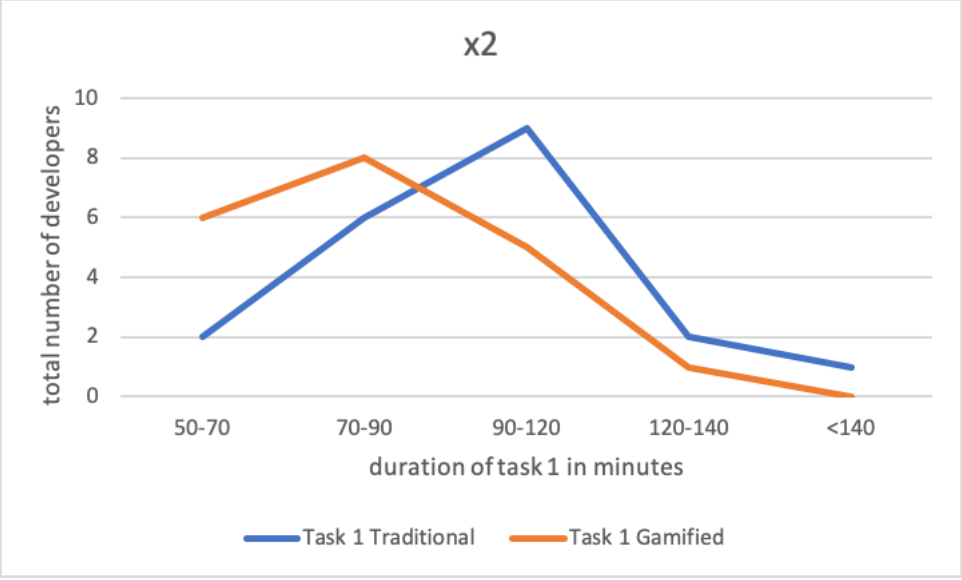

Figure 5.2: x2: Performance of developers' delivery of task 1

Figure (5.2) shows in the x dimension are the period of time in minutes and the y dimension shows the number of developers who did the task in each period of time. In the traditional phase, the curve is tilted to the right, which means that it tends to take more minutes. Whereas, the second curve is to the left, which means less time in carrying out the task.

Task 1 Traditional		Task 1 Gamified	
Mean	95,05086	Mean	80,79257
Median	94,57094	Median	84,38223
Standard Deviation	22,76322	Standard Deviation	18,31775
Minimum	57,23408	Minimum	52,03098
Maximum	146,8432	Maximum	133,4939
Sum	1901,0173	Sum	1615,851
Count	20	Count	20

Table 5.5: x2 Parameter Data Comparison of Traditional and Gamified Phases

In Table (5.4), in the traditional phase the minimum duration of time the developers did implement the task was 57,2. In the gamified phase, the minimum duration was 52,0 the difference was not higher as the mean in total due to the reason that this was done by one of the Senior Software Engineers. The mean is 95 minutes and in the gamified, it is lower 80 minutes. Meaning that the tasks took less time in the gamified phase than in the traditional phase. It was noticed that the enhancement of the senior developer's performance was not high. However, the enhancement was noticed more in the middle-level and juniors; refer to appendix D for more details

3. **Delivered tasks per an hour (x3)**;

For this parameter, we have made the 1 hour as a point to measure the number of tasks delivered by each developer.

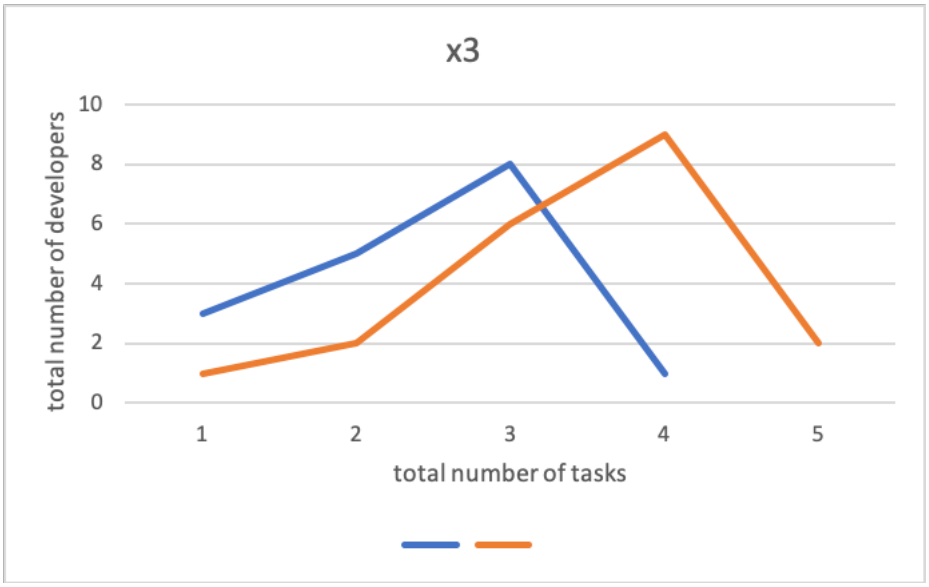

Figure 5.3 x3: Delivered Tasks per an hour

As shown in figure (5.3) the curve is tilted to the right in the gamified phase reflecting that the developer's performance was improving, the x dimension is the number of developers and the y dimension is the number of tasks delivered in an hour.

x3 Traditional		x3 with gamification	
Mean	2,58224	Mean	3,41273636
Median	2,86963	Median	4
Mode	3	Mode	4
Standard Deviation	1,16769	Standard Deviation	1,00893003
Minimum	0,60941	Minimum	0,69472727
Maximum	5	Maximum	5
Sum	51,6449	Sum	68,2547273
Count	20	Count	20

Table 5.6: x3 Parameter Data Comparison of Traditional and Gamified Phases

Table (5.5) shows the analyzed data from the parameter x3, the mean in the traditional phase was 2,5 and it was improved in the gamified phase to become 3,4; refer to appendix E for more details.

4. **Number of tasks developers work on / comment on/ involve in (x4)**

This parameter was to measure the engagement of the developers in the project. The number of tasks that the developers were involved in or commented on were counted for each developer.

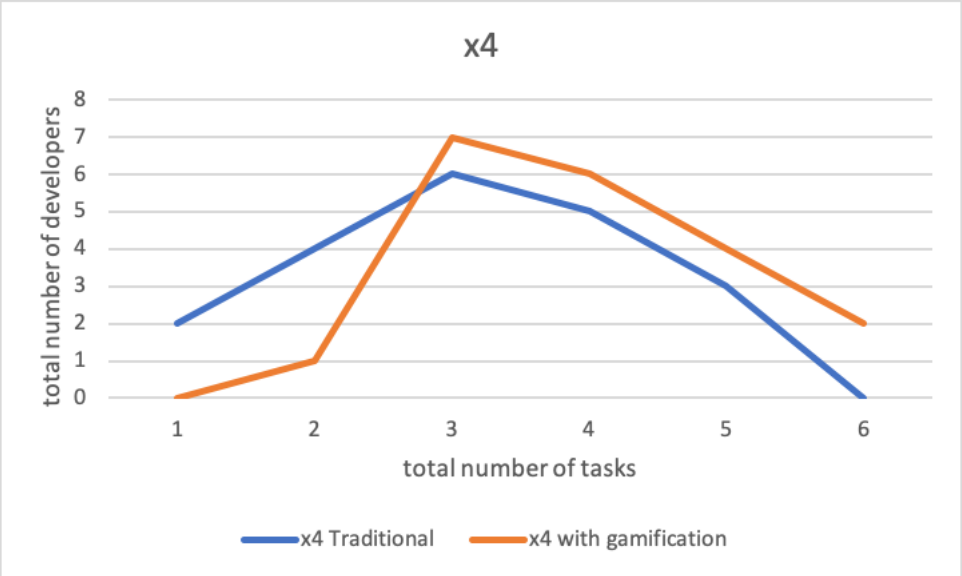

Figure 5.4: x4: Number of tasks developers were involved in

Figure (5.4) shows in the x dimension the number of developers and in the y dimension, the number of tasks the developers commented on or were involved in (tasks that are done by other participants). The chart shows that the developers in the gamified phases tend to participate more after knowing that this helps them reach the highest levels in the game.

x4 Traditional		x4 with gamification	
Mean	3	Mean	4
Median	3,3375646	Median	4
Standard Deviation	1,12530942	Standard Deviation	1,1145083
Minimum	1,1044948	Minimum	2
Maximum	5,0221341	Maximum	6
Sum	64,276185	Sum	79,877872
Count	20	Count	20

Table 5.7: X4 Parameter Data Comparison of Traditional and Gamified Phases

The Table (5.6) shows that in the traditional phase the minimum number of tasks the developers interacted with was 1 and in the gamified phase was 2. The mean in the gamified phase was 4 higher than the traditional phase which was 3; refer to appendix F for more details

5. **Number of bugs in code/ after the testing stage** (x5)

The focus was also to check the quality of the code of every developer. During the session, the delivered code was tested and checked. The problems were identified; the number of bugs the developer produced were counted.

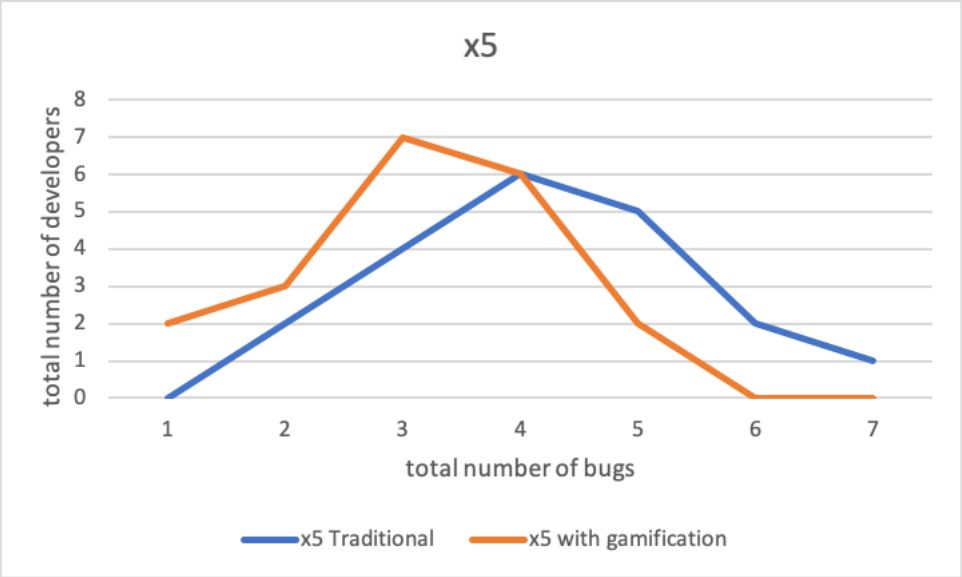

Figure 5.5: x5: Number of Bugs Developers Produced

Figure (5.5) shows the quality performance of the developers that it is reflected in the number of bugs they produce. The x dimension of the graph represents the number of bugs. The y dimension represents the developers' number. In the traditional phase, the curve is tilted to the right, which means an increase in the number of bugs. Whereas in the gamified phase the curve is tilted to the left expressing fewer bugs.

x5 Traditional		x5 with gamification	
Mean	4,187041673	Mean	3,197329827
Median	4	Median	3
Standard Deviation	1,344271755	Standard Deviation	1,164973064
Minimum	2	Minimum	1
Maximum	7	Maximum	5
Sum	83,74083346	Sum	63,94659653
Count	20	Count	20

Table 5.8: X5 Parameter Data Comparison of Traditional and Gamified Phases

Table (5.7) shows, that in the traditional phase the minimum number of bugs the developers produced was 2, and in the gamified phase was 1. The mean in the gamified phase was 3

lowers than the traditional phase which was 2. The indication that the quality of the developer's performance was improved in the gamified phase. The developers in the gamified phase have noticed the effect of bugs on the game results and have paid more attention to the quality of their work; refer to appendix G for more details.

6. **Number of finished tasks before the deadline** (x6)

We have measured in this parameter the number of finished tasks before the deadline following the estimated duration we have for every task.

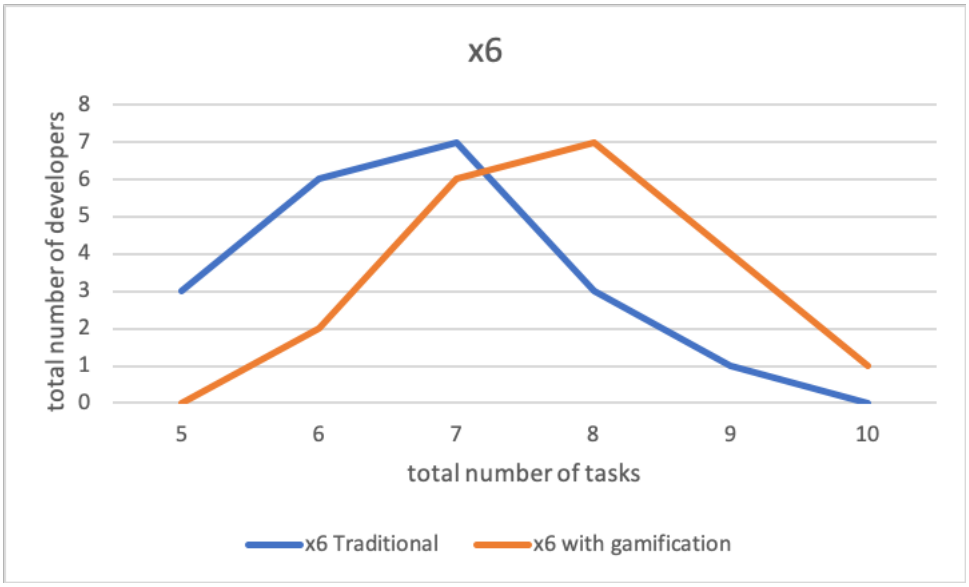

Figure 5.6 x6: Number of Finished Tasks Before the Deadline

In figure (5.6), the x dimension shows developer's number and the y dimension shows the number of finished tasks before the deadline. In the traditional phase, the curve is tilted to the left. Whereas in the gamified phase the curve is tilted to the right expressing a higher number of finished tasks before the deadline. The improvement of the developer's performance in the gamified phase was also observed in this parameter.

x6 Traditional		x6 with gamification	
Mean	6,68619256	Mean	7,8634419
Median	6,65610698	Median	8
Standard Deviation	1,06409267	Standard Deviation	1,04339768
Minimum	5,00220394	Minimum	6
Maximum	9	Maximum	10
Sum	133,723851	Sum	157,268838
Count	20	Count	20

Table 5.9: X6 Parameter Data Comparison of Traditional and Gamified Phases

Table (5.8) shows, that in the traditional phase the minimum number of finished tasks before the deadline was 5 and in the gamified phase was 6. The mean in the traditional phase was 6,6 lowers than the gamified phase which was 7,8. The indication that the developer's performance was improved in the gamified phase. The developers in the gamified phase have been informed that delivering the tasks on time will be reflected in the results of the gamified system; refer to appendix H for more details.

Factor	**Gamification Mean Value**	**Improvement (+ve, -ve, none)**
x1: number of delivered tasks;	7,06	+1,15
x2: task duration in minutes;	80,79	-15
x3: delivered tasks per 1hour;	3,41	+1
x4: number of tasks developers work on / comment on/ involve in;	4	+1
x5: number of bugs in code/ after the testing stage;	3,19	-0,98
x6: number of tasks finished before the deadline;	7,86	+1,17

Table 5.10: Parameters Comparison of Traditional and Gamified Phases

Table (5.9) shows the improvement of the parameters results in the gamified phase. The parameters x1 presenting number of delivered tasks, x3 presenting the number of delivered tasks per hour, x4 presenting the number of tasks developers were involved, and x6 presenting the number of finished tasks before deadline shows the positive increase of the developers' performance. In the other hand, x2 presenting the duration of the tasks in minutes and x5 presenting number of bugs, they showed a decrease, which indicates an improvement in developer's performance.

After finishing the sessions, the game elements and system output were shown to the developers. The points and badges were all collected depend on the parameters (x1, x2, x3, x4, x5, x6); refer to appendix I for more details.

5.2 Usability Testing

User experience is a new topic used to improve user enjoyment with software. It converts the hard stuff to easy stuff. A good product that reaches its full potential can be materially made by good design. On the other hand, to improve the user experience, there is a need to listen to the developers who would use the system and understand them, collect details and create a behavior usability assessment. Although, design software without the end-users will affect product delivery. To avoid such a problem interviews and testing sessions with the end-users should occur; observe the interaction between the end-users and a product then register all the usability problems. However, there is no need for hundreds of users to test a software because usually five users can uncover about 85% of usability issues Nielsen, Jakob, and Landauer, Thomas K [36]. Usability testing as a term used to evaluate software from a user perspective and it is a part of a larger effort to enhance any software. The main aspect of usability testing is to gather data from an end-user to make sure the software is useful. The main goals from usability testing: check if the product is easy to use, easy to learn and deal with; make sure to satisfy the end-user. In order to achieve the main aim of the current study of investigating the relationship between motivation and Gamification among developers and managers in software companies, also finding the relation between gamification and motivation.

5.2.1 Usability Testing Methodology

The usability evaluation of proposed system. During the case study sessions, participants were asked to spend 30 minutes filling the usability questionnaire. Questions were answered

after every section of the experience, discussing the current work environment, and the feedback of the proposed system, it is initial impression and overall satisfaction also more suggestions to be taken into considerations. Using our interviews and questionnaire to build the system and build it according to the feedback and comments took around six months to complete the needed reports.

5.1.1 Usability Evaluation

The starting point was to understand the current working environment for software development, this helps to better understand and evaluate the problem. Before and during the creation of the system, part of the study was made as a questionnaire for studying the current work environment, evaluation, and the acceptance of gamification ideas in software development projects. The first section of the questions addresses performance management in the developer's current work:

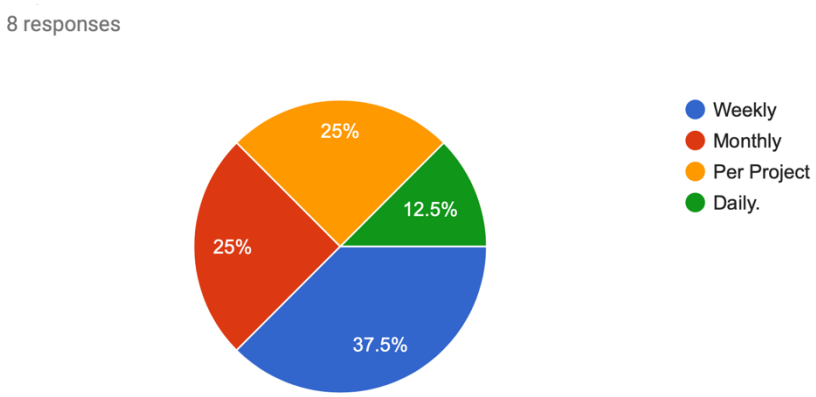

Figure 5.7 Reporting Durations

As described in Figure (5.7) there is a different timeframe to submit the reporting and the reporting was never dependent on the project or the sprint, but only related to a period of time. The proposed system divides and analyzes the tasks every sprint and shows the rate of the improvement in every sprint.

Questions were also posed to investigate the developer's satisfaction with regards to his current work performance evaluation.

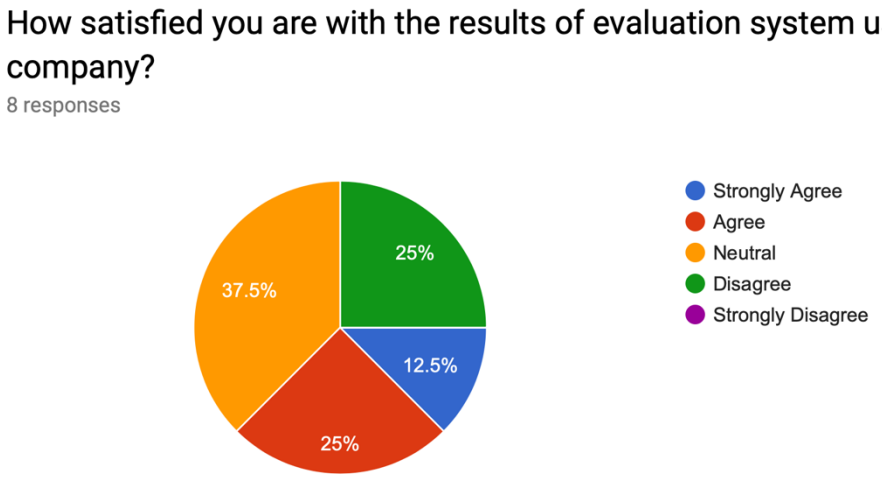

Figure 5.8: Satisfaction of Traditional Evaluation Techniques

Figure (5.8) demonstrates satisfaction after collecting the personal information of the developers. Questions regarding current employment status and situation were asked. Developers described their current work environment. Most developers were not satisfied with their evaluation system, some of them justified that the team leader always tended to a developer over another, which always reflected on the evaluation of their performance. Not only this, but also some described that team leads have no experience in the evaluation process and criteria. Also, the limited time of the team leaders and managers lead to some describing the feedback as not precise and not detailed which is reflecting the understanding of the problems they have and how to solve it. Other developers describe the process as good and fair enough. The proposed gamified system supports the team leaders' decisions and evaluations by viewing the results and achievements of every developer compared to each other in an unbiased model.

Later, questions were to investigate the developer's knowledge and opinion regarding game elements and gamification:

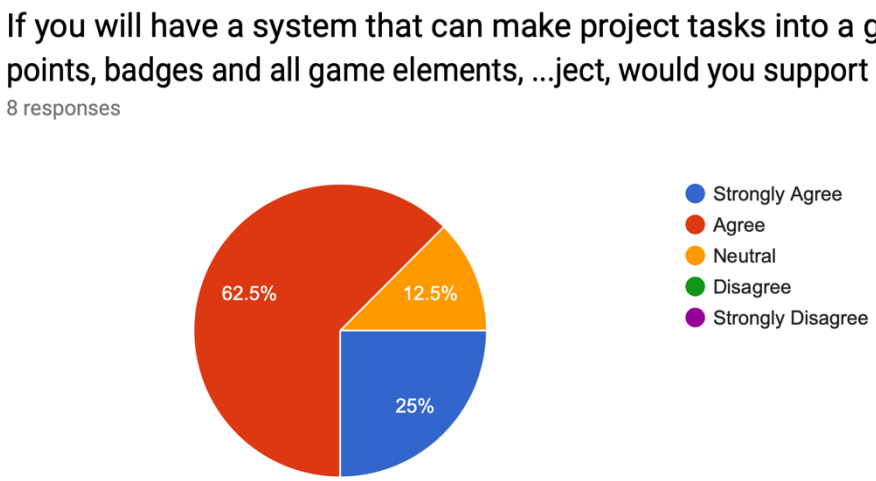

Figure 5.9: The Acceptance of Gamification among Developers in the Case Study

Figure (5.9) shows the accepting level of gamification among the developers. 62.5% of developers agreed, 25% strongly agreed that having game elements is an idea that could create a positive impact if it is applied within their projects.

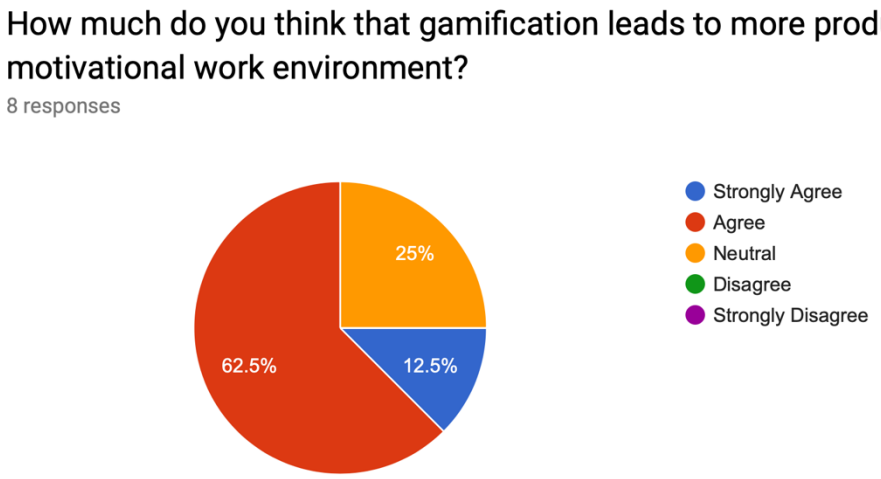

Figure 5.10: The Effect of Gamification in Motivating the Developer in the Case Study

Figure (5.10) shows that developers described that having gamification on their projects can motivate them to raise their productivity and create a more friendly and social networking environment with the other team members.

5.1.2 Usability Testing Results

In order to evaluate the usability of the proposed system, we have made a usability study using the Esurvey tool and the user experience questionnaire AttrakDiff was chosen for testing [37], the study was done by the eight participants after the test session.

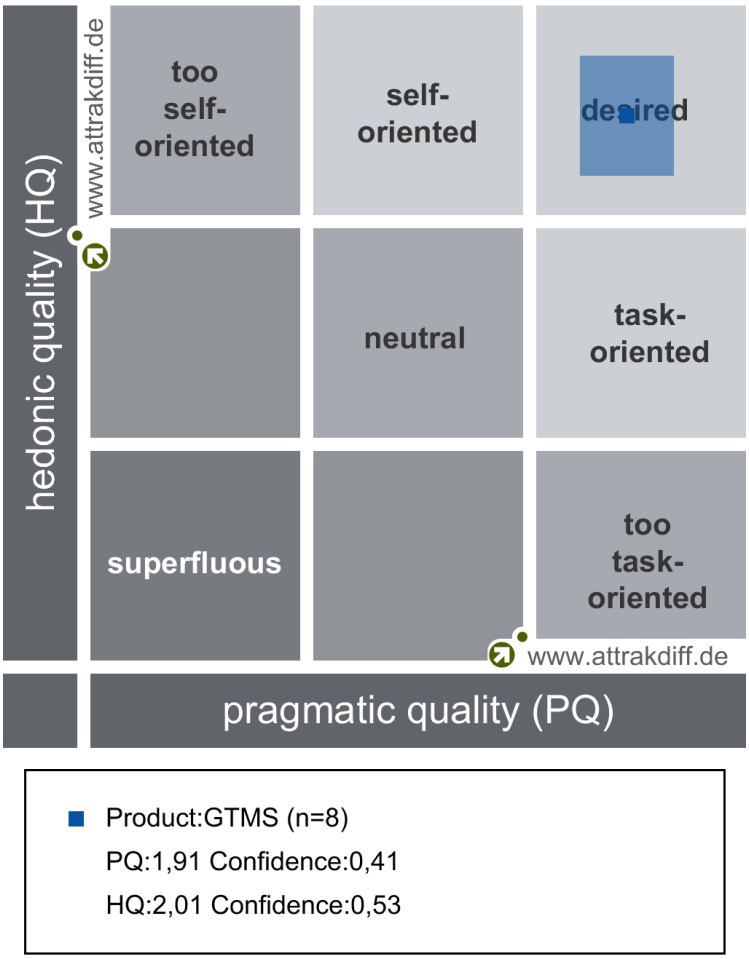

Figure 5.11: Portfolio-Presentation

Figure (5.11) shows that the average rating for pragmatic and hedonic quality of the system correspond to a position of the field 'desired'. The chart shows that the system has a slightly higher rating for hedonic quality. The system seems to provide a very good compromise of user satisfaction. The center points are surrounded by a small confidence rectangle, which indicates that subjects strongly agree in their evaluations.

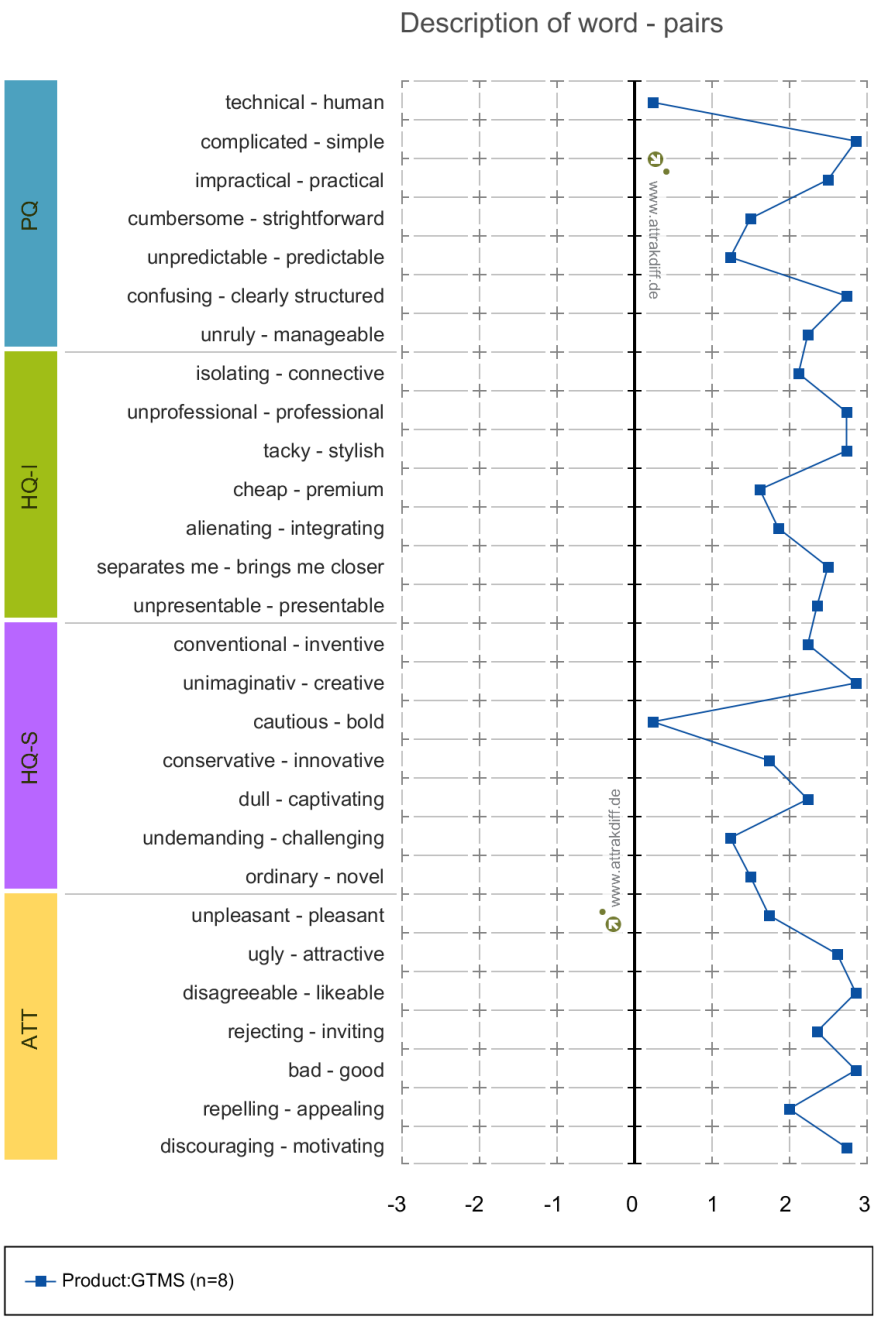

Figure 5.12: Description of Word-Pairs

Figure (5.12) illustrates an overview of the user ratings. The system was rated as very motivating and this is most likely related to game elements. The system had good ratings for properties such as simple, clearly structured, and likable. These are all favorable aspects of good user experience. They were also characterized as “good” and rated with very good for this property. However, the system needs improvement, especially regarding professionalism. A system that is more human (natural) than technical better connects its users to the world rather than isolating them, this comes as a result of the limitation of this

test, that real rewards were not tested as we are doing a case study this would provide a more positive user experience; refer to appendix J for more details.

5.3 Conclusion and Discussion

The Six-Step approach provided by Werbach and Hunter [6] was sufficient to obtain the best design of the gamification system and to merge it with the task management systems. The approach that Werbach and Hunter offered was comprehensive of all aspects of the game design, as well as of all aspects of business. In this research, reaching the best possible design of gamification that increases the performance of team members was the main objective, studying it and observe its impact on programmers in the Agile work environment in Information Technology companies. After following this approach, the Matrix of Achievement was created, A case study was made and the Metrics of achievements was assessed. All these steps helped in understanding the players and evaluating their performance.

Positive results on the Matrix of Achievement and on the Usability Testing were noted. The developers found that the gamified system leads to motivation, competition, and social interaction, all of which lead to a more engaged and motivated team. These advantages were observed on the Matrix of Achievements results. Six parameters of the Metrics of Achievements were tested and analyzed. The gamified phase showed the increase of the developer's performance in the gamified phase than the one without gamification.

The Usability Testing study shows that the developers see the system as desired. The gamified system also functions to ensure that the work of every developer was observed correctly, and in a transparent unbiased fashion. In conclusion, the gamified system has proven itself to be a successful approach in increasing the performance and productivity of team members in software projects.

Chapter 6 Findings and Results

6.1 Conclusion

Designing and applying a gamification system for software management is not easy nor straightforward. In this research, the topic of gamification was introduced in the context of software development projects. The study proposed an approach to embed gamification elements into software projects, and reflected on the added value to the management and development teams. The game elements design was implemented by following the Six-Step Approach and adapted to fit the purposes and needs of software projects.

In the proposed system, gamification was integrated into software projects by adding game elements to tasks and using the outcome to support the management teams in the evaluation and monitoring process.

The case study of this research highlighted issues in the traditional project management motivation, performance, and evaluation in development cycle. One issue that this research helped overcome is the inadequate reports of developers' performance and overlooking of their efforts, due to limited resources. Which has a direct impact on the developer's performance.

The results of the case study show the improvement of the developer's performance. The six parameters were tested and used in the case study. The results show the positive improvement of the developer's performance and productivity. Positive results on the proposed Matrix of Achievement and on the Usability Test were noted. In conclusion, the gamified system has been proven to be a successful approach in increasing the performance of team members in software projects. Gamification is an effective tool for supporting team management, especially in teams with diverse skills and personalities. The methodology used in the research could also be used to gamify other types of projects or applied in different environments.

6.2 Limitations

Adding a challenges section to give users more points will allow more interaction between team members. However, it is not easy to create challenge elements because it may lead to a negative effect on the developer, this area is critical to the application and needs more investigation. It is important to keep in mind that for example not every employee likes to compete against others.

Creating a generalized design is not easy; game design needs to be integrated properly with the project and each project has different qualities and different conditions.

Employees might fear disclosure of their data to the company, insurances, or authorities due to the reason that the gamified software uses data and activity logs, which could lead to privacy issues for them.

6.3 Future Work

The next step for this research would be aimed at completing further enhancement on gamification strategy and its connection with task data by improving an effective analysis of the results of the tasks. In addition, more validation of gamification elements in more extensive task environments would also be explored specially that the study shows slightly improvement in the Senior level developers which could open more opportunity to deeply investigate more factors. To achieve that, a bigger segment of managers and developers for testing to improve our approach to reflect their real-life challenges and improvement suggestions is needed. Further research is also needed to evaluate gamification design from both perspectives of developers and managers.

Finally, enhancing the algorithm used to calculate user points that depend on the proposed "Matrix of Achievement". We believe this requires more testing and experience to reach the best measurements. It is not easy to enhance products or projects especially when it still a minimum valuable product. One needs to repeat the test many times and study many users to make sure of the parts that need enhancement and the parts that do not. At some point one will end up needing to rewrite the whole application just to achieve user satisfaction.

References

[1] S. Deterding, D. Dixon, R. Khaled and L. Nacke, "From Game Design Elements to Gamefulness: Defining “Gamification”„" in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 2011.

[2] P. M. I. (PMI), Project Management Body of Knowledge Book.

[3] C. Marache-Francisco and E. Brangier, "Process of gamification. from the consideration of gamification to its practical implementation," 2013.

[4] T. DeMarco and T. Lister, Peopleware: Productive Projects and Teams, 3rd Edition, 2016.

[5] F. García, O. Pedreira, M. Piattini, A. Cerdeira-Pena and M. Penabad, "A framework for gamification in software engineering," *The Journal of Systems and Software,* pp. 21-40, 2017.

[6] K. a. H. D. Werbach, For the Win: How Game Thinking Can Revolutionize Your Business, Wharton Digital Press, 2012.

[7] R. Vesselinov and J. Grego, "Duolingo Effectiveness Study," 2012.

[8] P. Kukhnavets, "Why Agile is So Popular in Project Management World," 11 October 2018. [Online]. Available: https://hygger.io/blog/why-agile-is-so-popular-in-project-management/.

[9] "EFL Gamification 1: Intrinsic and Extrinsic Rewards," Mozuku.edublogs.org, 2013. [Online]. Available: https://mozuku.edublogs.org/2013/02/08/efl-gamification-1/. [Accessed 3 January 2016].

[10] "The history of gamification: from the very beginning to right now," Growth Engineering, 29 August 2019. [Online]. Available: https://www.growthengineering.co.uk/history-of-gamification/.

[11] P. M. Institute, Project Management Body of Knowledge, 1996.

[12] S. Haunts, Agile Software Development Succinctly, Syncfusion, Inc. .

[13] T. Dingsøyr, N. Brede Moe, T. Erlend Fægri and E. Amdahl Seim, "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation," *Empirical Software Engineering,* pp. 490-520, 2018.

[14] D. E.Strode, S. L.Huff, B. Hope and S. Link, "Coordination in co-located agile software development projects," *Journal of Systems and Software,* vol. 85, no. 6, pp. 1222-1238, 2012.

[15] J. M. a. H. M. Kumar, Gamification at Work: Designing Engaging Business Software, Aarhus: The Interaction Design Foundation, 2013.

[16] J. Hamari, J. Koivisto and H. Sarsa, "Does Gamification Work? – A Literature Review of Empirical Studies on Gamification," Hawaii, USA, 2014.

[17] G. Zichermann and C. Cunningham, Gamification by design: Implementing game mechanics in web and mobile apps, O'Reilly Media, Inc., 2011.

[18] P. Weiser, D. Bucher, F. Cellina and V. De Luca, "A Taxonomy of Motivational Affordances for Meaningful Gamified and Persuasive Technologies," in *ICT4S*, Copenhagen, 2015.

[19] P. Denny, "The effect of virtual achievements on student engagement," in *SIGCHI Conference on Human Factors in Computing Systems*, 2013.

[20] D. J. Dubois and G. Tamburrelli, "Understanding Gamification mechanisms for software development.," in *9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*, 2013.

[21] R. Al-Azawi, F. Al-Faliti and M. Al-Blushi, "Educational Gamification Vs. Game Based Learning: Comparative Study," *International Journal of Innovation, Management and Technology,* vol. 7, no. 4, 2016.

[22] N. Unkelos-Shpigel and I. Hadar, "Gamifying Software Development Environments Using Cognitive principles".

[23] R. Marques, C. Costa, M. Mira da Silva and P. Gonçalves, "Gamifying Software Development Scrum Projects," in *IEEE Conference on Virtual Worlds and Games for Serious Applications (VS-Games)*, 2017.

[24] M. Češka, "Gamification in the SCRUM Software Development Framework," 2015. [Online]. Available: https://is.muni.cz/th/sakwq/430640_MasterThesis_FINAL.pdf.

[25] V. Platonova and S. Berzisa, "Gamification in Software Development Projects," in *Information Technology and Management Science*, 2017.

[26] M. Trotzek, C. Reimann and E. Vitkauskaite, "Gamification Fundamentals and State of Research and Development. Proceedings of International Research Conference Dortmund, Dortmund," 2015. [Online].

[27] E. B. Passos, D. B. Medeiros, P. A. Neto and E. W. Clua, "Turning Real-World Software Development into a Game.," in *IEEE*, Salvador, 2011.

[28] B. Cugelman, "Gamification: What It Is and Why It Matters to Digital Health Behavior Change Developers," in *JMIR* , 2013.

[29] R. Hunicke, M. LeBlanc and R. & Zubek, "MDA: A Formal Approach to Game Design and Game Research.," in *Game Developers Conference AAAI Workshop*, San Jose, 2004.

[30] A. Mora, D. Riera, C. Gonzalez and J. Arnedo-Moreno, "A literature review of gamification design frameworks," in *IEEE - Games and Virtual Worlds for Serious Applications, VS-GAMES*, 2015.

[31] K. Werbach and D. Hunter, The Gamification Toolkit: Dynamics, Mechanics, and Components for the Win, Wharton School Press, 2015.

[32] B. Burke, "Why Gamification is Not a Game," *CIO Journal,* 2014.

[33] S. Deterding, K. O'Hara, M. Sicart, D. Dixon and L. Nacke, "Gamification: Using Game Design Elements in Non-Gaming Contexts," in *the International Conference on Human Factors in Computing Systems, CHI 2011*, Vancouver, 2011.

[34] A. Aldaghamin, N. Sleibi and C. Reimann, "Gamification of Agile Project Management for Software Development Projects, 1st International Conference on Research and Education in Project Management," in *REPM*, Bilbao, 2018.

[35] "Asana simplifies team-based work management," Asana, Inc., [Online]. Available: https://asana.com/de.

[36] J. Nielsen and T. Landauer, "A Mathematical Model of the Finding of Usability Problems," in *Conference on Human Factors in Computing Systems*, 1993.

[37] "AttrakDiff," [Online]. Available: http://www.attrakdiff.de.

ID	level	position	x1 Traditi	x1 with gamification
1	Senior 1	Frontend Developer	9	10
2	Senior 2	Frontend Developer	6	8
3	Mid-level 7	Frontend Developer	5	6
4	Mid-level 1	Frontend Developer	7	8
5	Junior 7	Frontend Developer	3	4
6	Junior 6	Frontend Developer	4	5
7	Senior 3	Backend Developer	8	9
8	Mid-level 6	Backend Developer	5	7
9	Junior 4	Backend Developer	4	6
10	Junior 3	Backend Developer	5	6
11	Senior 4	Software Engineer	8	9
12	Senior 5	Software Engineer	7	8
13	Mid-level 2	Software Engineer	6	7
14	Mid-level 4	Software Engineer	6	7
15	Junior 2	Software Engineer	5	6
16	Junior 5	Software Engineer	4	5
17	Junior 1	Software Engineer	6	7
18	Senior 6	Software Engineer	6	7
19	Mid-level 3	Software Engineer	6	7
20	Mid-level 5	Software Engineer	7	8

Appendix A x1 Traditional and Gamified Phases Data

Id	Level	Position	Id	120 minu	60 minutes	80 minutes	45 minutes	30 minutes	30 minutes	40 minutes	50 minutes	60 minutes	20 minutes	40 minutes
1	Senior 1	Frontend Developer	1	57,23	44,29	54,11	31,74	13,26	17,45	35,15	39,88	52,43	9,10	33,61
2	Senior 2	Frontend Developer	2	59,43	39,96	59,08	32,14	20,27	21,12	36,80	41,89	47,27	15,19	37,06
3	Mid-level 7	Frontend Developer	3	100,45	48,15	74,45	40,65	32,61	29,52	40,94	53,78	73,92	18,81	42,20
4	Mid-level 1	Frontend Developer	4	81,04	40,58	68,58	36,81	24,52	27,17	38,12	45,76	57,26	16,75	40,05
5	Junior 7	Frontend Developer	5	146,84	49,87	87,01	51,66	38,32	41,44	48,12	66,59	74,39	23,84	44,02
6	Junior 6	Frontend Developer	6	124,76	56,89	78,96	44,96	35,85	32,44	42,63	61,04	65,74	28,57	43,42
7	Senior 3	Backend Developer	7	72,35	45,40	64,22	35,01	18,21	25,31	35,31	47,00	53,48	15,70	37,06
8	Mid-level 6	Backend Developer	8	97,10	48,56	77,02	39,48	31,03	30,27	41,10	47,45	58,84	17,45	40,76
9	Junior 4	Backend Developer	9	115,70	52,83	84,43	47,60	34,40	32,41	41,96	60,08	62,26	20,40	43,06
10	Junior 3	Backend Developer	10	111,40	49,96	78,53	45,46	32,79	32,06	42,39	57,46	63,03	22,24	42,80
11	Senior 4	Software Engineer	11	77,42	38,57	65,63	36,30	23,50	24,44	36,96	40,87	48,67	10,72	38,24
12	Senior 5	Software Engineer	12	78,19	44,94	63,72	36,41	27,34	23,10	38,60	41,90	54,67	16,37	38,46
13	Mid-level 2	Software Engineer	13	81,15	46,62	68,77	38,72	27,77	27,00	38,21	43,39	55,89	16,60	40,48
14	Mid-level 4	Software Engineer	14	92,30	46,78	72,09	41,82	29,33	27,79	39,71	48,22	55,90	19,29	39,71
15	Junior 2	Software Engineer	15	110,31	59,36	78,42	42,33	33,46	31,91	40,30	49,06	60,15	19,86	41,43
16	Junior 5	Software Engineer	16	123,50	50,04	79,82	44,53	36,50	34,13	41,85	57,54	61,42	20,35	43,00
17	Junior 1	Software Engineer	17	103,85	49,05	77,00	39,91	34,15	32,04	40,96	54,88	61,31	18,59	40,53
18	Senior 6	Software Engineer	18	80,89	45,64	68,56	37,77	26,39	25,47	38,70	41,21	50,26	14,36	37,83
19	Mid-level 3	Software Engineer	19	90,27	46,73	69,78	38,87	29,52	26,65	39,08	47,44	56,24	16,52	38,97
20	Mid-level 5	Software Engineer	20	96,84	47,19	73,67	39,70	28,64	31,47	40,19	47,80	58,61	18,24	41,25
				95,05	47,45	71,41	39,48	28,40	27,99	39,42	48,77	57,75	17,64	40,00

Appendix B x2 Tasks Duration Before Gamification (Traditional)

Id	Level	Position	Id	120 minu	60 minutes	80 minutes	45 minutes	30 minutes	30 minutes	40 minutes	50 minutes	60 minutes	20 minutes	40 minutes
1	Senior 1	Frontend Developer	1	52,03	36,91	68,72	33,64	15,91	10,56	29,29	24,92	43,69	8,27	28,96
2	Senior 2	Frontend Developer	2	54,02	30,74	75,04	34,07	24,72	11,86	30,67	26,18	39,39	13,81	30,44
3	Mid-level 7	Frontend Developer	3	65,77	37,84	79,63	37,11	21,85	13,74	27,81	31,34	44,57	14,27	32,22
4	Mid-level 1	Frontend Developer	4	70,38	32,41	83,34	38,48	28,20	13,68	30,30	25,54	40,56	9,74	32,68
5	Junior 7	Frontend Developer	5	61,56	36,54	79,01	38,60	32,80	12,23	31,13	26,19	46,56	14,88	33,40
6	Junior 6	Frontend Developer	6	67,41	38,04	87,08	40,03	31,67	16,38	31,21	25,76	41,88	11,67	33,50
7	Senior 3	Backend Developer	7	73,68	33,82	86,41	40,49	29,42	14,84	30,74	26,92	47,72	13,62	34,05
8	Mid-level 6	Backend Developer	8	73,77	38,85	86,65	42,59	33,32	15,71	30,81	25,52	46,57	13,50	34,11
9	Junior 4	Backend Developer	9	68,38	38,94	87,92	42,75	35,42	18,77	31,52	27,90	48,87	13,43	34,42
10	Junior 3	Backend Developer	10	83,91	37,42	90,83	46,00	36,67	19,84	32,02	43,84	46,58	15,68	34,60
11	Senior 4	Software Engineer	11	88,04	39,33	92,82	43,67	35,80	20,73	32,41	39,84	48,84	14,83	35,05
12	Senior 5	Software Engineer	12	88,27	40,47	100,12	43,43	38,78	19,56	33,69	27,91	49,03	14,18	35,61
13	Mid-level 2	Software Engineer	13	91,32	39,14	96,79	44,71	40,77	18,79	33,02	48,89	61,60	15,29	35,77
14	Mid-level 4	Software Engineer	14	91,91	39,88	90,86	43,90	42,76	19,19	33,03	49,89	52,09	15,11	36,35
15	Junior 2	Software Engineer	15	84,85	46,74	92,53	47,83	41,49	20,89	33,03	44,60	50,12	16,14	36,51
16	Junior 5	Software Engineer	16	85,69	37,56	92,67	51,82	40,66	23,87	34,75	52,24	52,52	20,22	37,87
17	Junior 1	Software Engineer	17	90,39	42,95	99,63	56,16	41,28	24,44	34,12	54,61	51,88	18,55	38,93
18	Senior 6	Software Engineer	18	95,00	39,10	98,97	50,32	43,80	25,33	34,03	52,31	53,18	18,50	40,44
19	Mid-level 3	Software Engineer	19	95,97	47,41	97,91	50,81	43,03	26,54	33,30	55,49	54,78	25,97	40,94
20	Mid-level 5	Software Engineer	20	133,49	40,87	107,90	58,38	45,98	27,07	40,10	60,54	61,99	21,68	41,70
				80,79	38,64	88,79	43,50	34,65	18,26	31,94	37,36	48,44	15,14	35,05

Appendix C x2 Tasks Duration After Gamification

id	level	position	Task 1 Traditior	Task 1 Gamified
1	Senior 1	Frontend Dev	57,23	52,03
2	Senior 2	Frontend Dev	59,43	54,02
3	Mid-level 7	Frontend Dev	100,45	91,32
4	Mid-level 1	Frontend Dev	81,04	73,68
5	Junior 7	Frontend Dev	146,84	133,49
6	Junior 6	Frontend Dev	124,76	95,97
7	Senior 3	Backend Dev	72,35	65,77
8	Mid-level 6	Backend Dev	97,10	88,27
9	Junior 4	Backend Dev	115,70	90,39
10	Junior 3	Backend Dev	111,40	85,69
11	Senior 4	Software Eng	77,42	70,38
12	Senior 5	Software Eng	78,19	61,56
13	Mid-level 2	Software Eng	81,15	73,77
14	Mid-level 4	Software Eng	92,30	83,91
15	Junior 2	Software Eng	110,31	84,85
16	Junior 5	Software Eng	123,50	95,00
17	Junior 1	Software Eng	103,85	91,91
18	Senior 6	Software Eng	80,89	67,41
19	Mid-level 3	Software Eng	90,27	68,38
20	Mid-level 5	Software Eng	96,84	88,04

Appendix D x2 Traditional and Gamified Phases Data for Task 1

ID	level	position	x3 Traditional	x3 with gamification
1	Senior 1	Frontend Developer	5	5
2	Senior 2	Frontend Developer	4	5
3	Mid-level 7	Frontend Developer	2	3
4	Mid-level 1	Frontend Developer	3	4
5	Junior 7	Frontend Developer	1	1
6	Junior 6	Frontend Developer	1	2
7	Senior 3	Backend Developer	4	4
8	Mid-level 6	Backend Developer	2	3
9	Junior 4	Backend Developer	2	3
10	Junior 3	Backend Developer	2	3
11	Senior 4	Software Engineer	3	4
12	Senior 5	Software Engineer	3	4
13	Mid-level 2	Software Engineer	3	4
14	Mid-level 4	Software Engineer	3	4
15	Junior 2	Software Engineer	1	2
16	Junior 5	Software Engineer	3	4
17	Junior 1	Software Engineer	2	3
18	Senior 6	Software Engineer	4	4
19	Mid-level 3	Software Engineer	3	4
20	Mid-level 5	Software Engineer	3	3

Appendix E x3 Traditional and Gamified Phases Data

ID	level	position	x4 Traditional	x4 with gamification
1	Senior 1	Frontend Developer	3	4
2	Senior 2	Frontend Developer	2	3
3	Mid-level 7	Frontend Developer	4	4
4	Mid-level 1	Frontend Developer	4	5
5	Junior 7	Frontend Developer	2	3
6	Junior 6	Frontend Developer	1	3
7	Senior 3	Backend Developer	4	5
8	Mid-level 6	Backend Developer	3	4
9	Junior 4	Backend Developer	3	4
10	Junior 3	Backend Developer	5	6
11	Senior 4	Software Engineer	3	3
12	Senior 5	Software Engineer	2	3
13	Mid-level 2	Software Engineer	5	6
14	Mid-level 4	Software Engineer	3	3
15	Junior 2	Software Engineer	3	4
16	Junior 5	Software Engineer	2	3
17	Junior 1	Software Engineer	4	5
18	Senior 6	Software Engineer	4	4
19	Mid-level 3	Software Engineer	1	2
20	Mid-level 5	Software Engineer	5	5

Appendix F x4 Traditional and Gamified Phases Data

ID	level	position	x5 Traditional	x5 with gamification
1	Senior 1	Frontend Developer	2	1
2	Senior 2	Frontend Developer	2	1
3	Mid-level 7	Frontend Developer	5	4
4	Mid-level 1	Frontend Developer	4	3
5	Junior 7	Frontend Developer	7	5
6	Junior 6	Frontend Developer	6	4
7	Senior 3	Backend Developer	3	2
8	Mid-level 6	Backend Developer	5	4
9	Junior 4	Backend Developer	5	4
10	Junior 3	Backend Developer	5	4
11	Senior 4	Software Engineer	3	2
12	Senior 5	Software Engineer	3	3
13	Mid-level 2	Software Engineer	4	3
14	Mid-level 4	Software Engineer	4	3
15	Junior 2	Software Engineer	5	4
16	Junior 5	Software Engineer	6	5
17	Junior 1	Software Engineer	4	3
18	Senior 6	Software Engineer	3	2
19	Mid-level 3	Software Engineer	4	3
20	Mid-level 5	Software Engineer	4	3

Appendix G x5 Traditional and Gamified Phases Data

ID	level	position	x6 Traditional	x6 with gamification
1	Senior 1	Frontend Developer	9	10
2	Senior 2	Frontend Developer	8	9
3	Mid-level 7	Frontend Developer	6	7
4	Mid-level 1	Frontend Developer	7	8
5	Junior 7	Frontend Developer	5	6
6	Junior 6	Frontend Developer	5	6
7	Senior 3	Backend Developer	8	9
8	Mid-level 6	Backend Developer	6	8
9	Junior 4	Backend Developer	6	7
10	Junior 3	Backend Developer	6	7
11	Senior 4	Software Engineer	7	9
12	Senior 5	Software Engineer	7	8
13	Mid-level 2	Software Engineer	7	8
14	Mid-level 4	Software Engineer	7	8
15	Junior 2	Software Engineer	6	7
16	Junior 5	Software Engineer	5	7
17	Junior 1	Software Engineer	6	7
18	Senior 6	Software Engineer	8	9
19	Mid-level 3	Software Engineer	7	8
20	Mid-level 5	Software Engineer	7	8

Appendix H x6 Traditional and Gamified Phases Data

Developer	Level	Position	Points	Project Badges
1	Senior 1	Frontend Developer	600	KUDOS; Best Developer; Fastest Developer; Best Quality Developer; Overachiever
2	Senior 2	Frontend Developer	505	Fast Developer; Best Quality Developer; Overachiever
3	Mid-level 7	Frontend Developer	362	Shiny Developer
4	Mid-level 1	Frontend Developer	442	Team Worker Developer
5	Junior 7	Frontend Developer	272	Shiny Developer
6	Junior 6	Frontend Developer	302	Shiny Developer
7	Senior 3	Backend Developer	488	Team Worker Developer; Overachiever
8	Mid-level 6	Backend Developer	391	Shiny Developer
9	Junior 4	Backend Developer	359	Shiny Developer
10	Junior 3	Backend Developer	399	Team Worker Developer
11	Senior 4	Software Engineer	449	Overachiever
12	Senior 5	Software Engineer	406	Shiny Developer
13	Mid-level 2	Software Engineer	457	Team Worker Developer
14	Mid-level 4	Software Engineer	392	Shiny Developer
15	Junior 2	Software Engineer	354	Shiny Developer
16	Junior 5	Software Engineer	319	Shiny Developer
17	Junior 1	Software Engineer	395	Team Worker Developer
18	Senior 6	Software Engineer	456	Overachiever
19	Mid-level 3	Software Engineer	377	Shiny Developer
20	Mid-level 5	Software Engineer	424	Team Worker Developer

Appendix I Gamified System Output

Please click one item in every line.

confusing*	○	○	○	○	○	○	○	clearly structured
repelling*	○	○	○	○	○	○	○	appealing
bold*	○	○	○	○	○	○	○	cautious
innovative*	○	○	○	○	○	○	○	conservative
dull*	○	○	○	○	○	○	○	captivating
undemanding*	○	○	○	○	○	○	○	challenging
motivating*	○	○	○	○	○	○	○	discouraging
novel*	○	○	○	○	○	○	○	ordinary
unruly*	○	○	○	○	○	○	○	manageable

*required field

Please click one item in every line.

human*	○	○	○	○	○	○	○	technical
isolating*	○	○	○	○	○	○	○	connective
pleasant*	○	○	○	○	○	○	○	unpleasant
inventive*	○	○	○	○	○	○	○	conventional
simple*	○	○	○	○	○	○	○	complicated
professional*	○	○	○	○	○	○	○	unprofessional
ugly*	○	○	○	○	○	○	○	attractive
practical*	○	○	○	○	○	○	○	impractical
likeable*	○	○	○	○	○	○	○	disagreeable
cumbersome*	○	○	○	○	○	○	○	straightforward

*required field

Please click one item in every line.

stylish*	○	○	○	○	○	○	○	tacky
predictable*	○	○	○	○	○	○	○	unpredictable
cheap*	○	○	○	○	○	○	○	premium
alienating*	○	○	○	○	○	○	○	integrating
brings me closer to people*	○	○	○	○	○	○	○	separates me from people
unpresentable*	○	○	○	○	○	○	○	presentable
rejecting*	○	○	○	○	○	○	○	inviting
unimaginative*	○	○	○	○	○	○	○	creative
good*	○	○	○	○	○	○	○	bad

*required field

Appendix J Usability Test

الملخص

تصميم الالعاب لإدارة المشاريع المرنة للمشاريع البرمجية ومهام تطوير البرمجيات

اعداد: أريج إبراهيم سلمان الدغامين ,21510410

إشراف: د. رشيد جيوسي

تركز هذه الرسالة على تحسين أداء أعضاء الفريق العاملين في مشاريع البرمجيات المرنة. المشاريع المرنة معقدة وتحتوي على تغييرات سريعة وتتطلب وثائق أقل. تعتمد هذه المشاريع على أداء الفريق وتقييمه. تصميم الالعاب هو أداة تستخدم لتحفيز المستخدمين على التصرف حسب الرغبة، والتي يمكن دمجها بشكل مثالي مع تطوير البرامج.

نهج هذا البحث هو تحفيز أنظمة إدارة المشاريع لتطوير البرمجيات المرنة لضمان الأداء العالي لأعضاء الفريق مع التعرف على اختلافات الأفراد، وتشجيع كل عضو في الفريق على العمل نحو هدف مشترك. تم إنشاء نموذج أولي مقترح من خلال دمج عناصر الالعاب مع نظام إدارة المهام في مشاريع البرامج، خاصة لمهام التطوير لدراسة تأثيرها على أداء أعضاء الفريق. كما تمت دراسة الآثار المترتبة على تضمين عناصر الالعاب في بيئة العمل. تم فحص تطبيقه كقيمة مضافة لإدارة وتطوير فرق مشاريع البرمجيات في شركات تكنولوجيا المعلومات. وقد تم تحقيق ذلك من خلال نهج من ست خطوات للتشجيع على مهام تطوير البرمجيات التي تشمل تقديم مزاياها وتطبيقها والتفكير فيها من منظور الإدارة والموظفين.

يقدم هذا البحث نتائج حالة الاختبار. تم إجراء الاختبارات مع مجموعة من عشرين مطورًا، وتمت مقابلتهم بعد إكمال اختبار قابلية الاستخدام في سياق التلعيب. النتائج المقدمة لا تدعم فقط كيف يظهر تصميم الالعاب زيادة في أداء المطورين وإنتاجيتهم ولكن أيضًا يعرض فوائده على تقييم فريق الإدارة.