AN INTERACTIVE LEARNING APPROACH TO TEACH ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESSES

UFUK AYDAN

SEPTEMBER 2016

AN INTERACTIVE LEARNING APPROACH TO TEACH ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESSES

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

BY
UFUK AYDAN

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

SEPTEMBER 2016

Title of the Thesis: **An Interactive Learning Approach to Teach ISO/IEC 12207 Software Life Cycle Processes**

Submitted by **Ufuk AYDAN**

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

Prof. Dr. Halil Tanyer EYYUBOĞLU
Director

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

Prof. Dr. Müslim BOZYİĞİT
Head of Department

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

Asst. Prof. Dr. Murat YILMAZ
Supervisor

Examination Date: 02.09.2016

Examining Committee Members

Asst. Prof. Dr. Murat YILMAZ	(Çankaya Univ.)
Asst. Prof. Dr. Murat SARAN	(Çankaya Univ.)
Asst. Prof. Dr. Meltem ERYILMAZ	(Atılım Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

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

Name, Last Name : Ufuk AYDAN
Signature : 
Date : 02.09.2016

ABSTRACT

AN INTERACTIVE LEARNING APPROACH

TO TEACH ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESSES;

AYDAN, Ufuk

M.Sc., Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Murat YILMAZ

Serious games have been powerful and interactive tools that provide information about specific subjects. They can be considered as instructive and found beneficial to gain experience about particular technical skills and ultimately developed to teach engineering subjects. Although there are several studies, which are related to the application of serious games for software engineering field especially in software development, there is a no serious game that teaches fundamentals of ISO/IEC 12207 Software Life Cycle Processes.

“Floors” is a serious game that proposes interactive learning experience. The goal of the game is to introduce the fundamentals of ISO/IEC 12207 by creating a virtual experience them in an iterative flow of processes with interactive technical dialogues in a 3D computer simulated office environment. In addition, it can be used as a complementary tool for software engineering students to provide training for ISO/IEC 12207 processes. By playing the game such concepts will create awareness on participants about monitoring and maintaining the software development life cycle. This research provides a framework for the exploration of the research the data obtained from computer engineering students. Results suggest that there is a significant difference between students who played Floor and paper-based learning session. Taken together, these findings clearly demonstrate that participants who played Floors tend to have more positive experience and knowledge about the standard.

Keywords: ISO/IEC 12207, Software Development Life Cycle, Software Engineering, serious game, interactive learning process, gamification.

ÖZ

ISO/IEC 12207 YAZILIM YAŞAM DÖNGÜSÜ SÜREÇLERİ

ETKİLEŞİMLİ ÖĞRENME YAKLAŞIMI

AYDAN, Ufuk

Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Murat YILMAZ

Yazılım mühendisliği alanında özellikle yazılım projesi geliştirme kapsamında literatür dahilinde birçok iş ve çalışma olmasına karşın ISO/IEC 12207 Yazılım Yaşam Döngüsü Süreçleri temellerini ve etkileşimli öğrenme sürecini kapsayan uygun bir ciddi oyun örneği eksikliği bulunmaktadır. Ciddi oyunlar belirli konularda eğitici ve belirli beceri alanlarında yetkinlik kazanılmasını sağlayan uygulamalar olup oyun ve oyunlaştırma literatüründe önemli bir yere sahiptir.

"Floors", bir ciddi oyun olmakta ve etkili bir şekilde öğrenme sürecini etkileşimli ve ilerlemeli bir tasarım ile sunmaktadır. Oyunun sahip olduğu ve kullanıcılara edindirdiği esas avantaj ISO/IEC 12207 yazılım yaşam döngüsü süreçlerinin temel presipleri hakkında aşinalık kazandırmak, aynı zamanda ilerlemeli yapısı sayesinde belirli süreçlerin akışını etkileşimli diyaloglarla 3 boyutlu sanal bir ortamda birleştirmesidir. Tüm bunları amaç kapsamı için adanmış görsel bir ofis ortamında gerçekleştirmesidir *Floors* özellikle öğrencilerin ISO/IEC 12207'nin belirli süreçlerinin kapsamında temel bir eğitim sunan, yardımcı bir araç olarak kullanılabilir. Oyunun katılımcılar tarafında oynanması itibariyle belirtilen kavramların ve konseptin hakkında ve ayrıca yazılım geliştirme yaşam döngüsünün izlenmesi ve devam ettirilmesi ile ilgili kullanıcılarda belirli bir farkındalığın sağlanması temel amaçlardandır. Araştırma sonuçları katılımcı popülasyonundan elde edilen veriler ile yapılan anket çalışması ile belirli derecede farkın ortaya çıktığı belirlenmiştir. Çalışmanın sonuçları dikkate alındığında *Floors* ile tecrübe edinen katılımcıların standart hakkında daha olumlu bir bakış açısı edindiği ve temel kavramları kazandığı gözlemlenmiştir.

Anahtar Kelimeler: ISO/IEC 12207 Yazılım Yaşam Döngüsü Süreçleri, ciddi oyun, Yazılım Mühendisliği, etkileşimli öğrenme süreci, oyunlaştırma.

Acknowledgements

I would like to thank Asst. Prof. Dr. Murat YILMAZ for his glorious supervision and endless support throughout the whole study. With the help of his essential touches in every phase and provisions of extensive information the backbone of the study were created.

And undoubtedly my family. They are the great and exact examples of the definition of devotion and dedication. I thank individually to my father Uğur AYDAN, my mother Hülya AYDAN, my grandmother Meserret AYDAN, and my brother Can AYDAN for their boundless support while creating a comfortable and safe environment during my education and every stage of my life.

I would like to thank my dear friends Mert YILMAZ and Gökhan KARA for their loyal friendships and their brilliant ideas.

Finally, I wish to thank jury committee for their patience and understandings.

Ufuk AYDAN

Table of Contents

List of Tables

List of Figures

List of Abbreviations

PC	Personal Computer
UI	User Interface
3D	Three Dimensional
VR	Virtual Reality
FBX	Filmbox Format
IBM	International Business Machines
SPSS	Statistical Package for Social Sciences
NPC	Non-playable character
SE	Software Engineering
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineer
SRS	Software Requirements Specification

Chapter 1

1. INTRODUCTION

Undoubtedly, there is an essential need for software systems to continue to satisfy today's complex and costly business demands. Sommerville [95] claims, "*We cannot run the modern world without software" (Sommerville 2009, p. 4)*. The important point is in today's world there is no chance to maintain all of the work around us without software. The software is a set of computer programs and artifacts with related documentation for a particular customer or a defined goal.

The necessity and the essentiality make the software crucial for our lives. In many fields such as industry, logistics, finance, and entertainment (e.g. computer games, music market, cinema, etc.) require the control of computer-based systems. Without using the benefits of software systems, it is hard to build substantive international relations and trade [95]. Gibbs states that ongoing projects and the general applications from industry do not consistently employ without clear definitions and implementations of market demands via software engineering structures [47].

The developers and software engineers should be aware of the complexity of the software development and its expected responsibilities, before the start of both education and the projects. However this procedure hides various challenges in general and needs to be examined carefully in order to satisfy the desires. According to Sommerville [95] software systems are abstract and intangible. The constraints are not visible. Governmental laws and physical materials are not the main limitations of the development processes. Therefore, all of them simplify the processes of software engineering and software development. There is absence of physical borders to it. However, with this absence the complexity of software systems and their development procedure can reach extreme points. Because of the hard learning or understanding period, costs or expenses for modifications make the whole process unable to carry through. Based on the definitions of Humphrey [55] with the ambition of helping software engineers to develop skills and habits for industrial demands can be done with rapid developing methods because of nature of the software development. Moreover, rapid differentiation of software environments necessitates the usage of adoptable and versatile software product [53].

Additionally the control of the interrelation of every module in the software system requires related work. Creating, developing and maintaining a software system can be resembled in different manufacturing process and the “assembly line” of it generally covers and requires more work to do.

Developing from a simple program to a large scale information system there is a need for software development methodology. Moreover, software engineers are always in pursuit of developing new software techniques to make the development processes easier to build in accordance with the satisfaction basic demands and procedures as well as desired requirements of complicated and larger systems. As a consequence of this the outcomes of new software engineering techniques help rapid development of software products. Moreover, using software engineering techniques and methodologies create an essential approach to reach standards and chase the rivals on the market, without using any kind of software engineering technique and methodologies development progress may be interrupted with unwanted errors or defects. Too many companies rely on this kind of approach and produce many software products. At the end of the process their software product generally expensive, but not reliable. To solve this issue there is a need for better training on software engineering [91].

According to Calderon et al. [25], despite the general importance and urgency of this issue and also increasing demand for qualified personnel software project management lectures and syllabus contain too many theoretical and relatively not unconcerned contents for the students who are the future professionals. The general awareness and knowledge on a specific software development methodology and an appropriate life cycle framework is necessary to further implementations of ongoing business and also establishing new software engineering techniques. Boehm [20] states that there is a need for novel methodologies due to rapid changes and educational deficiency. In addition to this, software engineering and its related practices require analysis which a manager has to overcome enhancing and using products while preserving requirements of demands [88]. Kling and Scacchi [66] agree that there is a need for clear definitions of techniques in order to foster links between social prospects and software attributes. Without making clear distinguishes and understandings from managerial topics, it can be problematic to make further predictions about project. All of these denouements may help and show the importance of consciousness and awareness of development of software.

ISO / IEC 12207 Software Life Cycle Process [58] is a comprehensive framework for organizations to continue to their software projects in a more professional and well-planned way. However, there is not enough awareness about the benefits of software development methodologies and software life cycle frameworks in Turkish software industry. According to TSE (Turkish Standards Institute) [103] there is only a single company, which has the certificate of ISO/IEC 12207 Software Life Cycle Processes. Due this fact ISO / IEC 12207 training has potentially important factors especially for students who are future engineers in the industry and the organizations where they want to improve their software development units by selecting a set of essential processes. Generally it is important that maintaining and monitoring performance of the processes and their sufficiency should be permanently improved. This approach requires training and it is commonly performed by a dedicated seminar with a kind of lecture approach. Because of the dynamic changes of software system requirements and also their uniqueness such kind of training does not cover the fundamentals of software development and essential details of ISO / IEC 12207 processes adequately.

In this study, the main aim is to create a serious game for establishing awareness and knowledge about ISO/IEC 12207 software life cycle processes to help students to improve their understandings and decision making skills on software development processes through the life cycle of the software product. Although ISO/IEC 12207 software life cycle processes consists of the detailed definitions of processes, activities, and related tasks, it does not consists of any contents, which are potentially useful for creating an awareness and beneficial for decision making towards software product. The standard is designed to be a reference guide to read the substantial details of the processes, activities, and tasks.

In order to foster the awareness and the basic knowledge on processes, dialogues can be considered as convenient interaction tools. Additionally, visual elements and contents are also helpful medium to improve the understandings of the concept around individuals. Based on the findings of Vyas et al. [109] collaboration between people mostly relies on the communication between them. This communication can be related about ideas, intentions and knowledge. It plays important role in work environment to continue the information transfer and collaboration of the work. According to Bryson et al. [24] dialogues about useful information helps to engage employees in both discussions and employment relationships. Findings of these researches shows that even simple dialogues can be more accurate than just reading reports and procedures. Dialogues are conversations between people. It is commonly used tool in public and science in order to express behaviors and attitudes with functional

rules and words [5]. In order to overcome struggles and to resolve problems that occur within human societies dialogues have been used as powerful and convenient tools. Today, they are actively used in every aspects of life [64]. In general societies have understood and used the effectiveness of dialogues and tried to solve problems and conflicts that occur in daily life. Additionally, they all have experienced the importance of involving dialogues to understand certain subjects especially in educational concepts. Today, there is growing demand of creating applications of dialogue-based games in order to satisfy many subjects such as education, social, political, and health care [64]. On the other hand, there is a need for creating suitable environments for combining dialogues and objects in order to express related subjects and notions. 3D virtual environments and visualization of stuff endeavor many conveniences over traditional methods for representing goods, environments, and landscapes [67]. Resemblance of 3D renderings can be enhanced to human models in order to get satisfying perception of realism. Moreover, in simulations and serious games modeling and animating objects can get attentions easily while involving goals through game play in the virtually described environments [67]. Virtual environments can be used to monitor behaviors and actions of individuals [6].

In the proposed game, the combination of dialogues and virtual office environment are the key game mechanics for improving the awareness and knowledge of the students. Furthermore, there are dialogue based implementation of particular processes, which creates the essential awareness of ISO / IEC 12207 processes and virtual office environment with different units. By this way participants does not only read the fundamentals of the standard but they also participate the software development life cycle by monitoring essential phases.

The main structure of the study includes 7 chapters with this introduction chapter. This thesis covers the following chapters;

Chapter 2 contains the whole literature review of this study. In general, it emphasis on the details of the software engineering field with the definitions of software life cycle, software process model, essential activities and processes which are important for all development models and detailed information about Waterfall, Incremental, and Agile development models with their related tasks. Additionally, it contains substantial details and fundamentals of ISO/IEC 12207 Software Life Cycle Processes with its processes that are used in this study and additional updated processes with reference tables. It also mentions the terms that are frequently used in the standard and tailoring process of ISO/IEC 12207. Moreover, backgrounds of related works are introduced in this chapter. Thereby, definition of games,

game mechanics, gamification and serious game concept with its details and supportive examples (that are related with the study via their attributes) in presented with highlighting the distinctive features.

Chapter 3 covers the details of methodology which was conducted in this study. Additionally, in this chapter, research techniques, procedure of methodology and the participants of this study are clarified.

Chapter 4 contains the details of design tools and techniques which are required to develop the proposed serous game. Hence, the tools and the development environments such as Unity3D, Blender, Mixamo Fuse are described with its features and specified usage for this study.

Chapter 5 presents the details of implementation techniques in order to accomplish the development of this study. In general, it covers the main features of created sessions which are used to validate the defined methodology. These sessions are named as paper-based learning session and virtual learning session. Their distinctive features and main functionalities are described. At the end of this chapter features of the proposed serious game with its defined processes and quests are presented with diagrams and screenshots of game play duration.

Chapter 6 gives emphasis on the description of the analysis of data and the results. It starts with the demographics of participants and continues with survey scoring, analysis of the data, threats to validity, and validation interview sections respectively.

Chapter 7 summarizes the study with its attributes, discusses main outcomes of the study and explains what will be done for this study in the near future.

Chapter 2

2. Review of Literature

2.1 Introduction

This chapter briefly explains the details of software engineering field and its related topics such as life cycle models, development methodologies and their processes. It starts with the fundamental background information about activities that are generally used by all life cycle models with different scopes and importance, and continues the section that covers the definition of software life cycle model and software process model. Additionally, there is a brief explanation about essential activities for software engineering. According to Sommerville's [95] discussions, detailed information which is related about general statements of all processes in software engineering. After explanations of general processes and related activities, the chapter continues with the definitions and attributes of Waterfall development model with its intended usage of current study. Moreover, Incremental development model and agile development models are described.

2.2 General Activities in Software Development

The early models were designed to sustain a conceptual arrangement for adequate management of the advancement of the software products and systems. This kind of model or in other words schemes basically serves planning, managing, and coordinating development of software systems.

From sixties various kinds of descriptions of software life cycle models have been dropped in. These software life cycle models generally consists or modifies different versions of activities that are listed at the table below. See table 1.

Name of the activity	Description
System Initiation/Planning [89]	This activity initializes the planning phase with clarifying feasible systems, their supplements, their basic mechanism and if they are formerly used it reveals their origins and automation settings [88, 89].
Requirements Analysis and Specification [89]	This activity basically identifies necessary actions and desired attributes that new software product or software system will have. Generally it covers the operational characteristics, resource management, and support and maintenance infrastructure of the software system [85, 88, 89].
Functional Specification or Prototyping [85]	This activity identifies the computation relationships between operations and systematical behaviors of software system [89]
Partition and Selection [89]	According to defined requirements this activity divides the software system into directable parts which transform subsystems. These subsystems are used to determine the need of new, already existing or reusable software product or software system is available to the necessary parts [88, 89].
Architectural Design and Configuration Specification [89]	This activity identifies the connections and communications between interfaces, subsystems, components, and modules. This identification details the design and supplements the configuration management [89].
Detailed Components Design Specification [89]	This activity identifies the usage of data within modules. The data resources are transformed from input to output for

	designed modules of components [88, 89].
Component Implementation and Debugging [89]	This activity mainly converts preceding procedural and conceptual specifications into source code. According to implementation of operational source code this activity basically validates the desired operations [88, 89].
Software Integration and Testing [89]	This activity sustains and confirms the general integrity of the architecture of software system. It checks the completeness of source code in the modules, consistency of modules, connections between interfaces and their specifications. Additionally it validates the usage performance of subsystem and software system and their conformance with requirements [88, 89].
Documentation Revision and system Delivery [89]	This activity mainly converts procedural recording of software system descriptions into well-defined documentations such as user guide [89].
Deployment and Installation [89]	This activity provides steps and directions to successfully install the proposed software product into a computing area. Then it configures operational necessities, user privileges, lastly it runs some set of defined test cases to confirm the validity of required software system action [85, 89].
Training and Use [89]	This activity provides assistance with instructional definitions during the understanding of product's capacity in order to use system efficiently [89].
Software Maintenance [89]	This activity supports the main usage of software product in its objective area by sustaining advancements, updates, and fixes

	[85, 89].

Table 1 General Activities for all Life Cycle Models

2.3 Software life cycle models

To make a definition for the life of the system, it can be stated like descriptive or prescriptive to explain how a software system or product can be evolved [89]. In descriptive models the explanation is used to form a basis to understand and improve software development processes. On the other hand the explanation of prescriptive models can be used to develop how a software system can be established. They are used as a guideline and they are neater to get benefit from developing a software system. However, descriptive life cycle models characterize particular software system within specific attributes and settings. Although they are different with their definitions both of them suggest many purposes for software life cycle models. For instance, an organization guideline is important to manage plans, personnel, budget, and schedule. Therefore, documentation is essential and specification of what kind of them is required for user should be defined clearly. As a consequence of them, The basis of software engineering methodologies and tools are required in order to conduct processes [89, 3].

2.4 Software process model

The term generally covers interconnected array of activities, tasks, objectives, and events that combine a formulation process of accomplishing and developing software evaluation [89]. This kind of model is more generalized and formalized than software life cycles and it may be used to develop software life cycle activities.

In addition to this software process networks may be monitored to understand numerous interrelated task chains [21]. These chains cover “non-linear” array of operations that alter assets into software systems or products. The definition of non-linearity can be explained like being flexible to various and multiple alternatives of actions to iterate the general progress. Additionally, these chains of tasks are viewed as the units of computational work, for instance, user commands, selections and inputs that consist cooperation between user and

defined program. This cooperation is “structured discourses of work”. In addition to this the chains have started to be known as “workflow” [59]. These chains can be in form of either prescriptive or descriptive, and they contain tasks to be followed [89].

Software processes are the combinations of akin actions activities that help to accomplish producing a software product [95, 3]. These activities are capable of involving software development from scratch but generally today’s applications are developed with the practice of modifying existing versions or configuring predefined software system components. According to Somerville beside many software processes there are four main activities essential for SE such as software specification, software design and implementation, software validation, and software evolution. In conformance with Somerville’s concerns the table below gives the descriptions of these four activities. See table 2;

Name of the activity	**Description**
Software specification	This activity is creating the definitions of software functionality and its constraints while operation phase [95, 2].
Software design and implementation	This activity is related about designing and implementing the software product in conformance with specifications [95].
Software validation	This activity generalizes the validation process that is essential while ensuring the customer desires [95, 2].
Software Evolution	This activity involves the duration of evolution of the software to satisfy changing demands and different needs of customer [95].

Table 2 the Four Essential Activities for Software Engineering

In various forms of software development process those activities are part of the processes and comprehensive ones can be divided into sub-activities to maintain the fundamental part of the related process. Without activities there is no chance to specify or define processes and

their scope of mission. Like activities the processes also needs some essential definitions. Defined process descriptions must contain products, roles, and pre and post conditions [95]. In conformance with these concerns about process the table below gives the descriptions of statements that process should have. See table 3.

Name of the Term	**Description**
Products	Every activity of a related process has some outcomes. Dependently these outcomes may become reference or input for another process and implicitly its activity [95, 3].
Roles	All processes in any software process model have some responsibilities for related parties or personnel who are involved in the duration of process [95].
Pre and post conditions	From the view of different activities' accomplishment that can be before or after, there are conditions and statements that are waiting for approval to continue the iteration of processes [95, 2].

Table 3 Statements for Processes

Generally man organizations develop existing models and make hybrid of different types. In addition to this many other creates them from scratch to suit their needs. However common categorization can be stated like plan-driven processes and agile process. In conformance with the discussions of Munassar et al. [73] on plan-driven process consists of process that are planned and their progression is monitored and expected to map predefined plan. On the other hand agile process does not require a timely manner plan instead they are in progression with incremental approach and they stay open to reflect the changes of customer needs.

2.5 The waterfall model

This is the model that is the first published software development process [85]. Because of the shape of the phases the model is known as "waterfall". Figure 1 shows the main structure of the Waterfall model [85]. This model's attributes are in the principle of plan-driven process. One of the main primary purposes of Waterfall software development is being on assistance to manage large software development projects in organizational preferences [85].

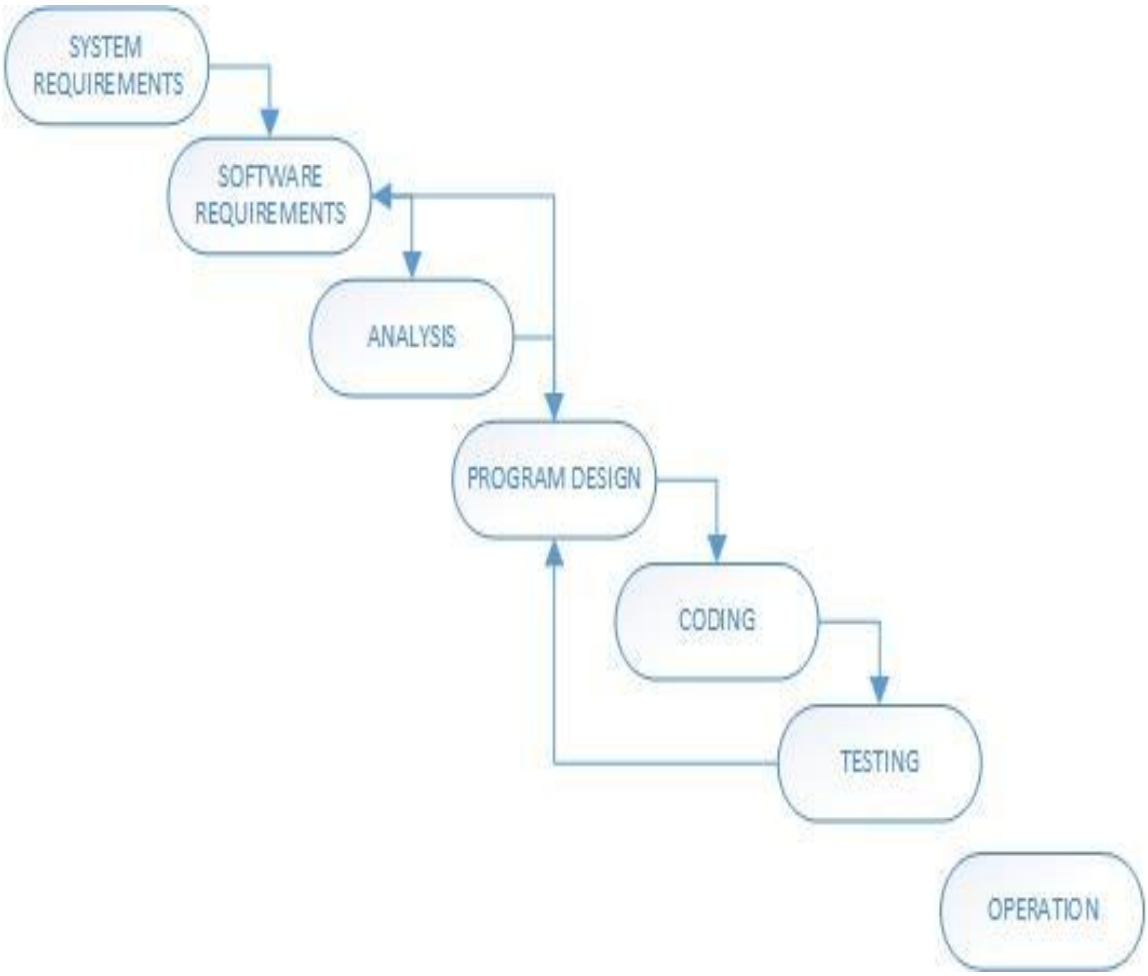

Figure 1 The Diagram of Waterfall Development Model [85]

An organization should schedule all activities before the software project start. Waterfall model consists of five fundamental activities. The table below explains the definitions of details these activities. See table 4.

Name of the activity	**Definition**
Requirements analysis and definitions	This activity determines the software system's goals, constraints, and services according to user preferences. These attributes are then defined as a system specification [95, 73].
System and software design	The process of design of the software system requirements can be allocated in either software or hardware parts of the entire system. Then according to this allocation the architecture of the software system is established in overall [95].
Implementation and unit testing	This activity treats the goals of the design activity as a set of interrelated programs, and their implementation phase until getting a software product, while unit testing verifies that every implemented unit and related specifications [73].
Integration and System Testing	The integration activity gets the individual units of program as integrated software to test the desired specifications have been met. According to this testing the software system become available to deliver to the customer [95, 73].
Operation and Maintenance	This activity is the last and the longest chapter in life cycle. The software product or system is installed in a suitable actually desired computing environment

	and monitored while it's operational usage. Maintenance activity on the other hand involves detecting and correcting defects and possible problems that are not encountered before. It also improves the software system with the direction of new desires and requirements [73].

Table 4 Activities of Waterfall Model

The mechanism behind the waterfall model is that continuation of the phases that are dependent on sequential phases. In waterfall model, a phase should not start until the previous phase has completed its activities. Each phase creates related documentation and they should be approved. In this mechanism phases use these documentations as a feedback or awareness of what to do and son on [73].

Beside the attributes of waterfall model software development is not a linear phase so there may be faults or stops during the phase and this makes the planned structure to behave badly according to user requirements, because their adequate implementations have to wait or will not completely met with requirements. These are the downsides of waterfall model but it is designed to be used with well-understood requirements and a well-defined process plan.

2.6 **Incremental development**

In incremental development software project is initially implemented and exposed to the user to get their feedback and according to this feedback the software project has started its evaluation phase which includes several versions of updates and improvements until an adequate and appropriate product has been developed [95]. See figure 2;

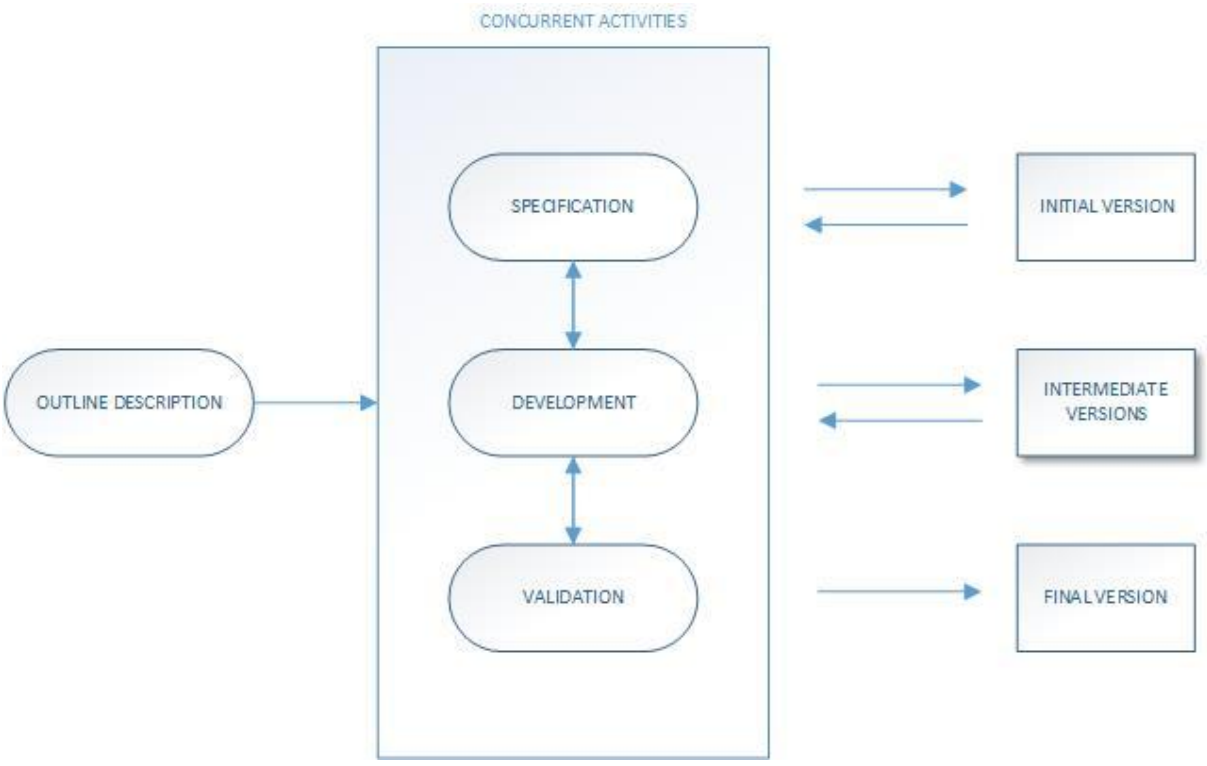

Figure 2 the Diagram of Incremental Development Model [78]

Incremental development bases on developing software systems with incremental releases which require firstly providing the most essential operational specifications and then providing the improved versions of software system. This continues with regular intervals [15]. The activities that are specification, development, and validation are interleaved. They are not separate and individual activities. This approach makes rapid feedback cycles. In principle incremental maintains the development of a software project with set of steps which makes the whole process cheaper and also easier because necessary changes in the software are available while its development phase [15]. In some cases Incremental Development can be either plan-driven or agile or the combination of both model. In plan driven case, software

system are iterated by planned increments on the other hand in agile case the first and second in increments cover more scope of usage that are the essentials and later increment are iterated according to the customer desires [95]. Generally, every increment consists several improvements of the functional operations of the software that are specified by customer. In addition to this approach, the early steps cover the most important and the most required attributes and functionalities of the software project, so the customer can aware of the operational functionalities of the specified requirements and start to evaluate the developing product. These attributes in general make Incremental Development beneficial to the business environment because the cost of changing user desires is reduced and the customer feedback can be taken in early steps of the development and also there is more rapid delivery than waterfall model with customers' early usage with the familiarity of the product. With time planned releases of software and maintenance methods Incremental model is popular and it is used by various organizations [89].

All those attributes of Incremental development are important and visible for both user and developer. However, without regular milestones or deliverables managers may have trouble to monitor the progress. Reflecting every increment or version can be problematic with its documentation process and also its costs. Furthermore, adding new software that can be a single unit of program, software service or an integration of related product can cause problems in later increments that are obviously harm the all software structure.

The distinction between plan-driven or agile development is dependent on the complexity and the scope of the entire software project. Disruption of adding new attributes and units of software are not usual and common factors. Additionally, larger systems need more stable infrastructure to increment additional attributes and versions of the product [95].

2.7 Agile software development

Due to global and rapid pace of changing the needs, opportunities and conditions creates a competitive environment so managers and developers have to respond to them. This makes implicitly increases the pressure on software development obviously. The need of faster development is becoming an essential approach day by day. The essential stage of rapid development does not specifically requires the obtaining requirements, system or software design, development and testing phases while the duration of software project, so traditional waterfall and incremental approaches and their attributes are left behind because they serve the end product after a long time that it actually specified. Rapid software development is required quick production and this production consists of series of increments rather than a single unit development. These increments include new software system attributes and functionalities. In general the most well-known agile method is Extreme Programming (XP), and according to users' explanations and their reports and customer involvement can be combined in project execution [89].

Based on Sommerville's [95] discussion on rapid software development, there are several fundamental characteristics which maintains specifically identified goals during the software development life cycle. The first concern is related about some processes that are requirements specification, design, and implementation. There is no need of detailed system specifications and documentations about system design. These are maintained by development process with the implementation of source code. Instead of well defined user requirements the documentation about only the most important attributes of the software system is enough to further implementations. Second concern is about the development of the software system. This process includes serious of versions of the product where the user evaluates each version and if there is a required change necessary implementations should be completed in upcoming or later versions of the software system. The third and the final concern is about the user interfaces. The user interfaces should be designed quickly with the help of icons, drawings and various visual elements. The generation process of the user interfaces depends on the usage and target user [73].

Careful project planning is an essential approach and the search for better software development depends on it by the diameter and the ramification of project. Progressing with large and expected to be long lived software products are importantly under concern with planning. However a predefined strategy for the development of the large software system

which is in principle of plan-driven approach is applied to a standard software project that can be managed more modestly creates problems and dissatisfactions in progress. According to this dissatisfaction engineers and developers proposed “agile methods”. In agile methods especially the development team is expected to focus on the development of the software rather than its design and documentation [95].

Agile methods generally rely on an incremental approach and they are intended to deliver the software product rapidly to the customers. Especially, it relies on the collaboration and contribution of customer ideas and opinions over the negotiations and contract. In addition to this, individual interactions play important role over specified tools and the processes [95].

To clarify the principles behind the agile development method the principle table (see Table 5) presents additional description.

Name of the principle	**Description**
Customer Involvement	In agile method the expectation from customers is their involvement through the development of the desired software product. The roles of them should be stated like providing and prioritizing the system requirements and evaluating these by product’s iterations [73].
Incremental Delivery	The fundamental principle of an agile method’s continuation through the life cycle is the incremental delivery. With the customers’ expectations and specifications each increment must be shaped [73].
People not process	This principle is actually specifies the importance of members who are responsible with developing the desired software product. Their skills and perspectives are far more important than procedural phases and they should be open and should not be

	circumvented by rules and requirements that are from particular process [95].
Embrace change	The flexibility of an agile method is being aware of dynamic system requirements which can be changed in any time according to user preferences and desires, so the development of software system should react these rapid changes and continue without defects [95].
Maintain Simplicity	Complexity is not a concern in an agile method because of the effort of being rapid. In both software infrastructure and the development processes the important part is focusing on simplicity while eliminating complexity [73].

Table 5 Agile development Principle Table

2.8 ISO/IEC 12207 Software Life Cycle Processes

2.8.1 Introduction

This section covers the substantial details of ISO/IEC 12207 Software Life Cycle Processes. It emphasizes on essential knowledge with detailed descriptions of processes with their activities. After details of processes following sections present additional and updated process tables with ISO/IEC 12207 references, to further information. It also presents the terms, which are frequently used in processes and the study, with description table. Last section of this chapter gives brief explanation of tailoring process of ISO/IEC 12207.

2.8.2 Background Information about ISO/IEC 12207

ISO/IEC 12207 is an international software engineering standard that defines the software engineering processes and activities, which are associated with software life cycle process from conception to end-product [104]. This standard defines a set of suitable roles for software practitioners and follows the plan-do-check-act cycle for improving the quality of the product [44].

ISO/IEC 12207 is based on the qualitative definitions of the processes, and therefore there are no implementation details of defined the tasks and activities [112]. Moreover, it does not measure the quality, it does not define specifically how to do activities and tasks, and it does not prescribe to particular methods, practices or tools. Its modular structure is suitable for tailoring purposes. Therefore, an organization can customize the necessary parts of the standard that are planned to be used based on the requirements of a software project [62]. Because of the high modularity of the standard, it is more easily to deal with factors that are affecting the software development such as complexity, schedule, cost, etc. In addition to that ISO/IEC 12207 can act as an inventory of processes, which give different perspectives to particular parts of the software life cycle process. These processes are categorized as organizational processes (i.e. management, infrastructure, improvement, training), supporting processes (i.e. documentation, configuration management, quality assurance, verification, validation, joint review, audit, problem resolution), and primary processes (i.e. acquisition, supply, development, operation, maintenance) [93].

ISO/IEC 12207 is a guideline based on a set of process descriptions for providing a base for adopting a role, which states a set of constraints for selecting process, activities or tasks that are required for software development. The standard proposes a set of views that can be used to label the processes connected to a role. To this end, it offers several different viewpoints as follows: contract, engineering, operating, quality management, and management views [62]. Firstly, there is a contract view that includes an acquisition process (i.e. for the acquirer) and supply process (i.e. for the supplier). Secondly, there is an engineering view which has a development process for product development and a maintenance process for up keeping the software. Thirdly, the operating view with the operation process that provides a guideline for operating the software. Fourthly, a quality management view that has six processes; joint review, audit, verification, validation, quality, and problem resolution processes [62]. ISO/IEC 12207 Software Lifecycle processes can be maintained by 7 main phases by any organization which has capability to support the standard's views and ability to handle software engineering requirements. These main phases are; requirements analysis, specification, design, coding, verification & validation, installation, maintenance & support [58]. Although the standard has explicit definitions and substantial technical content professionals and individuals may find it troublesome to follow every process. In light of this remarks, the goal of the study is to investigate the possibilities of a game that is designed for teaching the primary concepts of ISO/IEC 12207.

2.8.3 Details of ISO/IEC 12207 Software Life Cycle Process

Similar to the definition of ISO/IEC 12207, definition a software life cycle starts with a requirement analysis based on a need and eventually the life cycle ends with the retirement of a product [4]. The standard has an architecture, which is built by set of interrelated processes, which are consequent to modularity and responsibility. While defining modularity under the conceptualization of the standard is about being unique with every processes and availability of being capable enough to handle all types of projects. The processes that are designed for the standard have a modular structure. From the practical point of view, the modules have the maximum cohesion and minimum coupling where each process supports unique functionalities as possible [93]. To clarify every part which is associated with the life cycle has specific and well defined responsibility to take care. However, from a traditional point of

view, the modules of a life cycle should be studied distinctively. To understand the basics of the standard, the definition of organization and party should also be elaborated. The terms organization and part are required to highlight different viewpoints that can be acquired using the standard. The term organization defines a group of persons (or authorities) with a set of responsibilities that are organized for a particular objective. However, the party defines an organization that enters into a contract, which can be either from an organization or more. The name that is given to a part is usually correlates with the name of the process it performs (e.g. an acquirer is involved with the acquisition process). There are several roles, which can be directly related with the process names from the standard such as acquirer, supplier, implementer, maintainer, and operator, etc. [58].

The ISO/IEC 12207 processes can be organized into three main categories: primary processes; supporting processes; and organizational processes (see Figure 3). Primary processes described by the standard are acquisition, supply processes, development processes, operation processes, and maintenance processes. The goal of a supporting process is to support other processes while fulfilling a function. The supporting processes are identified as documentation processes, configuration management processes, quality assurance processes, joint review processes, audit processes, verification processes, validation processes, and problem resolution processes. The organization processes are employed by an organization to manage, control and improve the life cycle processes.

There are seven process groups that are defined by ISO/IEC 12207 which can be accomplished during the life cycle of a software system. Each of the life cycle processes within those groups can be defined with respect to its goals and expected outcomes. Figure 4 shows the activities and tasks that should be carried out to accomplish these outcomes [58].

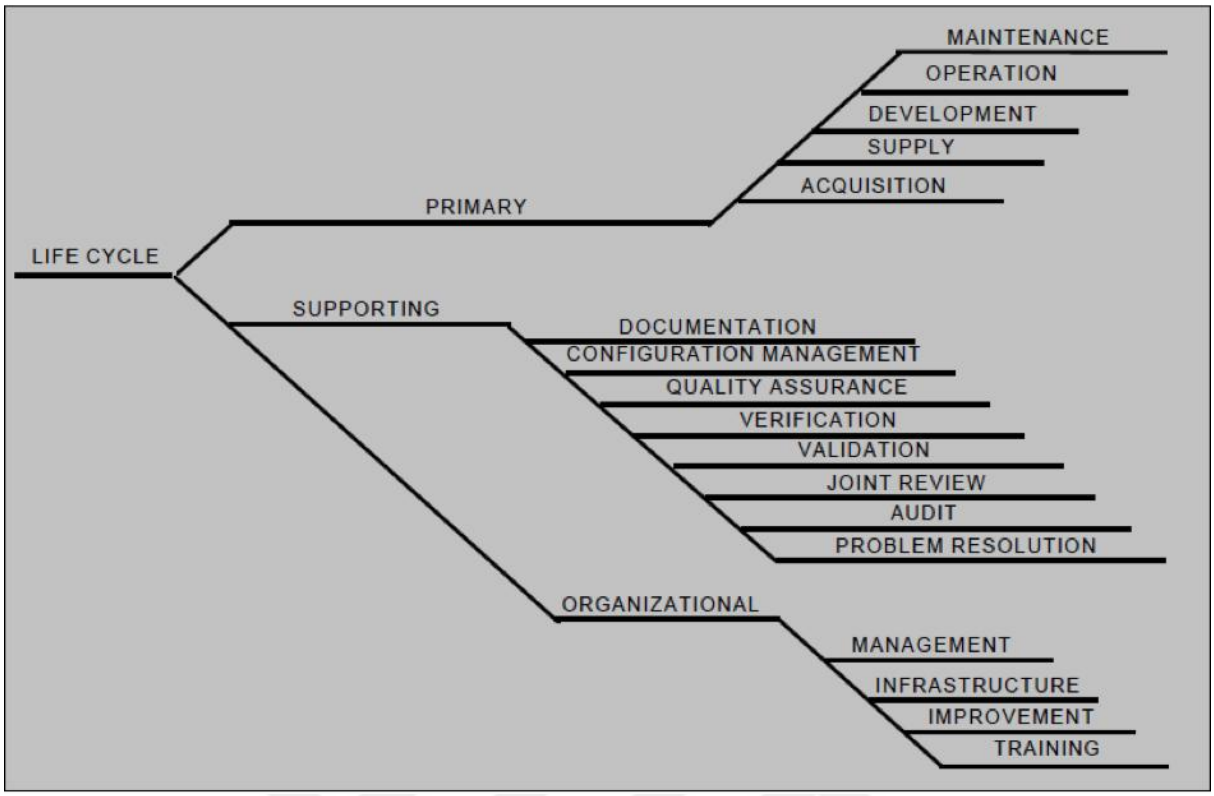

Figure 3 Life Cycle Processes of ISO/IEC 12207

- Agreement Processes
- Organizational Project-Enabling Processes
- Project Processes
- Technical Processes
- Software Implementation Processes
- Software Support Processes
- Software Reuse Processes

Basically, the design of ISO/IEC 12207 software life cycle process was constituted with a set of complementary components [58]. For instance, each process has its own activities that cover cohesive tasks where tasks have necessary actions [114]. A task takes several type of inputs (e.g. data, information) and generate outputs (e.g. data, information). A set of verbs such as will (for declaration of purpose), shall (for binding provision), should (for recommendation) is used to express requirements, recommendations or acceptable actions. The standard utilizes the fundamentals of quality management techniques which are the integral and indispensable parts of the total life cycle. Therefore, each process is a basic implementation of plan-do-check-act (PDCA) cycle. To implement ISO/IEC 12207 properly,

it is important to know that each process and individuals who are in charge of related processes must be aware of their particular roles and responsibilities in all processes. Based on the assigned roles, evaluations of particular tasks have to be carried out properly within the software development organization [114].

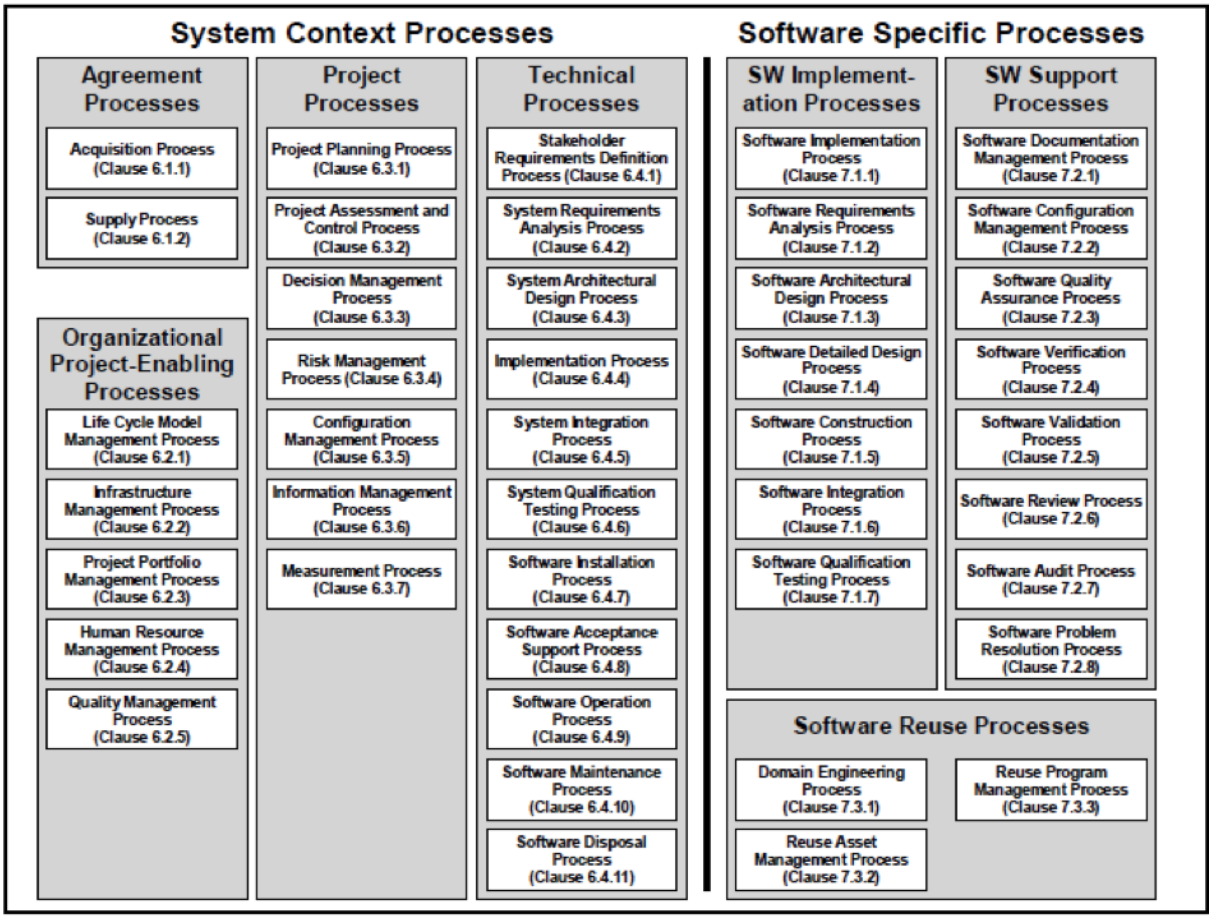

Figure 4 ISO/IEC 12207 Life Cycle Processes Groups

This standard requires outputs and these outputs have to be documented, but there is no specific or predefined format for any types of output to be documented. The organization can use their documentation methods also get benefit from standards. In addition to documentation base-lining is important issue. The standard requires the baselines of software related tasks and activities such as software requirements, software design, and coding. Baselining is a process in which quality and effectiveness of a method assessed by comparing before and after a change. The idea of baselining can be considered as an effective method to mitigate risks that establish certainty in milestones, to control costs and deadlines by prohibiting unnecessary (i.e. unplanned or open) changes in all parts of the software

development life cycle [93]. In particular, baselining can happen while joint review for clarifying the relations between acquirer and supplier. Nonetheless, it is not necessary for projects to perform baselining, which is the responsibility of the Development Process. It is not related with Configuration Management Process and it is not a must. Consequently, ISO/IEC 12207 covers the total software development life cycle and it is relatively a complex process especially when it is based on a viewpoint of variety of stakeholders who are working together in the same software development project.

However, the standard should be examined in the context of organizational objectives where the requirements of a project may hinder a solid interpretation. To avoid improper usage of the standard, these prerequisites below shall be met [93]:

- The requirement of qualified personnel;
- The requirement of understanding the organization's policies;
- Experience with the project's environment;
- Develop an understanding of the standard.

2.8.3.1 Acquisition process

Acquisition process mainly describes responsible actions which will be performed by the acquirer who obtains software service, product or system via a contract. The organization, which needs software, is defined as owner and this owner may contract the some parts or whole acquisition task to a responsible personnel. Then acquirer represents these for user [93].

The acquisition process starts with defining needs and tasks according to desired product or service and continues with request for proposal for supplier, selection of supplier and management of the acquisition process to the acceptance of selected product or service [93].

At the end of the process the required needs and product/ services acceptance criteria are defined [93]. This criterion substantially covers the acquirer's and organization's statements of acceptance. It is possible that there will be more than one supplier and according to that situation one or more of them can be selected. The whole acquisition process is monitored by acquirer so every specified statement and the defined constraints which are possibly schedule, cost, and compatibility are examined.

Acquisition process consists of several activities that are listed in table below;

	Name of the activity	**ISO/IEC 12207 Reference [14]**
Acquisition Process	Initiation	5.1.1
	Request-for-Proposal	5.1.2
	Contact Preparation and Update	5.1.3
	Supplier Monitoring	5.1.4
	Acceptance and Completion	5.1.5

Table 6 Activity Table of Acquisition Process

The acquirer starts the process after the analysis of the system requirements. He/she begins with describing the necessary needs to be acquired according to the organization's statements. The requirements should include and satisfy the concepts about organizational levels and user preferences [58].

Activities occurrence order in the acquisition process table should be in the order such as, the first three activities starts before the agreement of requirements, the last two activities starts after the agreement of the requirements.

2.8.3.2 Supply process

This process contains activities for the supplier. The process can start with two options. The first one is initiation comes up with a decision of preparing a proposal to answer an acquirer's request for proposal and the second one is signing the contract with the acquirer to provide software. Specifically this software means software service, which can refer to several aspects such as development of a product, maintenance of a software or operation of a system. The whole process includes identification of procedures that are needed for development and its management, the development and execution, and delivery of the software to acquirer [93, 58].

After implementation of this process acquirer is already defined and the response to his request is done. Additionally the compliance with acquirer and supplier for processes such as develop, maintain, operation, and installation of software is established [58].

The Supply process consists of several activities which are listed in table below;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Supply Process	Initiation	5.2.1
	Preparation of Response	5.2.2
	Contract	5.2.3
	Planning	5.2.4
	Execution and Control	5.2.5
	Review and Evaluation	5.2.6
	Delivery and Completion	5.2.7

Table 7 Activity Table of Supply Process

At the final steps of the supply process the software product or software service is submitted to acquirer with conformance of the admitted requirements. Moreover, end product is installed again according to agreement [58].

Activities occurrence order in the supply process should be in an order such as, the first two activities starts before the agreement of the requirements, and the last five activities start after the agreement of requirements.

2.8.3.3 Development process

This process includes tasks and activities that are related to developer/developers of the software product/service. Development may involve both development of the software from scratch and manipulating or modifying the existing software product [93].

Additionally, for the conceptualization of development process it can be employed in two different ways for instance, it can be a methodology for developing modules and prototypes or it can be a whole process to produce a product or services [93].

The developer organizes this process at project level consecutive to management process. The management process is initiated by this way. The definition or selection of an appropriate software life cycle model according to organization's defined specifications which are about scope, complexity, etc. can be done by developer whether or not they are stated as specifications at a documentation or agreement. Moreover activities and tasks of this process are summarized to selected model with conformance of needs and desires [21].

The developer places the outputs of the process according to documentation process and its activities for configuration management process. Additionally he/she tailor the usage of required tasks, methods, software development tools, etc. To integrate every necessary task developer should maintain requirements analysis, feasibility of software project, and operation [21].

Development process consists of several activities that are listed in the table below;

	Name of the activity	ISO/IEC 12207 Reference [21]
Development process	Process Implementation	5.3.1
	System Requirements Analysis	5.3.2
	System Architectural Design	5.3.3
	Software Requirements Analysis	5.3.4
	Software Architectural Design	5.3.5
	Software Detailed Design	5.3.6
	Software Coding and Testing	5.3.7
	Software Integration	5.3.8
	Software Qualification Testing	5.3.9
	System Integration	5.3.10
	System Qualification Testing	5.3.11
	Software Installation	5.3.12
	Software Acceptance Support	5.3.13

Table 8 Activity Table of Development Process

2.8.3.4 Operation Process

This process is related about the operator of a project or organization. The operation part involves both software system and support for users [93].

Because of the relevance of operation of a software product and integration with system, this process and its activities implantations make reference to the entire system. The operator should have and if it does not exist he/she should create a plan for operational activities which are conducting the essentials of the process. Additionally, operator should do mainly record and track the encountered problems and provide feedback to the problem resolution process. Moreover the operator should provide assistance to the user request and keep track of them, then this request will be provided to the Maintenance process [21].

Operation process consists of several activities that are listed at below table;

	Name of the activity	ISO/IEC 12207 Reference [21]
Operation Process	Process Implementation	5.4.1
	Operational Testing	5.4.2
	System Operation	5.4.3
	User Support	5.4.4

Table 9 Activity Table of Operation Process

2.8.3.5 Maintenance Process

It defines activities and tasks for maintainer of the software project. It is triggered by requirements of system modifications due to an error, a problem or any improvement for the code. Additionally, if there is a modification for the documentation again the process goes under concern. Main objective of the process is protecting total integrity of the software while modifying necessary entities. The process needs a trigger to be activated but ends with the retirement of the system [93].

The maintainer should maintain, develop and execute the necessary tasks, procedures and plans which are necessary for the process. According to an existing of problems or any necessary modification which can be requested by user or organization, maintainer should document the results of desired modifications' analysis and present required implementation plans and alternatives. If there is an occurrence of a "migration" of software system or software product from one environment to a new environment (a place, party or organization) the migration should be conducted according to its tasks and maintained by maintainer monitoring [21].

Maintenance process consists of several activities that are listed at below table;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Maintenance Process	Process Implementation	5.5.1
	Problem Modification Analysis	5.5.2
	Modification and Implementation	5.5.3
	Maintenance Review/Acceptance	5.5.4
	Migration	5.5.5
	Software Retirement	5.5.6

Table 10 Activity Table of Maintenance Process

2.8.4 The supporting Processes

ISO/IEC 12207 consists of eight supporting processes. Supporting processes can support other processes in case of various purposes to improve the quality of whole project. An instance of this process is appealed to in demand of acquisition, supply, development, operation or maintenance process, or another supporting process [93].

2.8.4.1 Documentation Process

This process is about saving and recording information that is produced through life cycle of project and concerned with developers, managers, and users [93].

To conduct the necessary activities and make proper communication options to smaller modifications the standard provides documentation process. The communication with engineers and managers is important to maintain large-scale software projects with minimum number of problems accordance. During software life cycle planning and identifying

documents are indispensable. The documents are produced in accordance with the defined specifications and plan [21].

This process contains several activities that are listed at below table;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Documentation Process	Process Implementation	6.1.1
	Design and Development	6.1.2
	Production	6.1.3
	Maintenance	6.1.4

Table 11 Activity Table of Documentation Process

2.8.4.2 Configuration Management Process

This process is invoked for identification, definition and baselining in software system in order to monitor alterations and release of items. It is useful to record and report the current stage of any item and alteration proposals to assure integrity and exactness of them. Moreover, monitoring storage, handling, and delivery of necessary items [93].

During the software life cycle many administrative and technical procedures need to be identified, defined and applied. Configuration Process monitors these procedures and also ensures the consistency of items which are used in software project. Additionally, the process controls their storage and delivery conditions [21].

To maintain this process there is a need to develop a plan for configuration management. This plan should describe the configuration management process, its activities and tasks in accordance of schedule which is defined prior by an organizational level [21]. The responsibility in this process is performing the activities that are specified and making them connected with other processes. Generally these processes are Development Process and Maintenance Process. Properly ensuring the items' consistency and their identification control

must be done with unique identifiers and if there is a request of change, under analysis of the changes the modifications should be clearly stated.

This process contains several activities that are listed at below table;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Configuration Process	Process Implementation	6.2.1
	Configuration Identification	6.2.2
	Configuration Control	6.2.3
	Configuration Status Accounting	6.2.4
	Configuration Evaluation	6.2.5
	Release Management Delivery	6.2.6

Table 12 Activity Table of Configuration Process

2.8.4.3 Quality Assurance Process

Through the software project life cycle there is a need for adequate availability of assurance of software items and software products. Another important point is determining the quality assurance by being unbiased. To succeed this process should be controlled by internal or external organizations or persons who are not directly responsible for the implementation of the process and also generating software products [21].

Quality assurance process may be tailored to fit software projects' objectives and it should be in coordination and communication with verification, validation, joint review, and audit processes.

With the help of a plan which is designed for quality assurance process, its activities, and tasks should be well documented in conformance with the initial contract. The process should assure that all the required processes such as acquisition, development, operation, maintenance, and support processes are maintained according to the agreement and

organizational plans. Moreover, this process should assure that all software development environment, software testing environment, and engineers' development methods are matched with the agreement and organizational plans [21].

This process assures the or services with their agreed requirements and consists of several activities that are listed at below table;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Quality Assurance Process	Process Implementation	6.3.1
	Product Assurance	6.3.2
	Process Assurance	6.3.3
	Assurance of Quality Systems	6.3.4

Table 13 Activity Table of Quality Assurance Process

2.8.4.4 Verification Process

This process involves the verification of products or services according to a given task. It determines the outcomes of tasks' correctness to fulfill the requirements of previous tasks. Additionally verification process tails the evaluations of processes. It combines verification of, software requirements, software design, coding, integration, and documentation [93].

Software verification process is about confirming each software service or worked adequately whether it satisfies the desired requirements. At the end of the proper implementation of this process a defined strategy for verification become concrete according to development and implementation. Additionally, all related software works have been gained specifically identified criteria. Furthermore defects and problems are monitored and identified so their results become more visible to the responsible parties and also to the customers.

Under the software verification process the main activity called verification consists of several tasks. At design verification implementation of design elements which covers inputs, outputs, interfaces, etc. have to be in an appropriate order of sequence. At the code

verification task to maintain verification activity the code should be readable, traceable and testable. Moreover, the ode part of the project properly implements the given design elements according to the requirements. At the integration verification task every components and units of software work have to be in a good integration with related parts of the software. This task is conducted with a proper integration plan. At the documentation task documents that are produced at the activities and processes have to be complete and adequate under a timely manner [58].

Verification Process consists of several activities that are listed at below table;

Verification Process	**Name of the activity**	**ISO/IEC 12207 Reference [21]**
	Process Implementation	6.4.1
	Verification	6.4.2

Table 14 Activity Table of Verification Process

2.8.4.5 Validation Process

Validation process checks whether the final software system or software product matches its desired usage. Validation process does not overtake any other evaluations. It supplements other evaluations [93].

The process is mainly about confirming the requirements which are specially intended the use of the end software product. At the end of acknowledged application of verification a specified validation approach is created and maintained. Additionally, particular criteria for the process and related software work are performed and identified. During the process implementation problems and defects are recorded and they become more visible to the responsible parties and also to the customer. To implement the process there is a need to a validation plan which have to be well documented, and the plan should consist the subjects that are related with the validation and resources that will be used [58].

The main activity of this process is called validation and it consist several related tasks. The tasks in this activity are generally formed in tests and the analyze period of these tests which are seeking for the validation of the end software product's satisfaction of its desired usage [58].

Both verification and validation are suitable for execution of acquirer, supplier, or independent party which can be located outside of the organization. If they are conducted by an independent party that also is a supplier or developer, this type I called independent verification process and validation process [93].

Validation Process consists of several activities that are listed at below table;

Validation Process	**Name of the activity**	**ISO/IEC 12207 Reference [21]**
	Process Implementation	6.5.1
	Validation	6.5.2

Table 15 Activity Table of Validation Process

2.8.4.6 Joint Review Process

Joint Review establishes structure to maintain the relations between acquirer and supplier. Circumstances of products of life cycle are reviewed for further approval. Main objective is evaluation of project's status and software products which can vary in different activities if they are suitable or not. Joint review may be conducted by two parties. This is generally conducted like one party reviews the other party and this is continued in the order of required review. Reviews which can be done in periodical intervals are necessary especially when the determined milestones are established [21].

The items, products or status should be agreed by different parties to supplement agreed statements. If any occurrence of problems which are detected during the joint review process they should be recorded and stated to the problem resolution process [21].

Joint Review Process consists of several activities that are listed at below table;

Joint Review Process	**Name of the activity**	**ISO/IEC 12207 Reference [21]**
	Process Implementation	6.6.1
	Process Management	6.6.2

	Reviews	
	Technical Reviews	6.6.3

Table 16 Activity Table Joint Review Process

2.8.4.7 Audit Process

The main objective of this process is determining the suitability of requirements, plans and agreement are appropriate. According to occurrence of determined milestones which are defined in software project's plans audit should be done. Like joint review process, audit process also needs being unbiased. Due to this, engineers, parties or organizations should not have direct responsibility of developing software products or services if they are in audit process and proposed personnel [58]. At the stage of any occurrence of problems, they should be recorded and stated to the problem resolution process. This process may be conducted by two parties. This is generally conducted like one party audits the tasks, activities or software products of other party [54, 21].

Audit Process consists of several activities that are listed at the table below;

Audit Process	**Name of the activity**	**ISO/IEC 12207 Reference [21]**
	Process Implementation	6.7.1
	Audit	6.7.2

Table 17 Activity Table of Audit Process

2.8.4.8 Problem Resolution Process

This process's main ambition is to detect problem in a loop pattern and then resolve them. Also take suitable actions to remove detected problems. The process mainly covers analysis and resolve of problems which occur during the implementation of different processes like development process, operation process, maintenance process etc. [21].

When a problem has been detected in any phase of the life cycle, this problem should be reported in a well prepared documentation. This process is maintained in a “closed-loop” pattern. All the detected problems with their reports enter this process and relevant parties are warned about the detection and the occurrence of the problem. When the problem is resolved again the problem’s details are recorded in conformance with agreement. Every problem in this process should be under a category and a priority attributes to maintain this process.

Problem Resolution Process consists of several activities that are listed at below table;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Problem Resolution Process	Process Implementation	6.8.1
	Problem Resolution	6.8.2

Table 18 Activity Table of Problem Resolution Process

2.8.5 The organizational processes

There are four Organizational Process. One organization maintains organizational processes to sustain corporate level actions. Additionally, one of these processes can support other processes with controlling manner [21].

2.8.5.1 Management Process

This process is related about software manager’s activities. Especially in acquisition, supply, operation, maintained processes [21].

Management process can be handled by various parties or a single party which has to manage its responsible processes. For execution of this process software manager is responsible. The initiation of management process is triggered by defining the requirements and necessary statements of the process which will be conducted. According to this definition

manager starts the feasibility of the usage of resources which will be used with activities while maintaining the processes. These resources can be mentioned like developers, items, environment, technology, software etc.

At the end of the management process after controlling and evaluating the outcomes of process, manager determines all software related actions and products whether they are matched with agreement or not.

Management Process contains several activities that are listed at below;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Management Process	Initiation and Scope Definition	7.1.1
	Planning	7.1.2
	Execution and Control	7.1.3
	Review and Evaluation	7.1.4
	Closure	7.1.5

Table 19 Activity Table of Management Process

These activities are very similar to the primary processes' but their goals are different in detail. However, every primary processes is the initialization of management process in general.

2.8.5.2 Infrastructure Process

This process ascertains the activities that build an adequate structure for any life cycle process. Such structure may contain software, hardware, techniques etc. [58].

To support other processes Infrastructures process is used. This process should be defined and well documented to satisfy the necessary statements of the processes. According to predefined factors such as availability of budget, schedule, plans and any other constraints that are the arguments of configuration of this process are used [21].

Infrastructure Process consists several activities that are listed at below table;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Infrastructure Process	Process Implementation	7.3.1
	Process Assessment	7.3.2
	Process Improvement	7.3.3

Table 20 Activity Table of Infrastructure Process

2.8.5.3 Improvement Process

Improvement process is completely about improving the software life cycle processes. The standard defines and provides most necessary processes with activities and tasks for an organization. These can be stated in an order like acquisition, supply, development, operation, maintenance and support processes. All of those related activities cover how the process works, how to assess and how to improve them. The organization conducts these activities at the organizational level. This creates an experience gain environment which would benefit to the organization. The main ambition is improving the processes firstly at organizational level and then with the help of continuing projects, developing software techniques and future projects also get benefit from the basic improvement process [93, 58].

To maintain the main ambition an organization should develop a suitable documentation as well as process to control, monitor, and improve the processes. The reflections of improvements should be visible in organization's documentations [21].

Improvement Process contains several activities that are listed at table below;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Improvement Process	Process Implementation	7.3.1
	Training assessment	7.3.2

	Process Improvement	7.3.3

Table 21 Activity Table of Improvement Process

2.8.5.4 Training Process

This process is about establishing programs for acquire and train personnel and also related resources in managerial and technical levels. The process consists training, material generation to be used in a well-timed program [93].

This process provides knowledgeable and qualified personnel. Every process is dependent on these personnel and there is an essential and suitable training should be available before the software project life cycle duration.

To properly maintain the training process there should be predefined training plan which is addressing important schedules, available resources, and needs to be acquired. All the materials like manuals, presentations, equipment should be developed and documented [21].

Training process consists of several activities that are listed at table below;

	Name of the activity	**ISO/IEC 12207 Reference [21]**
Training Process	Process Implementation	7.4.1
	Training Material Development	7.4.2
	Training Plan Implementation	7.4.3

Table 22 Activity Table of Training Process

2.9 Life Cycle Processes Tables

Life cycle process groups, see table 23;

Name of the Process	ISO/IEC 12207 Reference [58]
System Context Processes	5.2.2.1
Software Specific Processes	5.2.2.2

Table 23 Table of Life cycle process groups

System context processes, see table 24;

Name of the Processes	ISO/IEC 12207 Reference [58]
Agreement Processes	5.2.2.1.1
Organizational Project-Enabling Processes	5.2.2.1.2
Project Processes	5.2.2.1.3
Technical Processes	5.2.2.1.4

Table 24 Table of System context processes

Software Specific Processes, see table 25;

Name of the Processes	ISO/IEC 12207 Reference [58]
Software Implementation Processes	5.2.2.2.1
Software Support Processes	5.2.2.2.2
Software Reuse Processes	5.2.2.2.3

Table 25 Table of Software Specific Processes

Agreement Processes, see table 26;

Name of the Process	ISO/IEC 12207 Reference [58]
Acquisition Process	6.1.1
Supply Process	6.1.2

Table 26 Table of Agreement Processes

Organizational Project-Enabling Processes, see table 27;

Name of the Process	ISO/IEC 12207 Reference [58]
Life Cycle Model Management Process	6.2.1
Infrastructure Management Process	6.2.2
Project Portfolio Management Process	6.2.3
Human Resource Management Process	6.2.4
Quality Management Process	6.2.5

Table 27 Table of Organizational Project-Enabling Processes

Project Processes, see table 28;

Name of the Process	ISO/IEC 12207 Reference [58]
Project Planning Process	6.3.1
Project Assessment and Control Process	6.3.2
Decision Management Process	6.3.3
Risk Management Process	6.3.4
Configuration Management Process	6.3.5
Information Management Process	6.3.6
Measurement Process	6.3.7

Table 28 Table of Project Processes

Technical Processes, see table;

Name of the Process	**ISO/IEC 12207 Reference [58]**
Stakeholder Requirements Definition Process	6.4.1
System Requirements Analysis Process	6.4.2
System Architectural Design Process	6.4.3
Implementation Process	6.4.4
System Integration Process	6.4.5
System Qualification Testing Process	6.4.6
Software Installation Process	6.4.7
Software Acceptance Process	6.4.8
Software Operation Process	6.4.9
Software Maintenance Process	6.4.10
Software Disposal Process	6.4.11

Table 29 Table of Technical Processes

Software Implementation Processes, see table 30;

Name of the Process	**ISO/IEC 12207 Reference [58]**
Software Implementation Process	7.1.1
Software Requirements Analysis Process	7.1.2
Software Architecture Design Process	7.1.3
Software Detailed Design Process	7.1.4
Software Construction Process	7.1.5
Software Integration Process	7.1.6
Software Qualification Testing Process	7.1.7

Table 30 Table of Software Implementation Processes

Software Support Processes, see table 31;

Name of the Process	ISO/IEC 12207 Reference [58]
Software Documentation Management Process	7.2.1
Software Configuration Management Process	7.2.2
Software Quality Assurance Process	7.2.3
Software Verification Process	7.2.4
Software Validation Process	7.2.5
Software Review Process	7.2.6
Software Audit Process	7.2.7
Software Problem Resolution Process	7.2.8

Table 31 Table of Software Support Processes

Software Reuse Processes, see table 32;

Name of the Process	ISO/IEC 12207 Reference [58]
Domain Engineering Process	7.3.1
Reuse Asset Management Process	7.3.2
Reuse Program Management Process	7.3.3

Table 32 Table of Software Reuse Processes

To clarify the definitions that are frequently used in both the standard and the game, it is important to know the definitions of particular term that are related with ISO/IEC 12207, see table 33;

Term	Definition	ISO/IEC 12207 Reference
Acquirer	*One of stakeholders who is responsible for acquisition process*	4.1
Acquisition	*First primary process about getting appropriate software system, product or service*	4.2
Activity	*Sub-task of a process*	4.3
Agreement	*Mutual acknowledgement about specifications under a relationship*	4.4
Audit	*A process which is done by independently in accordance with requirements*	4.5
Baseline	*A kind of specification that has been reviewed thereafter which can be used to be developed for further procedures*	4.6
Configuration item	*A type of entity that is located at the configuration phase with unique identification for a reference*	4.7
Contract	*An agreement between two parties according to laws or organizational prospects*	4.8
Customer	*Individual or an organization that are waiting for the product or service*	4.9
Developer	*Responsible party or organization for the tasks such as requirements analysis, design, testing, etc.*	4.10
Enabling system	*A kind of system that provides "system-of-interest" while life cycle, it does not contribute functionality for the operation process*	4.11
Evaluation	*Systematically identification and determination of particular entities with specified criteria*	4.12
Facility	*Equipment, buildings, tools, etc. are useful*	4.13

	for facilitating the action	
Firmware	*The combination of hardware and read only computer program that is equipped on hardware device*	4.14
Implementer	*Responsible for the implementation of the tasks*	4.15
Life cycle	*The process of the evolution of an entity from conception through the retirement.*	4.16
Life cycle model	*A kind of framework to act as a reference for processes, activities, and tasks.*	4.17
Maintainer	*Responsible organization for performing maintenance activities.*	4.18
Monitoring	*The process of examination to determine the status of activities that are maintained by acquirer and the supplier*	4.19
Non-deliverable item	*The entity which can be hardware or software that is not essential to be conveyed according to the agreement, but it can be used in development phase*	4.20
Off-the-shelf	*Already developed and accessible to use*	4.21
Operator	*Responsible for performing the operation of a system*	4.22
Organization	*Group of people and facilities in conformance with an agreement to accomplish particular tasks*	4.23
Party	*Small scale organization that enters into a contract*	4.24
Process	*Set of related activities that perform particular input and outputs*	4.25
Process purpose	*The objective of performing the process and related outcomes*	4.26
Process outcome	*The consequence of the accomplished implementation of the process*	4.27

Product	*Entity which comes with the end of the process*	4.28
Project	*The objective of creating a product according to defined dates, specified resources, and requirements.*	4.29
Project portfolio	*The compilation of the projects which indicates the objectives of the organization*	4.30
Qualification	*The demonstrative process to indicate an entity and its capability of satisfying with specified requirements*	4.31
Qualification requirement	*group of conditions or criteria that have to be satisfied with defined specifications and desired usage environment*	4.32
Qualification testing	*testing phase to determine whether the software is appropriate for target environment and also its desired specifications, this is conducted by the developer*	4.33
Quality Assurance	*demonstrative process to provide adequate satisfaction of fulfilling requirements after the implementation of all planned activities*	4.34
Release	*particular version of an entity to be used in any purpose*	4.35
Request for proposal tender	*Document that is used to announce potential competitor to acquire proposed software system, product or service*	4.36
Resource	*A type asset that is used in a specific intention during the particular process*	4.37
Retirement	*the end of the active support of particular organizations (operation, maintenance), partial or total replacement of the software product or system is expected*	4.38
Security	*Insurance of data from unauthorized persons*	4.39

	or system and their intention to read or modify them	
Service	*Product related activities and their performance*	4.40
Software item	*Single unit or collection of codes (source, object, control, etc.)*	4.41
Software product	*Set of interrelated computer programs and related documentation*	4.42
Software unit	*Separated and compilable unit of program*	4.43
Stage	*Specific period that is located in the life cycle of an entity according to its specification*	4.44
Stakeholder	*An individual or organization who has rights to share, claim or interest in a system and its related attributes.*	4.45
Statement of work	*the document that is used by the acquirer to determine the tasks to be performed according to the contract*	4.46
Supplier	*An individual or an organization that enter into a contract with the acquirer for the purpose of supply process*	4.47
System	*The combination of interrelated elements that are specifically designed to achieve particular objectives*	4.48
System element	*A member of group of elements that constructs the system*	4.49
Task	*permissible action that is intended to contribute particular outcomes of the processes*	4.50
Test coverage	*The test cases that tests the requirements for the software system*	4.51
Testability	*The attainable test can be developed and*	4.52

	used to determine whether a requirement is satisfied	
User	*An individual or group who takes aid from the software system*	4.53
Validation	*Confirmation of objectives of an application with the requirements of intended usage*	4.54
Verification	*Confirmation about the specified requirements fulfillment*	4.55
Version	*Uniquely identified sample of an particular item*	4.56

Table 33 Table of the Terms and Definitions

2.10 **Tailoring of ISO/IEC 12207**

Although ISO/IEC 12207 software lifecycle standard shows a set of agreements of experts on some procedures for software development, there is no one-size-fits-all type of selection and tailoring of processes. However, the responsibilities of both acquirer and supplier are considerably important in the tailoring process [12]. According to acquisition process definition in standard, the accomplishment of supplier dependent upon accurate explanation of acquirer's intention in terms of system requirements and with aspect to software development process [37]. There is a common point of view to the quality of any software product mostly depends on nature of the software development process [31]. In fact, all successful companies in any size should follow well-defined activities and tasks. Another factor that increases the success of further development and reduces the current project's problems is receiving adequate feedback from both user and prior projects. To accomplish these factors and to improve the development process, companies should invest on software process improvement activities, however a number of companies still rely on the success of the employment of ad-hoc processes, which relies on individual's skills. Such processes are very difficult to reuse. In addition, they may have an adverse effect on the quality, maintainability, and cost of a product [37]. Moreover, too many reasons can be stated for a failure of process because of the absence of enough knowledge, lack of customer feedback and market related problems. To overcome risks, problems, and inadequate development process and to improve the quality, there is a strong need to follow some standardized methods. Due to these reasons ISO/IEC 12207 is one of the recent and valid to use standard that includes the steps that should be followed by contractors. Tailoring process shows itself here more clearly at the acquisition process and acquirer has a chance to tailor ISO/IEC 12207 [58, 12]. This process is totally good for providing guidance for quality related activities by mitigating risks and ultimately crucial to project success. Under tailoring circumstances such as novelty, size, budget, risks, technology, time etc. shall be inspected [57, 59].

2.11 Background of Related Works

2.11.1 Introduction

This chapter covers the definitions of games, gamification, and serious games. It starts with giving example definitions of games from the literature. It continues with background of gamification and then clarifies the concept of serious games with related examples of software engineering, health care and automotive industry to strengthen the idea behind the study.

2.11.2 Definition of Games

To clarify and bring to light of the definitions and the general notion of the games, there are various view points and experiences that are referable in the literature [11, 86]. Due to arguments while trying to define the essential terminology for the definition of game it can be sophisticated to explain in clear words [30]. Beside these controversies Suits [16] explains game as a kind of volunteer attempt to overcome obstacles, so there should be some voluntarily actions to overcome specified difficulties with specified or hidden goals are necessary.

At the end of the game which can be stated as end-state of the game can be in various types, but in general the expectation from a game is presenting clear statements for winner, because of this Salen [86] defines games via given rules or conflicts players should be engaged to accomplish a quantifiable outcome which is the final words in numerical expressions. Figure 5 presents the common Game Elements. These elements can be stated as dynamics, mechanics and components. First one is dynamics which generally integrate with whole gamification application and they related with the aspects of entire game play duration. They distinctly depict the big picture of the game play. According to them user realizes the main theme and progress of the application [39]. Second one is mechanics which can be seen as main process that creates activities, movements and operations for user to perform specific effort. In gamification context they generally describe active mechanisms during game play. Because of the main goal of them they intently related with user actions and their decisions [39]. The last mechanism components can be thought as definitive expressions of dynamics and mechanics. They are generally representing the instantiations of mechanics and dynamics while in game play. In addition to this during game play components can be located and

sprinkled in everywhere in order to engage users. Moreover, they are actively used to collect user feedback while they are performing game related actions and decisions [39].

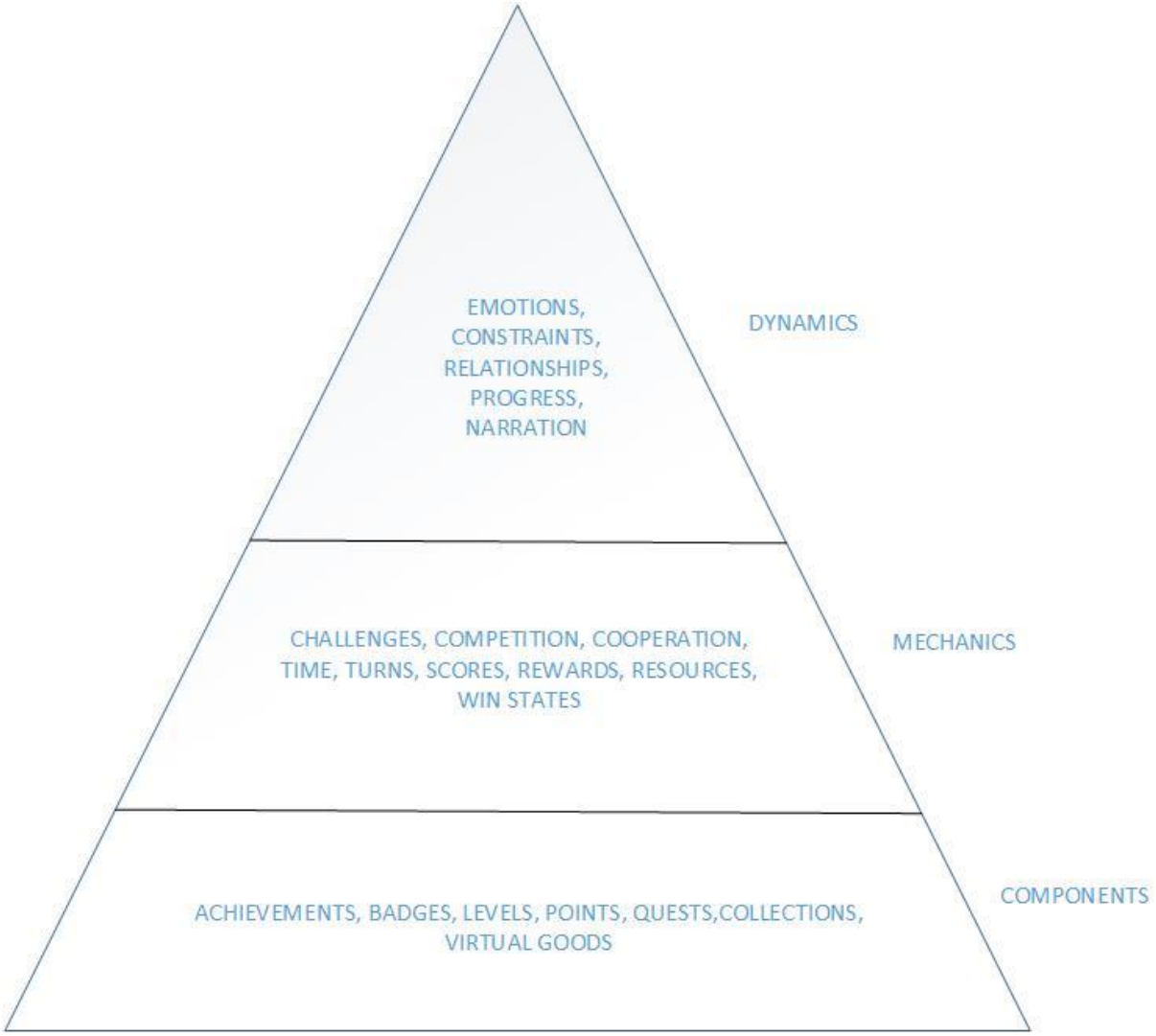

Figure 5 The Diagram of common game elements [39]

According to Uskov [106] there are several types and ways to use games in areas like education, learning, simulation of objects, and states that classification of these types (See figure 5) can be presented like below;

Entertainment education: it has the main goal of making the learning process more enjoyable with or without media-based. It is also suitable for classroom environment [106].

Game-based learning: it has the goal of the utilization of different types of cards to make learning process easier with educational purposes [106].

Serious games: this type mainly focuses on engaging learners (who can be students, employees, trainers, customers, etc.) to learn or gain skills about particular subjects [10].

Digital game based learning (DGBL): this type mainly focuses on actively use of games (especially computer games) in education. It also includes online learning [32].

Edutainment: this type is the combination of education and entertainment with the focus of being learning, education, and entertaining [106].

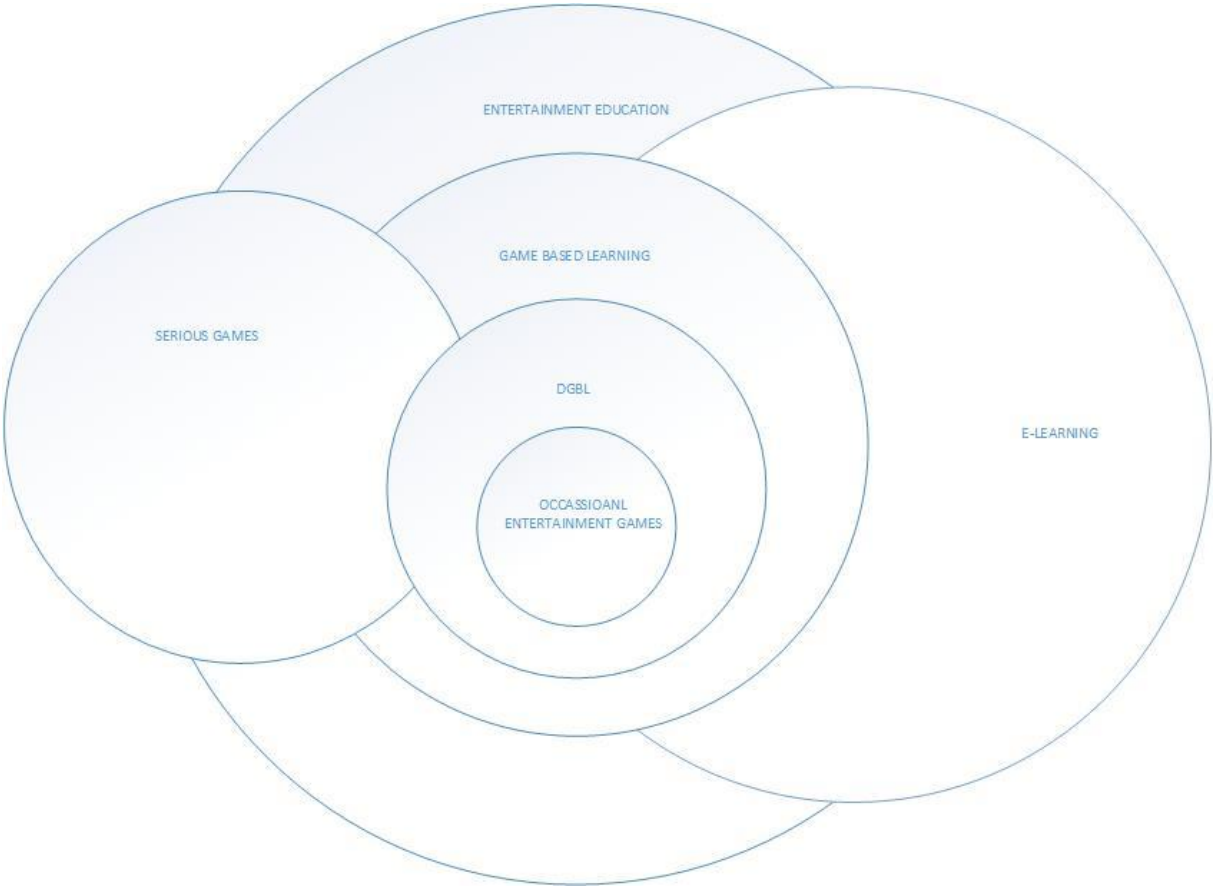

Figure 6 the Relationships between Educational Concepts and Serious Games [22]

2.11.3 Definition of Gamification

Although there are many definitions and thoughts about gamification in the literature it has been defined as "the use of game design elements in non-game contexts" [40, 7]. Gamification gets help and uses the philosophy behind the game design in non-game environments. In addition to these game elements, game mechanics are useful elements of the purpose of inducing certain behaviors of people [82]. While the process of game design with a different concept (Interactive learning approach) it is important to know how gamification takes the concept and become successful with the help of games, because gamification takes main features of games such as fun elements that they make players engaged to the game [113].

It has been used in many different domains especially in education and training [59]. By this way gamification's first and the most important feature reveals itself clearer, which is being useful for the motivation and engagement in particular tasks where people are responsible with certain jobs. It has also been used with the intention of improving performances of employees in the development of their daily work [57]. These kinds of improvements based on engagement and players' game based strategies and thinking. To reveal them it is important to use combination of instructional and cooperative design elements [63].

Many obvious and various kind of features and branches about gamification can be stated but there are several examples of the usage of gamification in software engineering. Because of the nature of "human-intensive" structure of the software processes, gamification has potentially beneficial features that need attention [66]. As a matter of fact that gamification can be discussed as a field, where managers, developers, and engineers get benefit from and improve their daily engagement and learning processes. Deterding et al. [40] states that many commercial domains have been using gamification elements for years. Additionally, there are some actively used commercial tools as examples of the gamification in software engineering field such as "*Visual Studio Achievements*" [70]. It is a kind of gamification plug-in into Visual Studio environment to take up challenges and to present the expertise levels of individuals. ScrumKnowsy [91] is an interactive gamification example that helps individuals also organizations to improve their performance which is labeled as "Scrum point" with teams in roles. It covers agile development attributes while giving individuals chance to compare decisions and choices of experts. Additionally, it is available via multi-player game that gives

chance to teams to work with together in their development areas [90]. MasterBranch [68] is a kind of gamification application which allows professionals to get score with their code and to show their reputation in the community.

Furthermore, there are various proposals that focus on training and teaching software engineering has been published in recent years [82]. According to literature review, which has been conducted by Xu [110] today's current state of gamification mainly focuses on the most obvious and usable game mechanics such as levels, points, leader boards, and badges. As an additional conclusion for the literature review many commercial tools have already included attributes about learning, teaching, and engaging factors. Table 34 shows gamification examples from Industry.

Gamification	Definition	Gamification Use	Game Elements
U.S. Army America's Army [8]	Both training and recruiting gamification tool to attract new recruits for U.S. forces [8]	Recruitment	Badges, Levels, Challenges, Achievements
Jillian Michaels [60]	With the help of suitable fitness challenges, it makes easier to track exercise program [60].	Motivation and Goal Tracking	
Samsung Nation [87]	Socially generated environment for users to share and involve discussion about user experiences with achievements and	Social Loyalty and Customer Engagement	Badges, Feedback, Stats, Challenges

	badges [87].		
Chore Wars [29]	Gamification tool for kids and adults to improve their motivation towards household and office related ordinary jobs in a defined program via avatars, challenges, etc. [29]	Competition, Employee Motivation	Avatars, Challenges, Levels, Stats
Nike+ [76]	Gamification tool with device enhancement to measure and store exercise related data [76]	Exercise Monitoring	Feedback, Badges, Challenges
FourSquare [43]	Kind of service to provide and recommend information about locations where users get experience [43].	Discovery Recommendations	Achievements, Challenges, Badges
Stackoverflow [96]	Website especially for coders to involve discussions about coding problems [96].	Communication with coding enthusiasts	Avatar, Badges, Levels

Table 34 Examples of Gamification Applications from Industry

2.11.4 Concept of Serious Games

A serious game is an interactive approach designed for a purpose other than pure entertainment [65]. A goal of a serious game is usually improving an educational aspect where participants certainly attend such activities with such an expectation. These interactive applications are widely preferred in training and education for medical and military personnel. Recently, serious games become more popular and therefore they are now found in any size, complexity and platform similar to casual games. In general, serious games, consists rules that restricts players behaviors and actions during game play because of being or concerned with more specific themes or subjects rather than commercial games. Due to its main purpose which is educating participants and increasing their awareness, serious games can be used in many different disciplines [107]. In particular, the education aspects of these interactive applications are heavily depending on the notion of play, which is an important factor for individuals' development and learning [1]. In addition, serious games are kind of simulations of real-world events or processes that are addressed to comprise particular problems. Therefore, they can be considered as serious activities such as exploring, training or advertising [51, 111]. However, they still can be entertaining, if their main purpose covers game elements well. Substantially, games have many attributes which have been seen in the case of different examples. For instance, serious games allow participants to experience different learning tasks by using the elements of fun. Another example of attributes is stating how actions affect the context. Players can create artifacts or complete tasks within in the orders of a serious game serves and without the effects of real world problems and stress. This can be interpreted to resembling sand box type games. Moreover, serious games promote an active participation while accomplishing its main goal. In fact, games are powerful tools, because they have the ability to change human behavior [69]. The ambition and the direction of this power can be useful with the help of incorporation with principles of learning [45]. This incorporation may encourage players' decision making and viewpoints about complicated positions in game play of a serious game and at the end of game play session they clearly distinguish achievements and their success in particular subjects [46]. Furthermore, games can help users with repetitive actions while learning certain subjects. Because particular tasks and clearly stated objectives of serious games make player easier to follow certain pathways and play their role for a set of planned behaviors. Such planned behaviors can be easily linked to the learning process where gaming may assist and ultimately create a user-oriented learning experience [115, 50].

There are numerous works which are related about serious games and applications. However in the literature there are only several serious games which are related to software project management. These are; Problems and Programmers [13], SIMSOFT [28], SimSE [74,75], SESAM [41], DELIVER [108], ProDec [25].

Problems and Programmers is a kind of serious game, which covers educational elements with card components [13]. The primary ambition while designing the game is being as a teaching tool to help, improve, and assist the student's understandings towards software engineering processes. The main goal is teaching the software processes and related issues that are generally encountered during the software development, which may not be adequately implied by any kind of software engineering lectures, homework, and projects. The most obvious distinctive attribute of Problems and Programmers among the other computer-based digital variants of serious game genre that it is developed as a non-digital card game where players are trying to finish a particular software project. From the developers' view using the physical elements are easier and quicker way to highlight the main attributes and characteristics of the game rather than programming. From the players' view game state and the player actions are more visible to monitor and manage the further decisions.

With the help of its competitive design players have a chance to learn and get benefit from each other. Specifically these attributes strengthen the main goal of the game. In addition to this the game has several objectives which reveals themselves with also these attributes, such objectives can be stated like (i) advocating the appropriate use of software engineering fundamentals, for instance, taking risks during decision making phase, skipping the design elements and focusing on development phase, etc. (ii) illustrating the specific actions and their occurrence in real software development environment for example visualizing ongoing problems, bugs, and changes of requirements as a non-linearity of the business, (iii) providing clear and instant feedback with concerns and decisions while game play, (iv) providing relatively easy and rapid game style without handling too many complexity of the conceptual elements, (v) encouraging interactions between players, because of the target user profile is students it is important to use collaborative learning methodologies while game play [23]. According to Baker [14] the important point is observing different players and their strategies while game play makes players not to just evaluate their decisions but also comparing components' strategies that are followed to reach the goal. As a consequence of this players learn from each other.

Mainly, in Problems and Programmers players have a chance to follow different models to manage their projects and figure 6 presents an example game play which covers the steps of waterfall model [14].

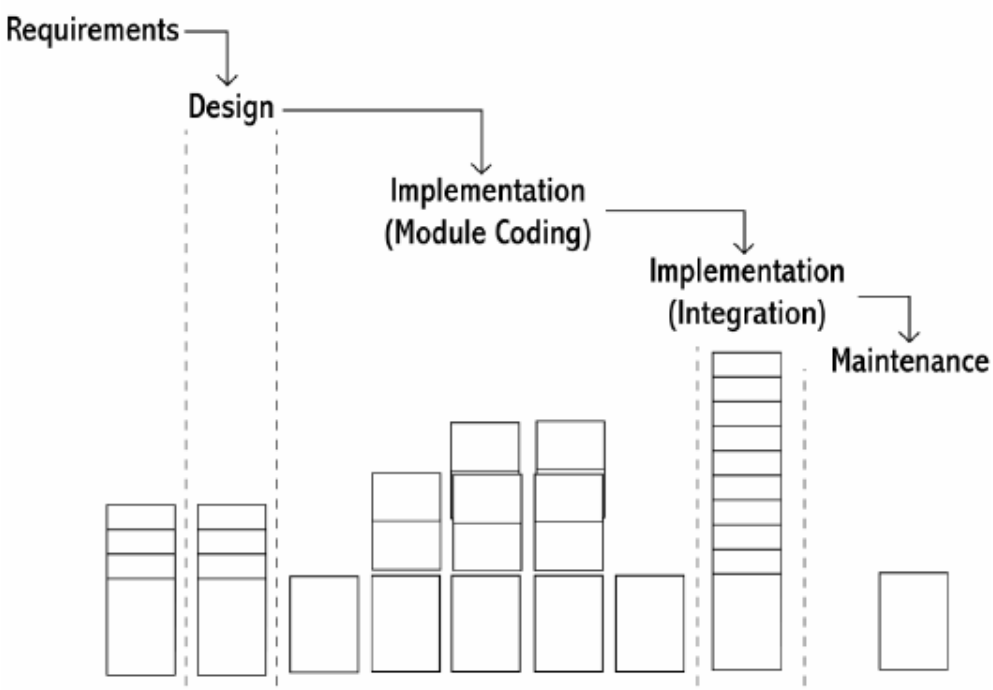

Figure 7 Example Gameplay set from Problems and Programmers [13]

By defining different card types such as program cards, problem cards, bug cards, developer cards, concept cards, etc. the functionality of the game play is improved and become more solid to the player to understand the general idea and the philosophy behind the decisions and actions during management process of a software project.

With highly visual elements and providing collaborative learning fundamentals problems and programmers is an engaging serious game to teach the basics of typical software engineering attributes.

SimSE [75] is a type of serious game that is playable with a single player. It is designed to serve an interactive environment which means there are graphical features and the game includes visual elements. The player takes on the role of a software project manager. By being a project manager, a participant is able to hire and fire employees, adjust to schedule, maintain the budget and they are responsible for further decisions about the project, especially assigning developers to particular tasks, monitoring their current work and mood. Furthermore, monitoring the ongoing processes, and the progress of developers are important to make decisions about project's plan. By this way players can give directions to design, develop, and test phases under the limitation of budget. All of these make players gain experience on managing and planning about software project development. This kind of virtual experience helps the players to understand processes easier and faster than lectures. Additionally, during lectures on computer science and programming students are generally exposed to theoretical issues and concepts of software engineering basics. The limitations about syllabus, time and many more factors are not sufficient to imply and teach exact or more concrete examples about the real environment and the business. This is the obvious and the most significant reason for creating such an educational serious game which covers the characteristics of project development.

The main goal of SimSE is fulfilling this absence via 2D graphical office environment where all the processes are maintained. This scene includes many office staff to visualize the environment with computer, employees, desks, requirements, artifacts, and plans. This approach makes everything clearer to players while making their actions. Moreover, with the support and the availability of the selection options of suitable moves players can continue to manage the project step by step.

SimSE has beneficial features for players who are the students mostly, but the most obvious important feature of the game is serving a model builder, which is a distinctive feature among other serious game variants [75]. The model builder allows user to create tools, projects, customers, employees, codes, artifacts, etc. without any specific programming inconvenience. The model builder helps and supports the customization of the duration of software processes like the real-world examples about software projects. In addition to this the game supports many predefined example of models for instructors who are able to use and adopt them according to their own desires and purposes to explain concepts of software engineering. If they want to specifically discourse on some unique feature or attribute of a concept, instructors can write new models from scratch [75].

With specific features of SimSE has great advantages on teaching and serving many contents about software engineering that are useful and beneficial on the learning process of students.

ProDec [25] (Project Decision) is a kind of simulation-based serious game. The ambition behind Calderon et al. is the intention to train and assess students in a software project management [25]. The main goal of ProDec is creating an awareness of planning, controlling, and managing of software project. Under the limitations like budget and time the game play is continued. Every player has a dedicated amount of budget and allocated time to finish the software project. The game play implements unplanned events for player to deal with these kinds of obstacles during game play.

Another intention is being a collaborative game. According to Calderon et al. the game should be played by teams of players. Group of players collaboratively involve game play to plan, create, develop, and design the elements of the project. This actually means that players are not individual and they do not compete with each other Furthermore, they have to be group to work with collaboratively and to win the game, although the game can be played by individuals to learn and get experience on planning and engaging the software project development.

The main beneficial characteristic attribute of ProDec is containing a report mechanism [25]. After game play, ProDec serves a detailed report including every decision the players have made. This helps reveling the assessment criteria that is indicated by the trainer of the game. Beside the concept which covers the significant processes the game has an extensive architecture. With the help of java and MySQL [25], the game consist of an administrator (ProDecAdmin) to maintain information and create different scenarios to the players to compete, and also it consist the application (ProDecGame) which is the source of activities and acts like a base. Figure 8 shows the architecture of ProDec.

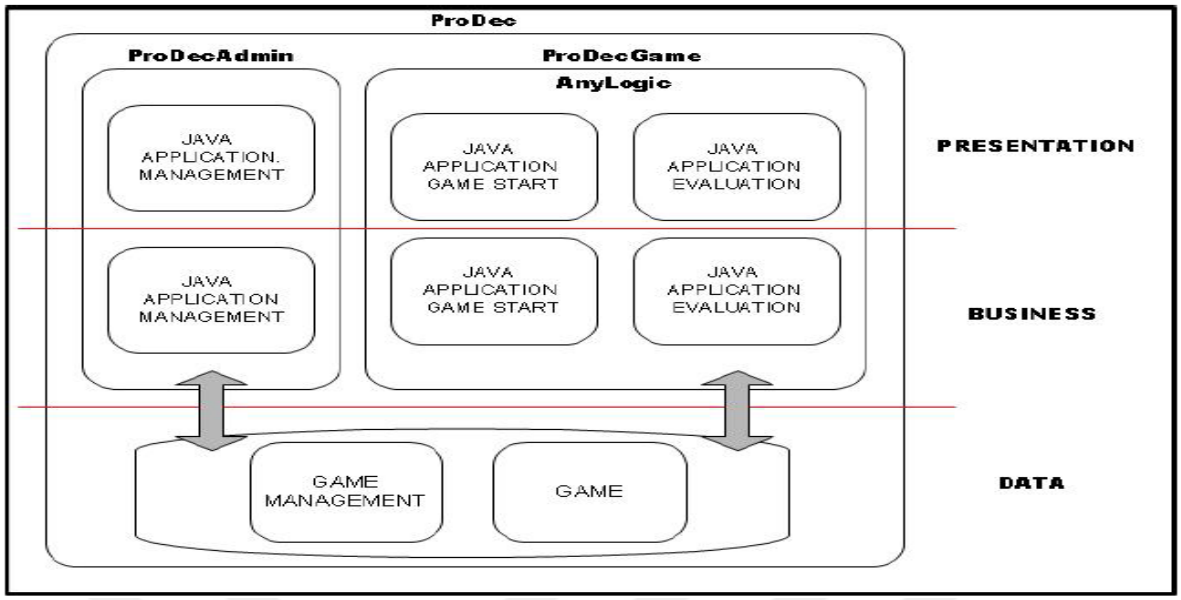

Figure 8 The Architecture of ProDec

ProDec serves and includes many attributes such as project planning, task identification, project control, project monitoring, and risk management with unpredicted events. Team management with allocation of tasks, etc. the gameplay's similarity with a simulation helps to generate dynamic / predefined models to cope with. Additionally, detailed reporting mechanism becomes a distinctive feature among others while maintaining assessment of players and their decisions. The game also consists of gamification elements such as the interface, Hall of Fame with reporting mechanism and badges.

SimSoft [28] is a type of serious game which consists of two game boards. For the game there is a printed board and a digital board. Besides just including physical components like Problems and Programmers [13], or SimSE [75] and ProDec [25] that they have digital components, SimSoft is the hybrid of all of them.

The A0 sized printed board is used to indicate players' move and current state of the game. Board shows flow of game and players can discuss state of the project and decide their moves and strategies. Additionally, poker chips are used to represent the budget that can be used to purchase staff for the project. The players are expected to make a combination of meaningful choices and manage spends via available budget. To be a companion to the printed board in the game there is a Java based digital board to indicate the current and historical state of the project with the mentions like reports and messages. The players are grouped into set of teams

to manage the software development from start-up to the final delivery [28]. Furthermore, they have the opportunity to observe the difficulties of resource management in every aspect.

SimSoft [28] is an example of serious game which covers resource management issues. Specifically, the game focuses on human resource management. The game gives an emphasis on the importance of skills and abilities of the personnel with their decisions and strategies. Lastly, DELIVER [108] is another type of serious game which consists of a printed board. It helps students to develop controlling projects performances. Its main ambition is totally motivate students in their learning progression.

All of these serious games examples are related with software engineering and contains mainly software project development. All of them combine educative and engaging elements that are obviously detectable while in game-play. However, in the literature there are many more examples of serious games that are both educative and contain unique characteristics which can be held up as an example. The following games covers generally health care related subjects but also contains social elements with interactive designs. Such games can be stated like Captain Novolin [26], Packy et Marlon [77], Remission [83], Pamoja Mtaani [80], Triage Trainer [102].

Captain Novolin is a kind of educational game on SNES (Super Nintendo Entertainment System) platform. It was released at 1992 an the main goal of the game is teaching players about diabetes [26]. The game's character Captain Novolin who is a diabetic hero directs the gameplay with his defined mission, which is saving the major of "Pineville" from the alien "Blubberman" [26]. The game integrates the mission with making situations more obvious and immediate such as the major of Pineville is also a diabetic person and there is only 48 hours remaining without insulin. Moreover at the starting of each level doctors help and instruct Captain Novolin what he should eat during gameplay. While controlling the eating habit in the game player must check his blood glucose level and if necessary inject insulin. While playing the Captain Novolin the game clearly instructs the player via doctors and become an assistant for the targeting players who are mostly children [27].

Another serious game that is about informing players about diabetics is Packy et Marlon. It was released at 1994 and the game can be played on Super Nintendo Entertainment System [77]. The game covers the subject with minimal graphics and tries to teach diabetic issues. A player plays with two elephants that are trying to recover stolen food while checking glucose level and shooting the enemy with water and peanuts. Gameplay covers the adventures of two

characters, while they are in pursuit player must control their eating habit healthy food, and regularly checking the blood glucose level and trying to find proper food to fight. The target audience is again children in general [78].

Re-mission is a third person shooter game in which player is put into a kind of cancer fighter Nano machine that fights with tumor cells [83]. Re-mission is a free game in serious game genre that can be playable on Windows and Mac. It was released at 2006 by HopeLab with a nonprofit ambition which is the game is free for cancer fighters and world's awareness of diseases. The main goal of the game is fighting with tumor cells with a Nano machine. Medical researchers generated this idea. By this way the game provides many essential information about various types of cancer and the medicine types that is used during treatment. The target players are kind and young people who have disease. The game has a potential for being addicted because of blasting tumor cells and enjoyable gameplay engaged players rapidly [83]. Like in real life the game serves options like chemo, antibiotics, radiation, etc. while fighting and shooting with cancer cells. Whether or not to have a disease the game covers much useful information with enjoyable environment. Additionally there is a follow up game that is Re-Mission 2 by same publisher. This time game covers improving psychological outcomes that are related with cancer and its cure [83].

Pamoja Mtaani (Together in the hood) is a multi-player serious game from the partnership between Warner Brothers Entertainment and U.S. President's Emergency Plan for AIDS [79]. It was released at 2008 with the ambition of instructing Kenyan youth about HIV/AIDS. The game covers contemporary East African context and serves an interactive play to instruct HIV prevention. While designing the game Kenyan writes, musicians and engineers were involved to demonstrate context to the youth more easily. Mainly, it assesses the changes and exposure in attitudes and common beliefs of people towards sexual concepts. Moreover it uses questionnaires while gameplay to make the evaluation robust. At the same time the game captures players' selections and activities to be demonstrative at the end [80].

Triage trainer is a type of serious game which tries to develop appropriate decision making in life saving [102]. It was release at 2008 by Trusim. The gameplay is a combination scene that covers a view of an explosion with broken glass, distorted metal bars and wood and also suffering civilians who got several different injuries waiting for being rescued [33]. While gameplay players have to follow some set of protocols to continue making right decisions to

react triage situations. Players move one point to another point for examining victims who are randomly generated. All victims have different symptoms that range from nose bleeding to exposed organs. The game's main expectation from the player is checking and conducting every essential step of life saving essentials like talking with patient, checking patient's breathe rate, and pulse rate. After this procedural walkthrough players have to identify the priority of the victim's injury. According to that categorization of priorities the game scores for players. By this way game provides a virtual training for a reference point to students [33].

Both software engineering and health care examples of serious games imply the main characteristics especially gamification elements and being educative to the players who are mostly the participants from deferent domains. Because of its purpose which is proposing beyond than pure entertainment, serious games can be used in different purposes like teaching a particular topic, encouraging education or a particular training, and promoting behavior change [92]. At this point the definition of gamification reveals itself clearer because gamification is also a process which covers game elements and game mechanics and their usage in outside of the game industry. Generally rewarding method is used for the continuation of games' or applications' survival in the market. However, the important point should be the process of improving or at least getting user engagement and also his/her participation to the game or any kind of application which originates from non-gaming environment. Mathhews [92] concerns about rewarding the good behaviors, "It's always the bad school kids who get the stars when they behave for one. The good kids never get anything." This situation can form the philosophy behind the gamification while using badges and points. Moreover, Fleetwood [84] also has statements about rewarding mechanism, which are used in wrong ways. The gamification comes in different applications and in variety of subjects. To make more generalize all the human activities can be gamified. The wrong and misleading definition for gamification is not about "pontification". The games are not equal to the points [84]. To make these concerns clearer just using point for rewarding mechanism do not transform and application into a gamified application or a serious game. The epicentrum is the purpose or whether a game play consist purpose. If a serious game successfully maintains a purpose it will effectively touch to the audience. Just using badges, rewards, points, etc. alone in the game play do not satisfy and guarantee the effectiveness of the game so it cannot direct any kind of behavior change. Moreover, the games do not have power to the change of behaviors. There is a need for using them in right purpose and context as a tool to shape the

behaviors. To gain this potential power games should be iterated, well researched and they should be suitable to the audience [91].

Overall, these serious games are the examples of dedicated goals in properly defined atmosphere. With the help of these games' main attribute which is exposing complex or important messages with a more captive and interactive ways, users have a chance to remind main concept and specifications. In conformance with this ambition we conducted our study on basics of ISO/IEC 12207 and related ambience.

2.12 **Summary**

This chapter reviews the literature and presented an essential background of the study. The main parts of this chapter includes the details about the fundamentals of software engineering, software project development, software life cycle and process models, ISO/IEC 12207 processes, general information about background related work, definition of games and game mechanics, gamification, and serious games concept with its supportive examples are presented. Especially, the more emphasis is given to the concept of the standard and the serious games examples in order to clarify the main structure of the study. This chapter can be a used in order to find out reference information about for serious game concept in the areas like health care, SE, and educational concept. In addition to this it contains extensive information about processes of the standard and key terminology for the usage and understanding of it. Every selected process and its related activities are explained. Moreover, for the general purposes general structure of the standard are given with reference numbers for further reading from the official documentation.

Chapter 3

3. Methodology

3.1 Introduction

This section explains the methodology behind the study. It starts with the descriptions of qualitative, quantitative, and mixed methodologies. After then, it explains the reasons of selecting quantitative methodology and attributes of the study. Following section describes the game model. Firstly, the study starts with the literature review about serious games that are related with software engineering and health care. In addition to this it includes proposing a serious game based learning framework. The proposed serious game's main goal is to create an awareness and knowledge about ISO/IEC 12207 software life cycle processes and SE basics.

3.2 Research Techniques

Generally, research methods or techniques can be divided into two main domains, i.e. quantitative and qualitative methods [71]. According to Denzin [38] qualitative researches are carried out with some traditional ways such as typing important notes, interviews, defining and marking cards, sorting and shuffling them. Additionally, with the findings of Tesch et al. [99, 100], qualitative research involves asking typical questions to the participants and then according to their answers the process of observation and analysis are conducted. Therefore it is necessary to understand these observations because it contains collecting data from experiences and feelings of individuals directly [97].

Besides qualitative research techniques, the other set of research methodologies can be categorized as quantitative research. Mujis [71] defines that this research type involves collection and analysis of numerical data via mathematical formulas and statistical tools. Relying on this analyze period results are becoming relative about the desired subjects that are waiting for the judgment. Moreover, the whole process is about quantifying the relationships between the variables, for instance, time, performance, etc. on sample of subjects such as

humans, animals, etc. [52]. To accomplish the research which is related to this study questionnaire and quiz have chosen. Furthermore, in order to complete the quantitative aspect of the study suitable analysis were pursued.

Another research technique can be stated as mixed method research [36]. According to Johnson et al. [61] mixed research is a kind of combination that consists of both qualitative and quantitative research methodologies. Additionally, mixed research has advantages like having attributes from qualitative and quantitative methods by consisting of both numerical and non-numerical data for the analysis [98].

In order to maintain the research which is related to this study mixed research technique has been chosen. Because of collecting numerical data and making interpretations from it, related analysis methods were followed. In addition to this interviews and collecting feedback requires additional analysis. Therefore, all of these procedures required mixed research methodology and we have chosen to get aid from the attributes of this methodology.

At the beginning of development phase of the serious game environment we have collected ideas from different perspectives in order to develop adequate tool. These ideas and opinions constructed the backbone of the serious game environment. After accomplishing the development phase of the game we have employed two learning sessions in pursuance of evaluating the efficiency of our tool. At the end of each session game play scale and quiz questions were subjected to the participants. By this way we have collected numerical data from participants, who have attended our sessions. This part contains quantitative data and related analysis method. At the end of the evaluation of quantitative data we have conducted interviews with regard to get feedback about game via verbal way. This part and the very beginning of the study contain qualitative data and related analysis method. Consequently, mixed research technique was conducted in this study.

3.3 **Procedure of Methodology**

Firstly, the study starts with the literature review about serious games that are related with software engineering. The main goal was collecting necessary and required information from various examples in order to construct a suitable framework where the serious game environment was developed. Secondly, prototype of this virtual office environment was created via Unity3D. In light of the feedback from experts from academia and students, the proposed serious game was designed and developed. The proposed serious game's name was titled as "*Floors*" to symbolize the different floors of a virtual office environment. By this way, various processes of the standard were discussed. Finally, to analyze the results participants who are mostly the students of computer engineering and computer science are accepted according to groups such as paper-based learning session and virtual learning session.

To maintain this study, a virtual office environment which contains 9 floors to express different processes of the standard were created in order to observe participants' experiences while they are in virtual learning session. It creates more positive and attractive experience for participants rather than paper-based traditional learning session. According to this procedure and to conduct the main ambition of the study user experience study is adopted and related evaluation technique is used.

According to the definition and the statements of Bernhaupt [17], there are various set of methods for measuring the experiences in the field of human computer interaction. This period can contain software system, or software product that includes dedicated goals or missions. Various methods such as interviewing, observing behaviors of participants, conducting surveys can be stated. In conformance with these methods in order to conduct this study surveying has chosen via using adopted version of Game Play scale [81] (See Appendix B.1). It is a kind of Likert scaling tool as an outcome of project from College University, London and was developed by Parnell for the usage of assessing and measuring user experience and feedback from sessions. Additionally, it contains distinctive subclasses and scales, which measures different characteristic features of approaches. This kind of information can help developers and researchers to get feedback from participants in order to validation of their approaches evaluation [81]. In pursuance of determining the assessments of results and the differences between surveys of each participant the adopted game play scale

was subjected to the participants in both traditional paper-based learning session and virtual learning session.

In addition to surveying at the end of each session participants were subjected to assessment quiz (See Appendix B.2), which consists combination of questions from ISO [57], IEEE Computer Society Project Management Quizzes [94] and Software Engineering basics from Sommerville [95]. To maintain this part, participants were grouped according to their involvement in the study. For instance, participants who attended in traditional paper-based learning environment were taken as one group and participants who attended in virtual learning environment were taken as another group. Each group answers the quiz at the end of their sessions. Furthermore, the data collected from each group is used to distinguish and analyze how they perform in quiz according to their different experience environments.

Both the questionnaire and the quiz were reviewed by three academics that are competent in computer science and software engineering for many years. According to the researcher's presentation and the examinations of one associate professor and two assistant professors about Game play scale and the quiz, they approved the suitability of the study with the defined methodology procedure.

With descriptive steps the methodology and the research design was taken place like the statements below.

1. This study starts with reviewing serious games in various areas such as software engineering, management .etc.
2. After completing the review of literature a prototype game was designed which contains the processes of the standard.
3. According to the feedbacks about prototype game the development of proposed serious game was started via using Unity3D.
4. In this study data collection duration starts with introducing researcher and his ambition with this study. Additionally, researcher explains the main features of Floors and its goal. After necessary information is given to participants traditional paper-based learning session takes place with distribution of official ISO/IEC 12207 Software Life Cycle Processes [58]. They are directed to follow definitions of key terminology and selected processes in order to accomplish learning session with the basics of the standard. This session takes approximately one hour. After participants covers the official standard’s terminology and its processes they subjected to the

adopted version of Game play scale questionnaire [81] in order to find out their experience feedback with the paper-based learning session. Then they were given the quiz to assess their success level based on their knowledge from this session.

5. In computerized part of this study participants involved the game play duration of Floors which is a serious game to teach basics of ISO/IEC 12207 Software Life Cycle Processes. This game was designed and developed for this study to visualize the real life organizational scenarios with the help of virtual office environment and representative character models for explaining the definitions and activities of the processes of the standard in order to experience more extensive and realistic quests and goals. Participants involved dedicated quests and dialogues while following the processes of the standard in a virtual office environment.
6. At the end of the sessions participants were given the adopted version of the game play scale to get their user experience feedback, and then they were subjected to the quiz to assess their success based on their knowledge which was obtained from virtual learning environment. By this way, participants who play Floors did not use pen and paper to learn the basics of the standard instead they actively involved in processes with dedicated quests and dialogues to accomplish the learning journey. Lastly, analysis of the results started. To clearly determine the results of this study the t-test for two independent samples were used.

The defined steps of this procedure were repeated for every participant. Figure 9 presents the diagram for the defined methodology.

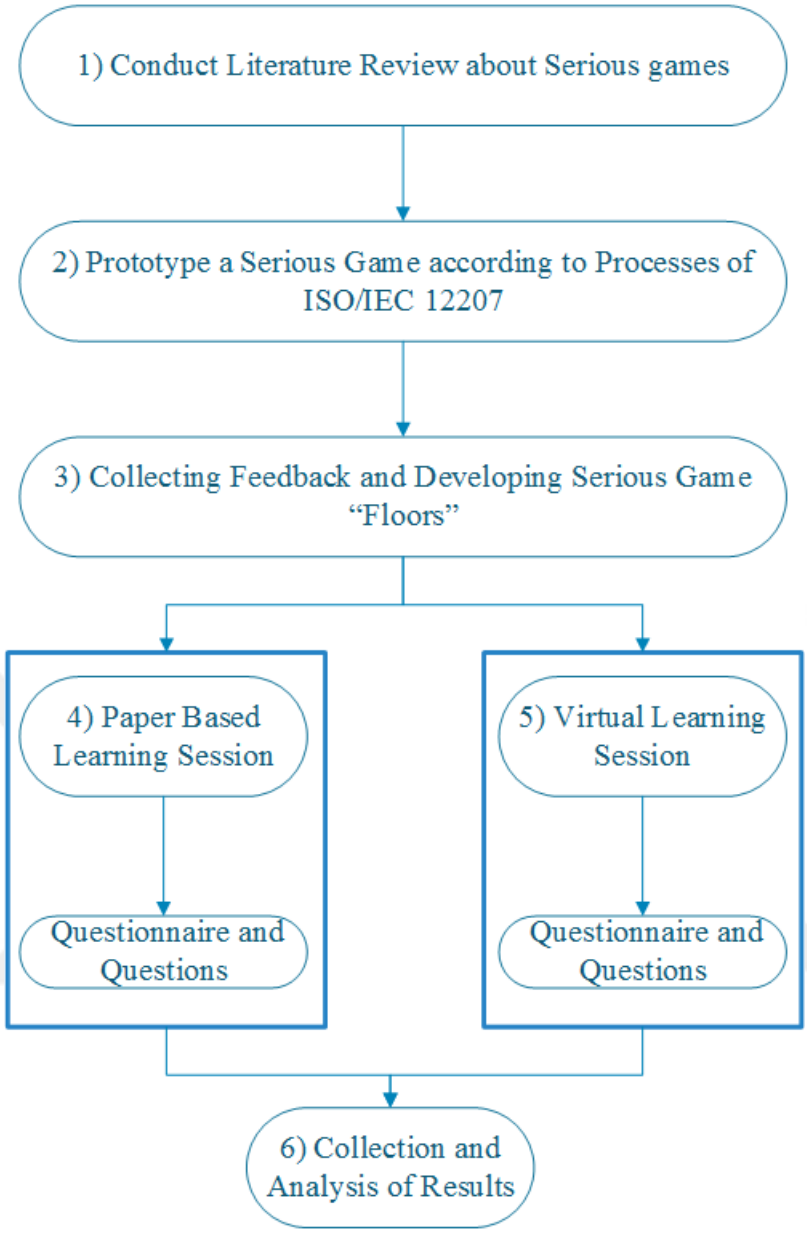

Figure 9 Diagram for the defined methodology

Consequently, entire procedure of the proposed methodology encompasses the mixed research method technique which includes both quantitative and qualitative research methods. In order to validate the effectiveness of the research independent sample t-test were used while the analysis phase of the study. Following chapter will introduce this methodology via visual parts and design elements of the proposed game in order to clarify the main goal of the study.

3.4 Participants

This research was conducted with the students mostly from University of Turkish Aeronautical Association between 25.04.2016 and 10.06.2016. Participants of this research were undergraduate freshman students who accomplished the same curriculum. A group of participants (n = 40) was selected for this study. Next, these participants were randomly divided into two groups that are named as paper-based learning session and virtual learning session. In paper based learning session participants were allowed to follow the selected processes of the standard via official documentation of ISO [58]. On the other hand, participants who are in the virtual learning session were allowed to play our serious game for completing the selected processes of the standard. At the end of the both sessions participants were subjected a questionnaire, which was, derived from a set of user experience questions particularly based on game play scale [81] and quiz questions [94].

3.5 Summary

In conclusion, this chapter provides detailed information about defined methodology for this study and the main characteristics of participants. First section started with explaining the useful definitions for research techniques were used for this study. After procedure of methodology section clarified the backbone of the study with descriptive steps and complementary information. At the end of this chapter participants section were introduced in order to present their main characteristics. To sustain the main ambition of the study which is exposing knowledge and teaching the basics of the ISO/IEC 12207 processes the serious game environment were created. This tool has been found usable by students who are studying computer science and software engineering in order to introduce the basics of standard's processes. In this manner, our tool can be used as a supportive element for students to understand software life cycle processes in a visual way.

Chapter 4

4. Design

4.1 Introduction

This section presents the tools and required techniques that are used in order to create the proposed serious game Floors. For the development of whole project Unity3D [105], Mixamo [9], and Blender [19] were used and this chapter explains their main usage. With glorious attributes of Unity3D game development environment lead to the main usage among others because of the competency of the researcher and its popularity in game development industry. Very similar to Unity3D, Blender offers many outstanding features while modeling certain objects in Floors with the help of its resilient infrastructure and being free of charge fostered the visual development of Floors. Lastly, Mixamo were used with its powerful character creation tools and optimization units for 3D environments. Via Mixamo very detailed NPCs were created. With the help of these development environments the study gets sophisticated features to present users more detailed 3D virtual office environment in order to comply main attributes of Floors.

4.2 Design tools of Floors

Floors is blended by various programming techniques and tools throughout the development phase with Unity3D, Blender and Mixamo in order to reveal visual elements of its main scenario.

Unity3D [105] is very popular game development environment which was developed by Unity Technologies [35]. This development environment contains many supportive predefined objects and modules in order to help users to focus on their specified design. Additionally, these objects and modules generally free to use without any limitations. Moreover, various kind of simulations which consist these kinds of intense computations and large sized graphical elements can also be developed with this environment. Additionally, it serves an environment to users to create both 3D and 2D games with comprehensive predefined selections of attributes and assets. The main reason for the selection of Unity3D for the development of Floors is being free and broad asset store which helps any kind of

developer for finding related assets for their particular needs. Figure 10 represents a scene from Floors which is created with Unity3D.

Figure 10 Screenshot of sample floor and animated NPCs from Floors

Besides enabling user to find out variety of particular visual elements it helps developers to find out correct coding schemes and well defined mapped objects in order to maintain specified objectives that are used in game play sessions. Therefore, "Floors" uses Box collider and isTrigger functions of Unity3D to accomplish main dialogue infrastructure. When the user interacts a dedicated area which is surrounded by box collider dialogue events are triggered by isTrigger function and OnTriggerEnter method and also with the help of GUI (Graphical User Interface) elements of Unity3D textual information, speeches, and tips come in front of the user. Figure 11 show Collider functions in 3D environment. IsTrigger and OnTriggerEnter methods controls if an object enters or exists the dedicated area and then allow programmer to code necessary actions to be performed during the game play [105].

“Floors” consists of Third Person Character Controller asset of Unity3D. This controller helps players to see the virtual world like their vision in real life. By this way they feel more realistic during game play period. In order to create more responsive game environment and to get the attention of the user during dialogue sessions and reading tips MouseLook events are fixed when the user interacts with collider area. By this way users are isolated from

irrelevant game objects and virtual elements while they are participating dialogues or reading tips in a more focused manner.

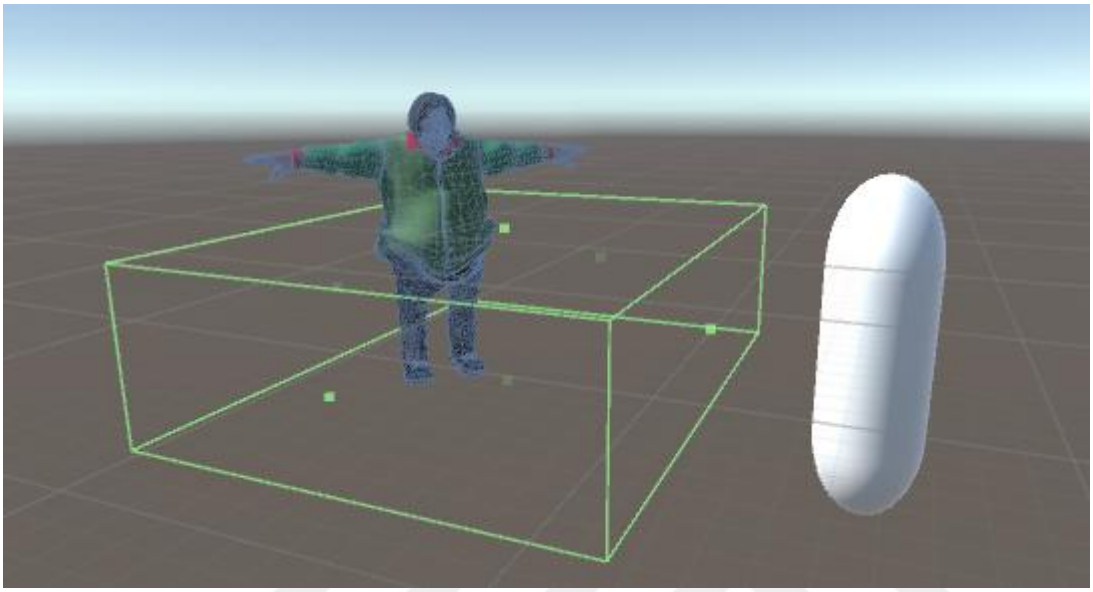

Figure 11 Screenshot of Collider and NPC with an outside game object in Unity

Mixamo Fuse is another program which s generally used to create 3D characters [9]. It is used to create both human and humanoid characters which can be used in game scenes. The main purpose behind using Mixamo Fuse is being close enough to detailed character creation while preserving system resources. Furthermore, there is a need for optimum graphical quality which is represented by polygons needs to be adjusted because of the constraints of video accelerator hardware and the amount of RAM that is allocated to calculations of polygons. Figure 12 shows the details of created character. Mixamo Fuse serves convenient usage of characters with approximately 10.000 polygons, which is moderately enough for general computer hardware and its computation power. With the help of convenient tools of Mixamo character creation duration contains many useful actions. Whole process starts with selecting gender of the model and then finding appropriate body parts, but the most powerful assistance is the customization of the model according to desires of the user or appropriateness of the

main theme of the virtual environment. Figure 13 shows the character creation phases with Mixamo Fuse.

Figure 12 Screenshot of sample detailed character from Floors created via Mixamo

Figure 13 Screenshot of base model creation phases in Mixamo

Blender is an open source 3D modeling environment that consists many tools and add-ons for maintaining the processes of modeling [72]. In this study Blender is used to create basic models of start menu, floor numbers and necessary office stuff, although it is a combination of potent tools for more specialized operations. Through the process of modeling essential stuff for this study basic and well-known techniques and approaches were used. By this way every creation with Blender incorporated to this project's genuineness. At the end of the modeling phase every creation exported as suitable format to be used in Unity3D. This format is known as .fbx (FBX) which is a file format to be used to sustain interchangeable operations of digital creations [19]. Figure 14 shows the start menu design in Blender.

Figure 14 Screenshot of 3D printing and objects which are created via Blender and used in Floors

“Floors” contains 22 NPCs (Non-Playable Character). 14 of these characters are actively used in game play duration. They have specifically defined scripts in order to get interaction with user. Moreover each of them is responsible one of the selected processes of the standard and he/she explains required definitions in conformance with the exact terminology according to the explanations of the standard. These definitions were transformed into dialogues in order to integrate necessary information via more explanatory way throughout the whole game play. By this way user followed a defined path between processes and follow characters and their explanations in pursuance of completing different floors. The official documentation of the standard contains comprehensive definitions because of this while coding these definitions in proper format for the game environment array data structure and nested conditionals were used with C# programming language. Therefore, every NPC has own dedicated class which contains information about its specified process and related activities. Beside these NPCs there are 8 more NPCs, which contains animated gifs in order to express the current status and objective of the floor. These NPCs have no direct interaction with user however they reflect the atmosphere of the specified floor with animations. These animations were adjusted with the help of Mixamo libraries and edited with Unity3D environment to locate them adequately in the game scene.

Consequently, in this study, every creation and asset were connected together inside Unity3D in order to preserve the main theme of Floors. Basically, the main 3D objects of Floors were created with Unity and Blender, animations and characters were created via Mixamo, various office stuff especially PCs, printers, chairs, sofas, desks and lightening equipment were imported from Unity Asset Store. These stuff were free and integration to the game environment did not require any additional cost.

4.3 **Summary**

To sum up, in this chapter, design tools for Floors are described. These tools can be listed as Unity3D, Blender, and Mixamo Fuse. These tools were extensively used for the development phase of Floors. Particularly, while creating the virtual office environment and defining game mechanics Unity3D was very helpful. Additionally, it helps to understand general topics about game developing and game physics with its convenient tools. On the other hand while creating objects Blender was very useful while demonstrating the goods in desired way. Beside Unity3D and Blender, the most extensive job while creating the NPCs and models Mixamo Fuse was used in order to create Floors main components adequately. All of these tools combined main characteristics of Floors with their features.

Chapter 5

5. Implementation

5.1 Introduction

In this chapter according to the definitions from chapter 4 the implementation of the proposed game is described. In conformance with the proposed serious game's main goal which is creating awareness and exposing knowledge about ISO/IEC 12207 software life cycle processes and Software Engineering basics, two different learning sessions were created in order to conduct the research. These sessions are paper based learning session and virtual learning session. Following sections will describe the sessions' main characteristics and the implementation details of proposed serious game. Last section describes the serious game via its virtual elements such as office environment, models, NPCs, animations, etc.

5.2 Main Features of sessions

The defined methodology helps collecting user experience feedbacks from two separated sessions which are virtual learning session and paper based learning session. The names of the sessions are given according to their target participants and their assigned task. In virtual learning session proposed serious game was played in order to find out the how successful the serious game environment was according to the user experiences and exposing the knowledge about the main structure of ISO/IEC 12207 processes. In this session participants were allowed to play the serious game and followed the game scenario about standard's processes with dedicated NPCs and virtually created environment. At the end of the session participants were subjected to survey to get feedback about their experiences and the quiz to find out how successful they were about the concept of the standard with their knowledge from the serious game.

On the other hand, in paper based learning session participants were allowed to follow the selected processes of the standard's documentation which was the same as the serious game (see Figure 15), but this session had no chance to use and get benefit from any digitally created element like the serious game. They used only pen and paper to follow and understand

the processes. At the end of the session same procedure like the virtual learning session was conducted in order to compare the results of sessions. Section 5.3 and 5.4 will give additional detailed information about characteristics of the sessions.

5.3 **Paper-based learning session**

On the side of revealing the learning of basics of the standard there is a need for time and budget to read and cognizance the official standard terminology and descriptions of processes with peculiar language. Moreover, there is no dedicated seminar or lecture in order to expose the definitions of processes of ISO/IEC 12207 for individuals, but in organizational level demands in conformance with specified criterion such as scale, budget, visions and missions companies can get or reach the adaptation period of this standard for their related implementations and project management politics. Unfortunately, finding an appropriate seminar or lecture for this concept is not possible. Therefore, individuals who want to take, learn or maintain this standard have to involve waiting limitless amount of time for learning the processes from the official documentation. In an alternative way of this duration individuals can follow processes with Software Engineering basics from the Internet via documentations from experienced employees and academically units, but again it takes some amount of time and deficiency of real experience to be productive in this field. During this session in conformance with tailoring process [58] participants followed the selected processes which are exactly the same ones with the virtual learning session for preserving the equality.

Consequently, individuals who has to involve paper based learning session can use only pen and paper in order to understand of the basics of the standard and accomplishing main structure within a defined time duration.

5.4 **Virtual Learning Session**

According to Bjork and Holopainen [18], computer games can create more attractive and interesting environments than paper-based versions for today's generations, and maintains the costs and values of marketing strategies.

In conformance with the literature review of this study the importance of software development in serious game environment is getting urgent day by day. Advancements in such serious topics with a serious game are important aspects of software development with the help of developing game engines and game design environments. To expose serious topics to the individuals with using game elements and also to maintain with serious game concepts can be beneficial to accomplish adequately the desires and goals. In this study one of the ambitions is getting rid of pen and paper in order to learn the basics of the ISO/IEC 12207. In virtue of created virtual office environment and modeled characters participants have a chance to experience a journey in an office by virtually. The virtual office environment is 3D environment enriched with animations; sounds,, textures, NPCs, interactive dialogues and quests. They are all desired attributes for accomplishing the dedicated learning goals in a non-conventional way. Moreover, their goal is to improve the learning the processes of ISO/IEC 12207 with their non-static attributes and characteristics of paper-based learning session. During this session participants followed the processes of the standard (see Figure 15) in conformance with tailoring process [58].

All of these dynamic contents during this session require some involvement while game play to express goals and features. In addition to these virtual environment is flexible enough for changing and adding to conduct various different subjects and theme. The simulation of office environment with real world examples and scenarios help to create any imaginary alternative environment. Using convenient tools and techniques for developing appropriate setting and attributes of the virtual office environment costs of many things were reduced while conducting the study in a realistic atmosphere.

5.5 Features of Floors

To follow the main goal of the study, a virtual office environment and an organization is created where software project management and ISO/IEC 12207 process can be realized. According to ISO [56] the standard is available to be used with different life cycle models. However, organization should choose the model(s), which are going to be used before the start of the development of whole processes. Based on this statement, Floors conduct the processes of ISO/IEC 12207 with Waterfall development model sequence via giving useful information during game play. Additionally based on the statements, which come from ISO [56], there is no necessity to use all of the processes that are stated in the standard. They are described as attributes of the standard and they are suitable to be adjusted according to organization was discussed (see Chapter 2 section 2.10 tailoring process). According to the requirements and the needs of the organization during development of the software product they can be shaped. The processes can be chosen specifically. Moreover, in Floors, there is a structure of virtual firm in order to conduct representation of related personnel and processes of the standard. Therefore, this study uses main and most frequently used processes especially in traditional development models, which are waterfall, incremental, and iterative. Figure 15 shows the diagram of selected processes and their occurrence sequence during game play. To make participants active during game play various quests are located in different locations of floors while continuing processes. By this way participants directly involve the processes and their required activities with these quests. Figure 16 shows quests and their locations in game play.

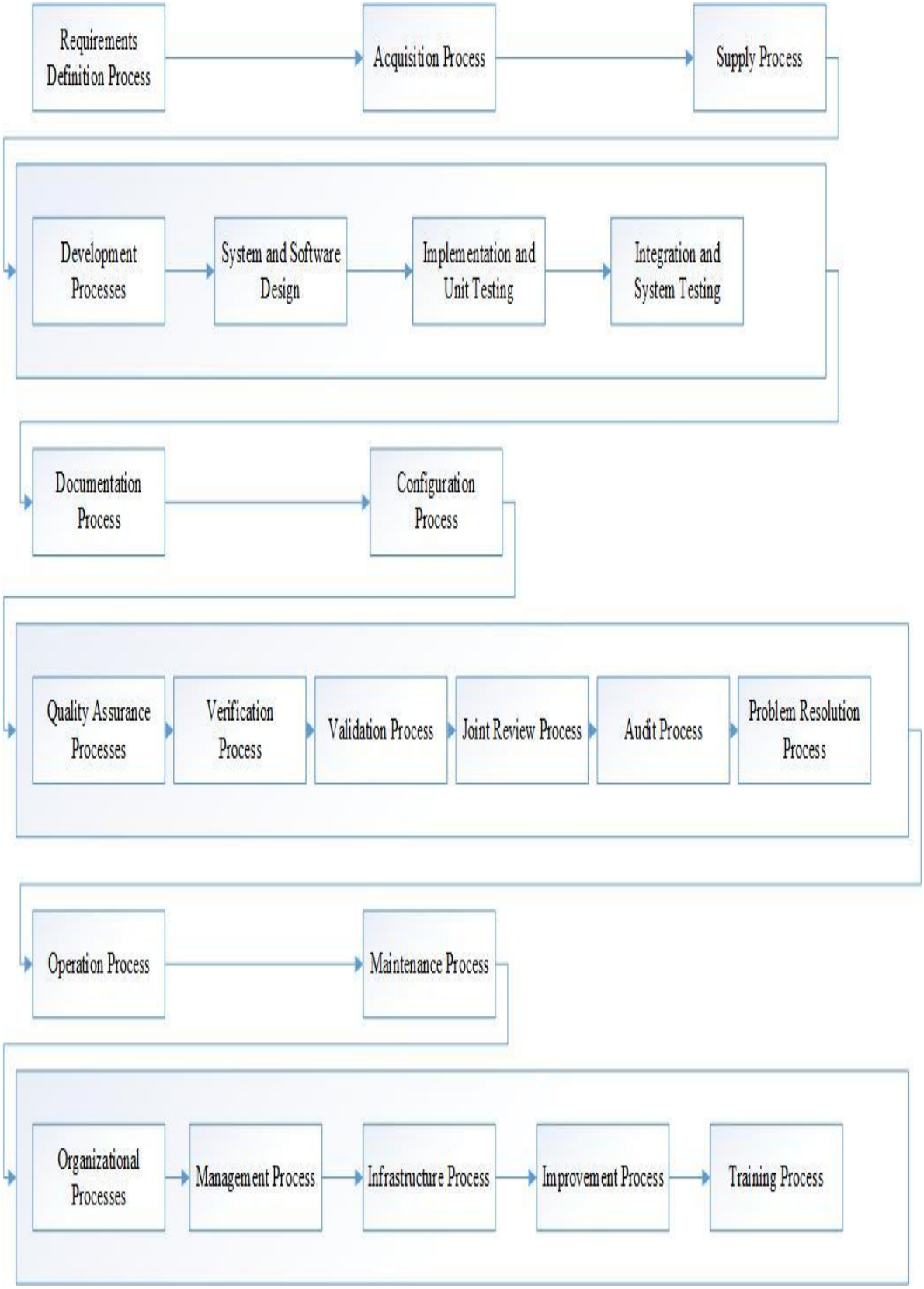

Figure 15 Selected and Implemented processes of ISO/IEC 12207 in Floors

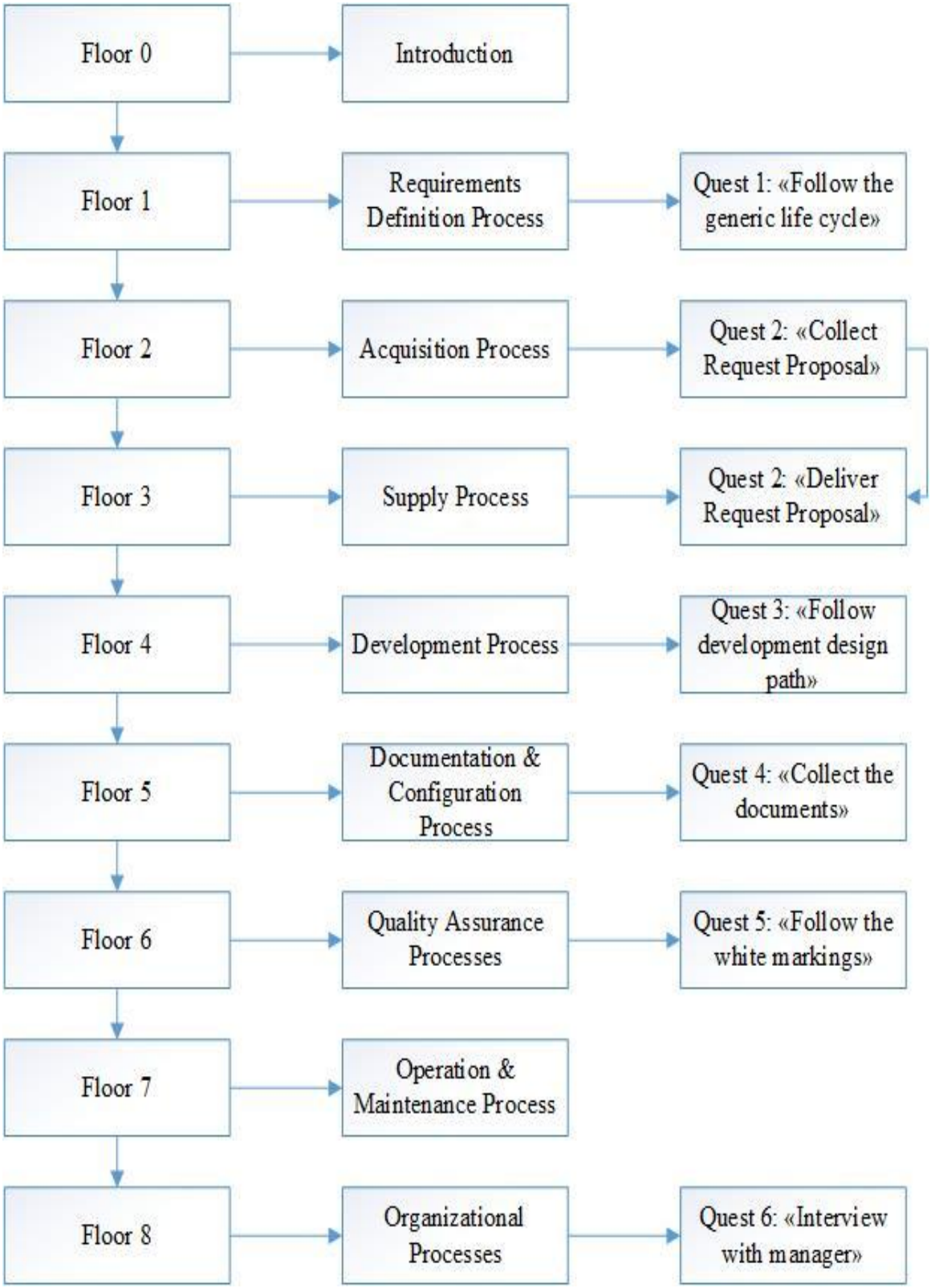

Figure 16 Dedicated quests for related processes and floors

Serious game environment has three scenes namely: the start menu scene, floors scene, and the game over scene. General game play duration takes place in floors scene where selected

processes of ISO/IEC 12207 Software Life Cycle Processes are explained with dialogues. When the game starts menu scene is presented to the user and this scene asks user to start the game by selecting start game button. This triggers the event that starts the floors scene where the actual game play appears. With the start of floors the first quest requires user to speak the NPC at 0^{th} floor for further instructions that is necessary for the user. Additionally she gives basic information about software engineering processes and main definition of ISO/IEC 12207. Figure 17 presents the screenshot of the 0^{th} floor. At the end of the dialogue she wants user to move forward to go upper floors. When the player reach the end of the floor h/she will come with flying stairs that can be used to step forward to the next floor at the same time this stairs contain sound and color changing collider functions to get attention of the player while progressing. With different arrangements of these stairs every floor has different difficulty level to be reached. By this way users try to complete small scale challenge in between different floors. Moreover they contain various tips about both the standard and SE topics to be more convenient tool for upcoming dialogues.

Figure 17 Screenshot of 0th Floor with Responsible NPC Personnel

When the user reaches 1^{st} floor with several tips about the definitions of the terms party and organization from stairs he/she encounters with a man who is responsible for analyzing user requirements. In this floor firstly desires of customers and getting detailed information about

them are explained briefly and then responsible NPC presents which and what type of activities are followed to maintain the process of Requirements Definition. Especially the definition of activity eliciting requirements is emphasized with its details to exposure the importance to the user. Figure 18 show the general perspective of 1st Floor. In the middle of the dialogue the result of the process which is Software Requirement Specification (SRS) document and its details are clearly described. At the end of the conversation NPC directs the user to the gate to reach the stairs which follows the 2nd Floor.

Figure 18 Screenshot of 1st Floor with responsible NPC personnel for Requirements Definition Process

At the second floor the oldest NPC welcomes the user for the explanation of Acquisition Process. In this floor the definition of the term acquisition is clearly defined to show the similarities between definition and the usage in the standard. After observations of game plays of users and their behaviors they understand the processes with the help of term definition and mutual activities, which are done by supplier, are explained. Additionally, a small quest that contains gathering and delivering relevant documents to the supplier is started to get the attention of the user by involving them in a virtual duty. At the end of the process user is directed to the stairs that follow the third floor. Figure 19 presents the general view of 2nd floor.

Figure 19 Screenshot of 2nd Floor with responsible NPC as an Acquirer

At the third floor responsible personnel welcomes user for supply process. NPC starts the explanation with his duties and responsible activities to maintain supply process. At the same time user get an achievement for completing the quest by arriving supplier and delivering

relevant documents to the NPC. After getting general description and basic definition of the process user learns details of being supplier. Figure 20 shows the general look of 3rd floor where supply process takes place. With the help of necessary information, which is given in second and third floor; users can see, learn and involve the acquisition and supply process and they have a chance to understand their mutual relation by involving dedicated quest.

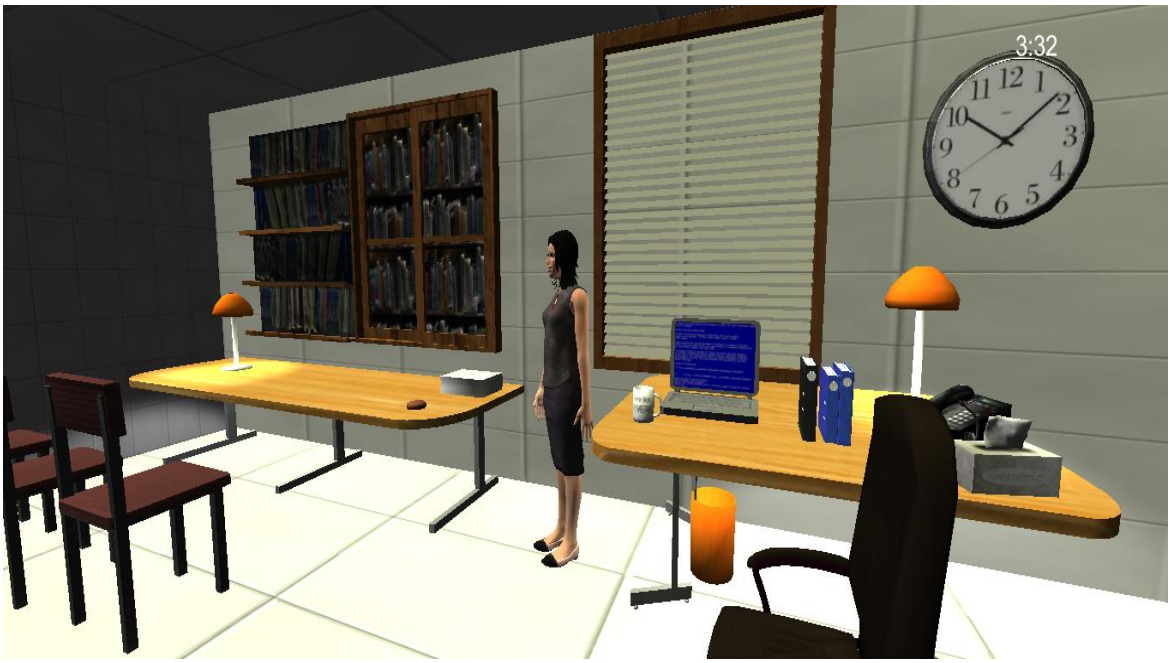

Figure 20 Screenshot of 3rd floor with responsible NPC as a supplier for Supply Process

In the 4th floor user involves development process. This floor's main goal is giving impetus to user to explain that this process is the backbone of the whole life cycle. In addition to this NPC explains the types of development process and their details. Moreover to get more attention from the user virtual elements such as computers and various textures are used to present the order of Waterfall development model structure and development process. By this way users have a chance to visualize the basic steps of Waterfall Model and more specialized topics which are related with design and testing especially the topics System and Software Design, Implementation and Unit Testing, Integration and System Testing covered with the help of textual and virtual elements of this floor. Figure 21 shows the main design of

development process floor. Beside technical information and visualization of Development Process of ISO/IEC 12207 and basics of SE the importance of allocation of hardware and software usage in a project life cycle is discussed among this floor's quests. According to observations users generally follow upcoming process which is documentation process without any hesitation with the help of visualization of user's path and quests.

Figure 21 Screenshot of 4th floor with NPCs as developers for maintaining Development Process

In fifth floor Documentation and Configuration Process are both discussed with different dedicated NPCs. In documentation stage basics of saving and recording information through life cycle and various kinds of documents sources, which are developers, managers and customers are explained. Via visual elements such as working printer, computers and incoming mails users are subjected the movements and motions during dialogue. Figure 22 presents general view of documentation process. In configuration stage identification of software items, modifications and releases of them are emphasized and the importance of ensuring completeness of them clearly explained to the users. While ensuring completeness of different software items users can see communication between different parties in the organization. Figure 23 presents general view of configuration process.

Figure 22 Screenshot of 5th floor with responsible NPC for Documentation Process

Figure 23 Screenshot of 5th floor with responsible NPC for Configuration Process

In sixth floor Quality related process of the standard are explained. Based on the definitions from chapter 2 section 2.8.4.4 processes such as verification, validation, joint review, audit, and problem resolution are described with examples and comparisons via different NPC. Each process is treated with a unique explanation approach to get more attention of the user. In this floor users follow markings to properly navigate inside the floor to complete every step without any conflict.

Firstly, quality assurance process' basics are covered especially while determining quality being unbiased during the period of maintaining the activities of this process is discussed and then relations of this process with inputs and outputs are revealed. Users can learn the differences between internal and external organizations and parties during quality assurance process session. Figure 24 shows general view of quality assurance process.

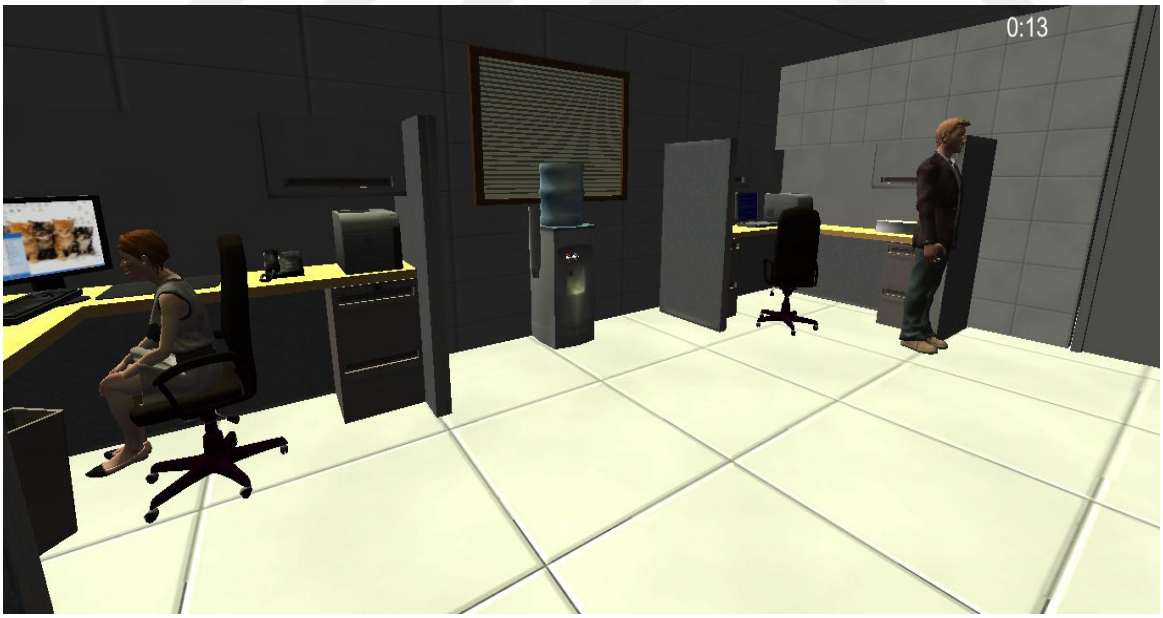

Figure 24 Screenshot of Quality Assurance Process section with NPCs

In verification process session definition of this process and outcomes are clearly explained. Consecutively, validation process starts and gives basic definition of the process. Moreover the differences and similarities are given to user with textual elements without any time interval to get immediate outcomes. Therefore, following these processes consecutively gives chance to user to see difference in definitions immediately. Figure 25 show verification session and validation session with a responsible NPC. After completing verification and validation user directly follows the white markings to reach joint review session where the framework to maintain the interactions between acquirer and supplier are showed from different perspective. Additionally, the definitions such as project status, products and activities of reviewer and reviewee are given to clarify how this process works and conducting parties' duties. Figure 26 shows NPC who explains the joint review process.

Figure 25 Screenshot of Verification and Validation Processes dialogue with responsible NPC

Figure 26 Screenshot of joint review process with responsible NPC

In audit process session users involve dialogue to learn responsible personnel duties and main definitions. Additionally, he/she involves main activity determining requirements plans and their appropriateness with the help of NPC. All of those processes create, record and state problems or defects and the user follow all of them to reach problem resolution process. By this way users can see the coherent infrastructure while maintaining the processes of the standard.

At the stage of problem resolution process users have a chance to understand a loop pattern by implicitly involving quests that are performed by the user previously. In this session users can clearly see the classification and prioritize attributes of problems to maintain the process adequately. At the end of all of these processes the user is directed to follow stairs with different types of tips to reach upcoming floor.

In 8th floor two processes operation and maintenance are maintained with two different NPC and dedicated working space. At first during the stage of operation process users involve dialogue to understand relations and connected processes to this process. NPC defines every relation according to the occurrence in the standard, i.e. tailoring, training and supply. By this way users can learn the exact relations like in real life and perform relevant quest to accomplish in virtual environment. Figure 27 presents overall design of operation process

workspace. After accomplishing operation process, the user can find his/her way to maintenance process with navigational elements and small tips. When the user reaches responsible personnel for maintenance process, the dialogue starts with the basic definition of the process. Users have chance to learn how this process is triggered to start and its main objective according to explanations that comes from NPC.

Figure 27 Screenshot of Operation Process with responsible NPC and work elements

In addition to them, users have a chance to understand the terms migration and retirement of software product or service with particular examples and related definitions. Moreover, users can get benefit from examples to learn how these activities are performed in accordance with the standard. Figure 28 shows operational environment of maintenance process. After maintenance process session user is directed to the longest stairs path follows manager's floor where managerial level topics are covered.

At the end of the longest stairs journey where the user gets various kinds of useful information and tips he/she reaches the 9^{th} floor where project manager's office is located. Before starting the dialogue with manager users are now familiar with the concepts and the processes of the standard as well as he/she gets general knowledge about SE related topics with the help of generic life cycle example and necessary information to complete the main scenario of the game.

Figure 28 Screenshot of Maintenance Process with responsible NPC

In the 9th floor an NPC representative for project manager of this virtual company welcomes user and he explains organizational life cycle processes with expanded information encapsulates the attributes of whole standard. The dialogue starts with manager's responsibilities and his relations with stakeholders. Therefore urgency of his critical responsibility and marketing strategies are revealed at the beginning of the dialogue. With the help of sympathetic personality of manager and his clear language users have a chance to follow every necessary information without any restraints. By this way a virtual interview with a manager is established and they can get benefit from this floor with the goal of accomplishing the interview and learning various specialized topics that are about standard's processes.

Without giving direct information about definitions and duties of project manager users find themselves in a natural speaking environment and narrative scenario while talking with the manager and get convenient tips indirectly from this dialogue such as management process and its relations with directly relevant process and specific tasks. Additionally, throughout the life cycle processes there is a curiosity about the beginning point of every process and its related activities. In this floor manager explains the initialization period of those processes and their control, execution, and evaluation. Through the middle of the dialogue every explanation is similar to the previous floors but in 9th floor manager expects users to answer some questions especially organizational life cycle processes with their knowledge. The manager explains and completes four types of these organizational life cycle processes with

the expectation of user involvement. Such types are management, infrastructure, improvement, and training. With catechizing about these processes users directly involve an interview with manager to learn about them and their details. Figure 29 shows the general view and NPC representative of project manager in 9th floor.

Figure 29 Screenshot of 9th floor with Organizational Processes with manager and NPCs

At the end of the dialogue with manager user have followed 17 main processes and 12 relevant activities of ISO/IEC 12207 Software Life Cycle Processes in conformance with Tailoring Processes definition and requirements to connect relations in every processes. As a final message when the player accomplished the last process Game Over message comes to the screen to mention for successfully completing the main structure of ISO/IEC 12207 Software Life Cycle Processes with Floors.

5.6 **Summary**

In conclusion, this chapter presents the main features of sessions and the main features of Floors. In order to conduct the study created sessions such as paper-based learning session and virtual learning session were described with their main characteristics and detailed features for clarifying the objectives of the study. While explaining the main features of Floors every process, which was selected, to satisfy the tailoring process needs were explained in detail. Additionally, to engage participants during game play, every quest and followed path were described with their distinctive features. The main aim is to demonstrate the proposed serious game's visual parts especially virtually created goods and modeled NPCs via standard's processes and defined dialogues.

Chapter 6

6. Results and Analysis

6.1 Introduction

This section briefly explains the results of the research and analysis of obtained data which were collected from participants in different sessions. In addition to this data process and its results are presented according to sessions. Moreover, there are also demographics of the participants in order to present main distribution of participants. All of the data analysis was done with the help of IBM SPSS Statistics program.

6.2 Measurements

In this study, variables such as age, gender, faculty department, and education were asked to fill at the start of the survey. However, this part was maintained for only showing the demographic information about participants and these variables have no direct impact in analysis session. As a part of quantitative research and for collecting the feedback from participants 5-point Likert scale based questionnaire was conducted. This is the adopted version of Game Play scale [81] to conduct assessment of Floors game play session (virtual learning session) and paper based learning session. Finally, at the end of both sessions participants were subjected to quiz which contains combined questions from ISO [57], IEEE Computer Society Project Management Quiz [94], and Software Engineering basics from Sommerville [95].

6.3 Demographics of the Participants

This section presents the demographics of the participants. These values have no direct impact on analysis and do not involve any correlation. Simply, they reflect the main attributes of participants.

In this research, there were 40 (n = 40) participants. 19 (47.5 %) of them were female and 21 (52.5 %) of them were male.

Second demographic was age value which was asked to participants while filling survey. 4 (10 %) of them were at age 19, 19 (47.5 %) of them were at age 20, 10 (25%) of them were at age 21, 4 (10%) of them were at age 22, and 3 (7.5%) of them were at age 23.

The other demographic was department in order to present the participants' departments. All of the participants were related with computer science via their departments or their double major. 27 (67.5%) of the participants were studying computer engineering, 10 (25%) of them were studying Electrical and Electronics Engineering in double major in computer engineering, 3 (7.5%) of them were studying industrial engineering and double major in computer engineering.

The last demographic was education which indicates the education level of participants who were involved the sessions. All of them (40) were undergraduate students in this study.

6.4 **Determining the score of survey data**

In order to determine the score of the game play scale survey 5-point Likert Scale was used in the range of "strongly disagree" which has the value 1, to "strongly agree" which has the value 5. To calculate the overall score of each participant every element of the survey were summed (see Appendix B.1). Game play scale survey questions contain both positive and negative questions for finding out user experiences from different perspectives. The questions 3, 4, and 8 presents negative meaning so for the sake of preserving statistical precision the likert scale were reversed (the value of "strongly disagree" is 5 and the value of "strongly agree" is 1) for these questions [101]. There were 15 questions (see Appendix B.3) in the questionnaire so the highest score is 75, the lowest score is 15 and the medium score is 45. Figure 34 shows scores of both sessions' surveys. Table 35 presents the scores of both sessions in Gameplay Scale survey.

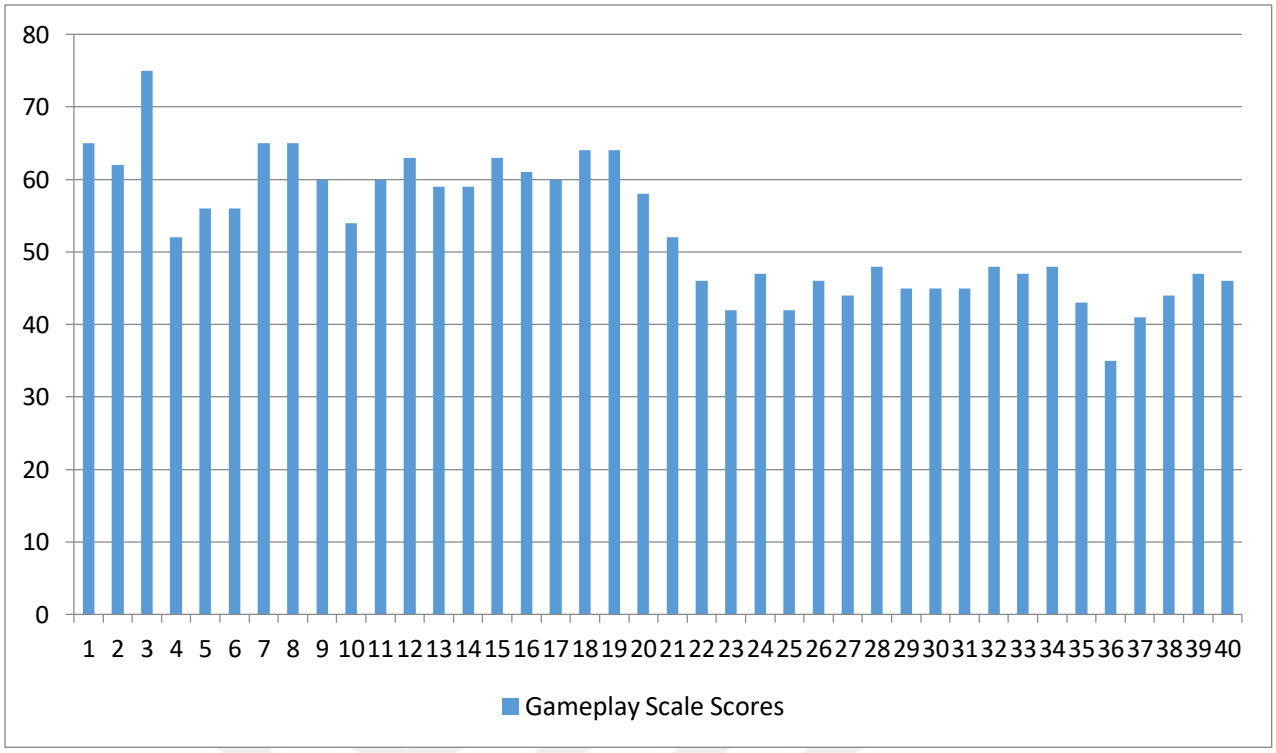

Figure 30 Calculated Gameplay scores from both sessions

Virtual Learning Session	Paper Based Learning Session
65	52
62	46
75	42
52	47
56	42
56	46
65	44
65	48
60	45
54	45
60	45
63	48
59	47
59	48
63	43
61	35
60	41
64	44
64	47
58	46

Table 35 Calculated Gameplay Survey Scores in Sessions

Beside survey scores there are also quiz scores of both sessions. In the quiz (see Appendix B.3) there were questions about processes of the standard and software engineering basics to reveal the knowledge of the participants after their session. There are 50 questions which indicates different perspectives and various subject in conformance with the processes of standard and game scenario. Each question is 2 points so the highest score is 100, the lowest score is 0 and the medium score is 50. Figure 35 presents the scores of the quiz from both sessions. Table 36 shows the scores of both sessions' quiz results.

In this study, average (mean) of Gameplay Scale scores in Virtual Learning Session is 61,05 and related standard deviation is 4.98. At the same time average (mean) of Paper Based Learning Session Gameplay Scale Scores is 45,05 and related standard deviation is 3.49. On the other hand the average (mean) of Quiz scores from Virtual Learning Session is 59.1 and related standard deviation is 8.12. Beside this the average (mean) of Quiz Scores from Paper-Based Learning Session is 50.1 and the related standard deviation is 9.68. According to Parnell [81] with the computations of Spearman's correlations Affective Experience of Gameplay Scale is 0.839.

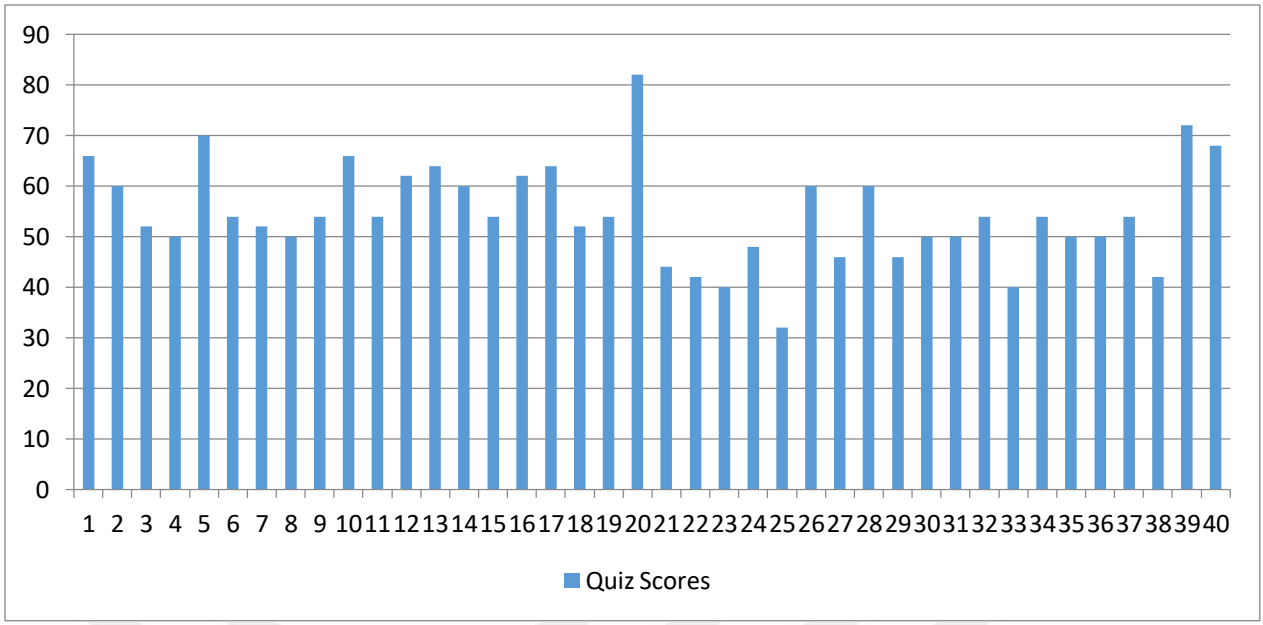

Figure 31 Calculated quiz scores from both sessions

Virtual Learning Session	Paper Based Learning Session
66	44
60	42
52	40
50	48
70	32
54	60
52	46
50	60
54	46
66	50
54	50
62	54
64	40
60	54
54	50
62	50
64	54
52	42
54	72
82	68

Table 36 Calculated quiz scores from both sessions

6.5 **Analysis of the data**

The main purpose of this research is to detect more beneficial and positive user experience that participants from the virtual learning session over paper based learning session. With this ambition participants were subjected to adopted version of game play scale at the end of each session. Due to different sessions were involved the research independent t-test for two independent samples was required to conduct the analysis of this study [49]. According to this context the hypothesis of the research;

H_0: The experience that the participants received from both sessions has no difference.

H_1: The experience that the participants received from both sessions has a difference.

In order to clarify while testing the hypothesis the null one states that there is no significant difference between two population means;

$$H_0 : \mu_1 = \mu_2 \text{ or } \mu_1 - \mu_2 = 0$$

While calculating the independent sample t-test IBM SPSS Statistics 23 was used. To accomplish the test, level of significance was taken as 95% (0.05). The degree of freedom (df = N - 2) were calculated as 38 since there were 40 (N = 40) participants. The equation for the degree of freedom is described like;

$$\text{df} = df_1 + df_2$$

$$= (n_1 - 1) + (n_2 - 1)$$

$$\text{where N} = n_1 + n_2 \text{ so df} = \text{N} - 2$$

According to t distribution table (see Appendix C.1) the demanding t value is 2.021 [49]. All of the independent t-test calculation was conducted via computer so the probability was too small to be considered [49]. In conformance with these values t value was calculated as 4.274 (t = 4.274). Table 37 presents the Independent sample t-test for this study. In agreement with the calculations and the comparisons of t-test distribution table the null hypothesis (H_0) is rejected [49]. Thus, H_1 which determines that there is a difference in population means between paper-based learning session and virtual learning session is granted statistically.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Game_Play_Scale_Score	Equal variances assumed	2,656	,111	4,274	38	,000	13,05000	3,05336	6,86879	19,23121
	Equal variances not assumed			4,274	21,637	,000	13,05000	3,05336	6,71155	19,38845

Table 37 IBM SPSS Statisitics output for Gameplay Scale Scores via Independent t-test

Participants who involved virtual learning session tend to have more positive experience and participants who involved paper based learning session tend to have slightly neutral experience according to Gameplay scale questionnaire scores. To state how exactly the measurement's variability there is a chance for obtaining this measurement's actual treatment effect. There is a measure for finding the effect size that is Cohen's d [49]. It can be stated as the standardized computational measure of the mean difference. Cohen's d is calculated as follows;

$$\text{estimated d} = \frac{M_1 - M_2}{\sqrt{Sp^2}}$$

To find the value of standard pooled variance (Sp), the formula as follow;

$$Sp = \frac{\sqrt{(N_1 - 1) * {S_1}^2 + (N_2 - 1) * {S_2}^2}}{\text{df}}$$

In this formula S_1 and S_2 stands for the standard deviations of the independent samples. The values are respectively 13.20 and 3.49. For the denominator df stands for the degree of freedom which is 38 due the sample size 40 (df = N - 2). According to these values standard pooled variance was calculated as 9.65 (Sp = 9.65) and Cohen's d was calculated as 1.35 (d = 1.35). To state how effective the result is there are defined values for d which are 0.2 (small effect), 0.5 (medium effect), and 0.8 (large effect) [49] . Due to this scale the effect size of this study (d = 1.35) is large.

While testing the hypothesis the independent t-test calculation lets researchers for computing the effect size by calculating the percentage of variance for r^2. The computation of r^2formula [49] is state as follows;

$$r^2 = \frac{\mathrm{t}^2}{t^2 + df}$$

In this formula the t value is 4.274 (t = 4.274) and df value is 38 (df = 38). The result is 0.324. According to the scale that is defined for the values of r^2 0.01 (small effect), 0.09 (medium effect), 0.25 (large effect) [49]. In conformance with these values this study r^2 = 0.324 (32.4%) has large effect.

Additionally, this research also sought the assessment of participants via their obtained knowledge from sessions. For the sake of accomplishing this ambition again independent t-test for two independent samples were used while the analysis of obtained quiz scores. Again level of significance was taken as 95% (0.05). The degree of freedom (df = N - 2) were calculated as 38. The critical t value should be 2.021. Under the light of these values t value was calculated as 3.186 (t = 3.186). As a supportive statement with this calculation the null hypothesis (H_0) is rejected again. Table 38 presents the output from IBM SPSS Statistics with quiz data.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Quiz_Scores	Equal variances assumed	,129	,722	3,186	38	,003	9,00000	2,82452	3,28206	14,71794
	Equal variances not assumed			3,186	36,881	,003	9,00000	2,82452	3,27636	14,72364

Table 38 IBM SPSS Statistics output for quiz scores via Independent t-test

Similar to Game play Score analysis for the quiz scores analysis again Cohen's d and r^2 was calculated in order to support the results and to clarify how effective the study is from the side of quiz assessment. Cohen's d value was calculated as1.06 (d = 1.06) which shows the large effect since it is greater than the value 0.8 and the r^2 value was calculated as 0.210 ($r^2 = 0.210$) again it shows the large effect due to it is greater than 0.8 [49].

6.6 Threats to validity

There can be threats which can cause some decrease in essence of the results. However, in this research, possible threats were eliminated in order to get optimum results. According to the definition from Yilmaz [112] "threats to validity" can be seen as potential circumstances which can change usefulness and exactitude of the study in a wrong way. In addition to this definition threats can be accepted as factors which effects to the results in a bad way [42].

There can be some threats that can be negatively alter the results of the study. The threats for this study can be stated like;

- Background of the participants may change or affect the results of data analysis. However, in order to preserve the target participants' main characteristics, demographics were collected and they clearly present age, education and department data to reflect the similar and familiar backgrounds. Because of this reason participants were selected from undergraduate students who take exactly the same courses.
- Time interval between surveys and quiz can cause a possible threat. However, there were no extraordinary condition since surveys and quizzes were conducted immediately at the end of the sessions and the same day.
- Survey scoring can cause a threat, but to avoid this problem both sessions were subjected to same liker scale questionnaire and the same adopted version of Game play scale.
- The participants, who can be in both sessions, may intently select wrong answers to the questions, which are subjected to them in sessions.
- A participant, who is in the virtual learning session, may also examine and study the official standard documentation.
- A participant, who is in the paper based learning session, may also play Floors and get benefit from its features.

Although all situations have been considered, the above threats may affect the results of the study negatively.

6.7 Validation Interview

This first part of this study was conducted using quantitative research techniques. , However, a set of validation interviews for “Floors” were also conducted in order to explore the outcomes of this research. In order to accomplish qualitative attitudes of the study these interviews were conducted. With the help of three academics who are experienced instructors in computer engineering and software engineering for many years, feedbacks were obtained. After completing sessions these interviews were maintained. In these interviews 3 questions were asked to get direct opinions and feedback about Floors. The questions are listed below;

Question 1: What do you think about main characteristics of Floors?

Question 2: What kind of suggestions can you make for further improvements of the game?

Question 3: What kind of advantages / disadvantages Floors have over paper based learning session?

According to answers for the first question one of the answer is "*the game creates an atmosphere where you can feel the office environment and do not afraid what will going on with the stairs, just walk and talk with model and follow the path*". They all agree that designing a process path and interactive involvement while game play helps their understandings of the concept of the standard. For the second question all of the answers were about adding more features while completing the game scenario. One answer clearly stated, "*being interactive game needs constant care and Floors have potential for expanding its features*". Lastly, for the third question participants stated there were no doubts about having advantages as being interactive and graphically endorsed. One answer states that "Undoubtedly the game has many advantages over reading the processes and trying to understand the standard. Graphical creations and characters help the observations."

All the answers were collected and classified according to their attitudes towards the questions. Table 39 presents the collected results of interviewees.

	Interview 1	Interview 2	Interview 3
Question 1	Positive	Positive	Positive
Question 2	Positive	Neutral	Neutral
Question 3	Positive	Positive	Positive

Table 39 Collected and classified answers from interviews

According to these interviews, participants granted that our serious game environment has a potential feature for increasing the decision making skills through the game play duration. In addition to this the serious game environment helps them to involve the process in a more visual way with the help of graphically fostered 3D office environment. In conformance with the considerations and opinion of participants, our serious game environment is a valuable tool to expose knowledge about the standard.

6.8 **Summary**

In conclusion, this chapter includes information about demographics and data analysis which were collected from participants. Details about demographics especially frequencies and percentages of each element in sample population were described. Additionally, this chapter focuses on details of the analysis which was conducted via IBM SPSS Statistics 23. The analysis method independent t-test was selected and performed to support the validation of the research. The next chapter includes conclusion and possible future plans and works which are related to the study.

Chapter 7

7. Conclusion and future works

The primary goal of the study is to design and implement a serious game to help individuals to improve their knowledge about ISO/IEC 12207 in an interactive virtual 3D environment. Rather than just reading the ISO/IEC 12207 standard, participants are able to involve a live process with dialogues and quests. Every necessary step which was taken to develop the game completes particular objects in order to present adequate environment. Every digital creation was specifically designed for preserving this ambition. Moreover, several different programs were used in order to make more refined virtual environment with engaging features. The data analysis indicates there is obvious difference between the results of each learning session. In this methodology independent samples t-test were used in order to present the difference between sessions. According to this analysis the proposed methodology works as it was expected. Moreover participants got some beneficial knowledge beside their experiences in conformance with the quiz results. The research process is based on a hybrid approach, i.e. quantitative (game play scale questionnaire, quiz questions) and qualitative analysis (validation interviews) indicate that significant user experience and beneficial knowledge were obtained by participants via digitally interactive way. Beside the positive feedback in different perspectives there were also some cases where participants stayed neutral especially the subjects in fun elements. Because serious games' do not directly entertain the users with traditional way of entertainment, rather they are informative and instructional [11]. Due to the ambition of creating a serious game maintaining the game scenario particularly in the standard's processes was a hard process and actually it was an obvious obstacle while the development phase of Floors. In order to overcome these outcome future plans will be dedicated to be more interactive and to be more engaging. Future plans can be stated like; (i) graphically leveraged game play duration, (ii) animations with more detailed expressions, (iii) adding more graphically fostered creations.

Floors initially designed as a serious game which runs on desktop computers. According to options and opportunities that are apparently about economical subjects preliminary development phase was planned only for desktop environment. However, today's society requires to stay online and being mobile instead of reaching the computers only from accustomed places like homes, schools etc. Mobile version of Floors can be an aim in order to get more feedback from people and also to reach different domains. But this aim significantly

requires strict optimizations to work adequately on mobile platforms. Due to less computational power of mobile devices every digitally created object should be optimized to satisfy the mobile devices" capabilities. Particularly, every 3D creation’s graphical representations should be decreased in order to occupy less amount of memory. In addition to this there should be instant editing and adjustments for the frames of the animations again to preserve the amount of memory at the same time computational power.

Notably, for the future improvement of Floors, Virtual Reality (VR) can be stated. Virtual Reality is a kind of computer technology which simulates the specified environments to enable user interaction [48]. Additionally, users get perceptional feelings about their existence in a virtually created world which is known as immersion [34]. As a relatively new game technology VR can enhance the prominent strength of Floors. Virtual Reality does not require traditional input devices. Users have chance to control their actions and movements without any limitations. However, in traditional gaming there is a need for specified input devices and controllers in order to move and complete specified activities during game play. By using VR technology users can examine and explore the features of the game and the game play scenario by themselves in a more free way.

8. References

1. **Abt, C. (1987)** Serious games. University Press of Amer
2. **Acuña, S. T., & Ferré, X. (2001, July).** Software Process Modelling. In ISAS-SCI (1) (pp. 237-242).
3. **Acuña, S. T., López, M., JURISTO, N., & MORENO, A. (1999).** A process model applicable to software engineering and knowledge engineering. International Journal of Software Engineering and Knowledge Engineering, 9(05), 663-687.
4. **Acuna, S.T., Juristo, N., Moreno, A.M., Mon, A.:** A Software Process Model Handbook for Incorporating People's Capabilities. Springer-Verlag (2005)
5. **Adiyeva, P., Turan, A. F., Ergobek, S., Dudarisha, S., & Kemalkizi, Z. (2014).** The Role of Dialogue in the Interpretation of National Folklore. Procedia-Social and Behavioral Sciences, 143, 79-82.
6. **Akdemir, Ö., Vural, Ö. F., & Çolakoğlu, Ö. M. (2015).** Prospective Teachers' Likelihood of Performing Unethical Behaviors in the Real and Virtual Environments. Procedia-Social and Behavioral Sciences, 174, 3441-3446.
7. **Alvarez, J., Djaouti, D.:** Introduction au serious game. Questions théoriques (2010)
8. **America's Army.** Retrieved from https://www.americasarmy.com/
9. **Animated 3D characters Mixamo** Retrieved from https://www.mixamo.com/
10. **Apperley, T. H. (2006)** Genre and Game Studies: Toward a critical approach to video genres. Simulation & Gaming, 37 (1), pp6-27.
11. **Avedon, E. M., & Sutton-Smith, B. (1971).** The study of games. New York et al..: Wiley, 405
12. **Aydan, U., Yilmaz, M., & O'Connor, R. V. (2015, June).** Towards a serious game to teach iso/iec 12207 software lifecycle process: an interactive learning approach. In International Conference on Software Process Improvement and Capability Determination (pp. 217-229). Springer International Publishing.
13. **Baker, A., Navarro, E.O., Van Der Hoek, A.:** An experimental card game for teaching soft- ware engineering processes. Journal of Systems and Software 75 (2005) 3–16
14. **Baker, A., Navarro, E.O., Van Der Hoek, A.:** An experimental card game for teaching soft- ware engineering. In: Software Engineering Education and Training, 2003. (CSEE&T 2003). Proceedings. 16th Conference on, IEEE (2003) 216–223

15. **Basil, V. R., & Turner, A. J. (1975).** Iterative enhancement: A practical technique for software development. Software Engineering, IEEE Transactions on, (4), 390-396.
16. **Bernard Suits,** Grasshopper: Games, Life, and Utopia (Boston: David R. Godine,1990), pp. 34-41
17. **Bernhaupt, R. (2010).** User experience evaluation in entertainment. In Evaluating User Experience in Games (pp. 3-7). Springer London.
18. **Bjork, S., & Holopainen, J. (2006).** Games and design patterns. The game design reader, 410-437.
19. **Blender** Retrieved from https://www.blender.org/features/2-77/
20. **Boehm, B. W., (1976)** Software Engineering IEEE Trans Computer, C-25, pp. 1226–1241.
21. **Bolcer, G.A., R.N. Taylor,** Advanced workflow management technologies, Software Process Improvement and Practice, 4, 3, 125-171, 1998.
22. **Breuer, J. S., & Bente, G. (2010).** Why so serious? On the relation of serious games and learning. Eludamos. Journal for Computer Game Culture, 4(1), 7-24.
23. **Bruffee, K. A. (1999).** Collaborative learning: Higher education, interdependence, and the authority of knowledge. Johns Hopkins University Press, 2715 North Charles Street, Baltimore, MD 21218-4363.
24. **Bryson, A., Forth, J., & George, A. (2012).** Workplace social dialogue in Europe: An analysis of the European Company Survey 2009.
25. **Caldero´n, A., Ruiz, M.:** Prodec: a serious game for software project management training. In: ICSEA 2013, the Eighth International Conference on Software Engineering Advances. (2013) 565–570
26. **Captain Novolin,** Learn about diabetes in this educational platformer by assuming control of a diabetic superhero, whose task it is to save the mayor from alien abductors, Retrieved from http://www.giantbomb.com/captain-novolin/3030-4948/
27. **Captain Novolin,** Who thought this would be fun? Retrieved from http://www.gamespot.com/captain-novolin/user-reviews/2200-331845/
28. **Caulfield, C., Veal, D., Maj, S.P.:** Teaching software engineering project management–a novel approach for software engineering programs. Modern Applied Science 5 (2011) p87
29. **Chore Wars.** Retrieved from. http://www.chorewars.com/
30. **Clark C. Abt, Serious Games** (New York: Viking Press, 1970),6.

31. **Clarke, P., & O'Connor, R. V. (2011).** An approach to evaluating software process adaptation. In Software Process Improvement and Capability Determination (pp. 28-41). Springer Berlin Heidelberg.
32. **Clearwater, D. (2011)** What defines video games genre? Thinking about genre study after the great divide. Loading…, 5 (8).
33. **Collins, H., (2008).** Video Games Hone Triage Skills: British video game developer creates a prototype that sharpens first responders' triage skills. Retrieved from http://www.emergencymgmt.com/training/Video-Games-Hone-Triage.html
34. **Costello, P. J. (1997).** Health and safety issues associated with virtual reality: a review of current literature (pp. 1-23). Advisory Group on Computer Graphics.
35. **Creighton, R. H. (2010).** Unity 3D Game Development by Example: A Seat-of-Your-Pants Manual for Building Fun, Groovy Little Games Quickly. Packt Publishing Ltd.pp. 1-115
36. **Creswell, J. W., & Clark, V. L. P. (2007).** Designing and conducting mixed methods research.
37. **Demirors, O., Demirors, E., Tarhan, A., Yildiz, A.:** Tailoring iso/iec 12207 for instruc- tional software development. In: Euromicro Conference. Volume 2., IEEE Computer Society (2000) 2300–2300
38. **Denzin, N. K., & Lincoln, Y. S. (2011).** The SAGE handbook of qualitative research. Sage.
39. **Derntl, Michael, Kroncik Milos, Klamma Ralf, Chacon Jonathan, Hernandez-Leo Davinia (2014).**Gamification of Learning Design Environments Workshop. 10th European Summer School on Technology Enhanced Learning, Malta
40. **Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011, September).** From game design elements to gamefulness: defining gamification. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments (pp. 9-15). ACM.
41. **Drappa,A.,Ludewig,J.:**Simulationinsoftwareengineeringtraining.In:Proceedingsofthe 22nd international conference on Software engineering, ACM (2000) 199–208
42. **Fayter, D., McDaid, C., & Eastwood, A. (2007).** A systematic review highlights threats to validity in studies of barriers to cancer trial participation. Journal of clinical epidemiology, 60(10), 990-e1.
43. **FourSquare.** Retrieved from https://tr.foursquare.com/about

44. **Futrell, R.T., Shafer, L.I., Shafer, D.F.:** Quality software project management. Prentice Hall PTR (2001)
45. **Gee, J.P.:** What video games have to teach us about learning and literacy. Computers in Entertainment (CIE) 1 (2003) 20–20
46. **Gee, J.P.:** What video games have to teach us about learning and literacy. Macmillan (2014)
47. **Gibbs, W.W.:** Software's chronic crisis. Scientific American 271 (1994) 72–81
48. **Gonzales, D., Criswell, D., & Heer, E. (1991).** Automation and Robotics for the Space Exploration Initiative: Results from Project Outreach (No. RAND/N-3284-AF/NASA). RAND CORP SANTA MONICA CA.
49. **Gravetter, F. J., & Wallnau, L. B. (2016).** Statistics for the behavioral sciences. Cengage Learning.
50. **Gulec, U., & Yilmaz, M. (2016).** A serious game for improving the decision making skills and knowledge levels of Turkish football referees according to the laws of the game. SpringerPlus, 5, 622.
51. **Herranz Sánchez, E., Colomo-Palacios, R., de Amescua-Seco, A., and Yilmaz, M. (2014)**. Gamification as a disruptive factor in software process improvement initiatives. Journal of Universal Computer Science, 20(6), 885-906.
52. **Hopkins, W. G. (2008).** Quantitative research design.
53. **Hosier, W. A. (1961).** Pitfalls and safeguards in real-time digital systems with emphasis on programming. Engineering Management, IRE Transactions on, (2), 99-115
54. **Hugos, M.,** Enterprise Games, O'Reilly 2012.
55. **Humphrey, W. S. (1995).** A discipline for software engineering. Addison-Wesley Longman Publishing Co., Inc..
56. **ISO/IEC 12207** Most Frequently Asked Questions Retrieved from http://www.12207.com/12207-q-a.html
57. **ISO/IEC 12207:2008** Systems and software engineering Software life cycle processes Retrieved from http://www.iso.org/iso/catalogue_detail?csnumber=43447
58. **ISO/IEC:** Amendment to ISO/IEC12207-2008-Systems and software engineering Software life cycle processes. (2008)
59. **ISO/IEC: ISO/IEC12207-1995**- Systems and software engineering Software life cycle processes. (1997)

60. **Jillian Michaels.** Retrieved from. http://www.jillianmichaels.com/fit/the-community/the-challenges
61. **Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007).** Toward a definition of mixed methods research. Journal of mixed methods research, 1(2), 112-133.
62. **Jones, A.:** Iso 12207 software life cycle processes fit for purpose? Software Quality Journal 5 (1996) 243–253
63. **Kapp, K.M.** the gamification of Learning and Instruction, Pfeiffer, 2012
64. **Karunatillake, N. C., Jennings, N. R., Rahwan, I., & McBurney, P. (2009).** Dialogue games that agents play within a society. Artificial intelligence, 173(9), 935-981.
65. **Khosrow-Pour, M.:** Encyclopedia of information science and technology. Volume 1. IGI Global (2008)
66. **Kling, R., and W. Scacchi**, The Web of Computing: Computer Technology as Social Organization, Advances in Computers, 21, 1-90, Academic Press, New York, 1982.
67. **Lindquist, M., Lange, E., & Kang, J. (2016).** From 3D landscape visualization to environmental simulation: The contribution of sound to the perception of virtual environments. Landscape and Urban Planning, 148, 216-231.
68. **Master Branch.** Retrieved from https://www.masterbranch.com/
69. **McGonigal, J.:** Reality is broken: Why games make us better and how they can change the world. Penguin Pr (2011)
70. **Microsoft Visual Studio** Achievements. Retrieved from. https://channel9.msdn.com/achievements/visualstudio
71. **Muijs, D. (2010).** Doing quantitative research in education with SPSS. Sage.
72. **Mullen, T. (2011).** Mastering blender. John Wiley & Sons. pp.1-100
73. **Munassar, N.M, A., & Gouvardhan, A. (2010)** A comparison between five models of software engineering UCSI, 5, pp. 95-101
74. **Navarro, E.O., Baker, A., Van Der Hoek, A.:** Teaching software engineering using simulation games. In: ICSIE04: Proceedings of the 2004 International Conference on Simulation in Education. (2004)
75. **Navarro, E.O., van der Hoek, A.**: Simse: An interactive simulation game for software engi- neering education. In: CATE. (2004) 12–17
76. **Nike+/NikeFuel**. Retrieved from https://secure-nikeplus.nike.com/plus/
77. **Packy & Marlon** ["Edutainment" Action Game], So...what do we have here?, Retrieved from http://www.gamingsanctuary.com/PackyAndMarlon.html

78. **Packy and Marlon,** IGN, Summary, Retrieved from http://www.ign.com/games/packy-and-marlon/snes-605708
79. **Pamoja Mtaani** (Together in the Hood) video game for Kenyan youth, Retrieved from http://www.yahanet.org/gallery/video/383
80. **Pamoja Mtaani**, An HIV/AIDS prevention videogame created through a Public-Private Partnership between PEPFAR and Warner Bros. Entertainment, Retrieved from http://www.slideshare.net/kgavoni/pamoja-mtaani-games-for-health-2009-presentation
81. **Parnell, M. J., Berthouze, N., & Brumby, D. (2009).** Playing with Scales: Creating a Measurement Scale to Assess the Experience of Video Games.University College London, London, UK.
82. **Pedreira, O., García, F., Brisaboa, N., & Piattini, M. (2015).** Gamification in software engineering–A systematic mapping. Information and Software Technology, 57, 157-168.
83. **Re-mission, Common Sense Media** says Shooter lets young cancer patients take aim at their disease, What's it about?, Retrieved from https://www.commonsensemedia.org/game-reviews/re-mission
84. **Robertson, Margaret, 2010**, Can't Play, won't play, Retrieved from http://hideandseek.net/2010/10/06/cant-play-wont-play/
85. **Royce, W. W. (1970, August).** Managing the development of large software systems. In proceedings of IEEE WESCON (Vol. 26, No. 8, pp. 328-388).
86. **Salen, K., & Zimmerman, E. (2004).** Rules of play: Game design fundamentals. MIT press,pp. 99-113
87. **Samsung Nation**. Retrieved from. https://www.samsung.com/us/support/account/
88. **Scacchi, W. (1984).** Managing software engineering projects: a social analysis. Software Engineering, IEEE Transactions on, (1), 49-59.
89. **Scacchi, W. (2001).** Process models in software engineering. Encyclopedia of software engineering.
90. **ScrumKnowsy Team Edition**, MixRank. Retrieved from. https://mixrank.com/appstore/apps/578752073
91. **ScrumKnowsy**. Retrieved from. http://www.scrumknowsy.com/home
92. **Serious Games,** Can games change your business and change the world?, Post-event briefing by David Rowan, editor of WIRED magazine, Retrieved from http://www.trusim.com/files/SeriousGames_Nov2010.pdf

93. **Singh, R.:** International standard iso/iec 12207 software life cycle processes. Software Process Improvement and Practice 2 (1996) 35–50
94. **Software Project Management Knowledge Area Certificate** Retrieved from https://www.computer.org/web/education/project-management
95. **Sommerville, I., (2009).** Software Engineering 9 th ed.
96. **StackOverFlow.** Retrieved from. Http://stackoverflow.com/
97. **Strauss, A., & Corbin, J. (1990).** Basics of qualitative research (Vol. 15). Newbury Park, CA: Sage.
98. **Tashakkori, A., & Creswell, J. W. (2007).** Editorial: The new era of mixed methods. Journal of mixed methods research, 1(1), 3-7.
99. **Tesch R., (1990),** Qualitative Research: Analysis Types and Software Tools". Psychology Press, pp. 1-177.
100. **Tesch, R. (2013).** Qualitative research: Analysis types and software. Routledge.
101. **Tezbasaran, A. (2008)** Likert tipi Ölçek Hazırlama Kılavuzu.
102. **Triage Trainer,** Trusim Case Study - Triage Trainer, Retrieved from http://www.trusim.com/?page=CaseStudy
103. **TSE** Yazılım Yaşam Döngüsü Süreçleri Belgelendirilen Kuruluşlar. Retrieved from https://www.tse.org.tr/tr/icerikdetay/945/1216/belgelendirilen-kuruluslar.aspx
104. **Tsui, F.F.:** Essentials of software engineering. Jones & Bartlett Publishers (2014)
105. **Unity Game Engine** Retrieved from https://unity3d.com/
106. Uskov, V., & Sekar, B. (2014, October). Gamification of software engineering curriculum. In Frontiers in Education Conference (FIE), 2014 IEEE (pp. 1-8). IEEE.
107. **van der Woude, T.(2012).** Defining serious games.
108. **Von Wangenheim, C.G., Savi, R., Borgatto, A.F.:** Deliver!–an educational game for teaching earned value management in computing courses. Information and Software Technology 54 (2012) 286–298
109. **Vyas, D., van der Veer, G., & Nijholt, A. (2013).** Creative practices in the design studio culture: collaboration and communication. Cognition, Technology & Work, 15(4), 415-443.
110. **Xu, Y. (2011).** Literature review on web application gamification and analytics. Honolulu, HI, 11-05.

111. **Yilmaz M., Saran M., O'Connor R.V. (2014)** Towards a quest-based contextualization process for game-based learning. In: Busch C (ed) 8th European Conference on Games Based Learning, Academic Conferences and Publishing International Limited, vol 2, pp 645–651.

112. **Yilmaz, M. (2013).** A software process engineering approach to understanding software productivity and team personality characteristics: an empirical investigation (Doctoral dissertation, Dublin City University).

113. **Yilmaz, M. and O'Connor, R. V.,** A Scrumban Integrated Gamification Approach To Guide Software Process Improvement: A Turkish Case Study, Tehnicki Vjesnik (Technical Gazette), Vol. 23, No. 1, 2016

114. **Yilmaz, M., O'Connor, R. V., & Clarke, P. (2015).** Software development roles: a multi-project empirical investigation. ACM SIGSOFT Software Engineering Notes, 40(1), 1-5.

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

Appendices

Part 1

Appendix A

A.1 Speech Text of Identification for Virtual Learning Session

Hello, my name is Ufuk Aydan. I am a graduate student in computer engineering at Çankaya University. To accomplish my thesis a serious game was developed in order to introduce ISO/IEC 12207 Software Life Cycle Processes and to teach basics of software engineering and the standard's processes.

To complete this two independent sessions were prepared. These sessions are virtual learning session and paper based learning session. You are involving the virtual learning session. To complete the session you will play the game which includes processes of the standard. After you finished the game play duration we have a questionnaire that will be filled by every participants. There is no need for writing names and all of the information you provided will be kept by researcher. The questionnaire is not too much long and takes approximately 20 minutes to complete. Please carefully read the statements and answer them according to your experiences during the session. At the end of your questionnaire please complete the quiz which contains questions from you were introduced during the session. Participants are allowed to ask any type of question that are related with the study please do not hesitate.

Thank you for your interest.

A.2 Speech Text of Identification for Paper-based Learning Session

Hello, my name is Ufuk Aydan. I am a graduate student in computer engineering at Çankaya University. To accomplish my thesis a serious game was developed in order to introduce ISO/IEC 12207 Software Life Cycle Processes and to teach basics of software engineering and the standard's processes.

To complete this two independent sessions were prepared. These sessions are virtual learning session and paper based learning session. You are involving the paper-based learning session. To complete the session you will follow the official documentation of the standard. Additionally, you will have a path of selected process that you will follow. After you finished the session duration we have a questionnaire that will be filled by every participants. There is no need for writing names and all of the information you provided will be kept by researcher. The questionnaire is not too much long and takes approximately 20 minutes to complete. Please carefully read the statements and answer them according to your experiences during the session. At the end of your questionnaire please complete the quiz which contains questions from you were introduced during the session. Participants are allowed to ask any type of question that are related with the study please do not hesitate.

Thank you for your interest.

Part 2

Appendix B

B.1 Adopted Version of Gameplay Scale Survey

Age					
Gender					
Faculty Department					
Education					
	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The experience that I had from session was positive					
I was able to follow the processes					
I have had problems to understand definitions					
The assessment duration was boring					
In general I always be aware of what to do during the session					
I thought the session was interactive					
I found the session was fun					
I was unaware of understanding what to do next					
Involving the session made me happy					
The session trained me in the concept and processes					
I forgot about my surroundings while involving the session					
I realized the session's characteristic features					
Aesthetically the session was impresive					
I thought the level of difficulty was right for me					
I understood and followed my aim during the session					

B.2 Assessment Quiz

Software Engineering, Software Development Life Cycles, and ISO 12207 Software Life Cycle Processes Quiz

1. _____ is a property that have to be exhibited in order to understand the specified problem addressing to software in the real world.
   a. Software Requirement
   b. Requirements Specification
   c. Software Engineering
   d. Requirements Validation
2. _____ are kind of statements that can be seen in natural language and also with diagrams to present what services the software system is expected to operate. They are written for customers.
   a. Market Analysis
   b. Consistency Analysis
   c. User Requirements
   d. Requirements Specification
3. _____ is a document which presents detailed descriptions about software system services and constraints.
   a. Consistency Analysis
   b. System Requirements
   c. User Requirements
   d. Software Requirements
4. Which activity is associated with quality control?
   a. Coding
   b. Managing
   c. Testing
   d. Planning
5. Identifying needs, requirements and subjects that are related to customers' desires. This is done in which stage?
   a. Documentation
   b. Analysis
   c. Validation
   d. Elicitation
6. The main feature of _____ is production support.
   a. Waterfall
   b. Iterative
   c. Agile
   d. Maintenance
7. Name the basic steps of Software Development Models?
   a. Requirements, design, testing, coding, and maintenance
   b. Requirements, testing, design, coding, and maintenance
   c. Requirements, design, coding, testing, and maintenance
   d. Requirements, design, coding, maintenance, and testing
8. What are the following topics are related with the project manager?
   a. Business Development
   b. Scheduling
   c. Selection of Software Development Processes
   d. All of the above
9. What is the general name of the entire process of ensuring quality?
   a. Quality Management
   b. Quality audit
   c. Quality check
   d. Quality standard

10. The name of the software life cycle activity which involves obtaining the requirements from the user.
    a. Requirements Analysis
    b. Requirements Elicitation
    c. Domain Analysis
    d. Market Analysis
11. Who is the person with complete responsibility in the success of project and relations with stakeholders?
    a. Project Leader
    b. Project Manager
    c. System Analyst
    d. End User
12. What is the name of software life cycle activity which involves testing the software product in an environment which matches the operational environment?
    a. System Testing
    b. Unit Testing
    c. Acceptance Testing
    d. Regression Testing
13. Who is the person responsible for reading programming specifications that is specified by system analyst and then translate that specifications into suitable code?
    a. Project Manager
    b. Project Leader
    c. System Analyst
    d. Programmer
14. The name of the software life cycle activity that involves updating and improving the software product to ensure usefulness.
    a. Testing
    b. Delivery
    c. Maintenance
    d. Training
15. The individual who uses the software product after it has been developed and marketed.
    a. Project Manager
    b. Project Leader
    c. End User
    d. Tester
16. Select an appropriate waterfall model sequence
    a. Requirements, Feasibility, Design, Implementation, Testing
    b. Requirements, Feasibility, Design, Testing, Implementation
    c. Testing, Implementation, Design, Feasibility, Requirements
    d. Feasibility, Requirements, Design, Implementation, Testing
17. The name of the software life cycle activity that involves teaching end users how to use the software product adequately.
    a. Delivery
    b. Installation
    c. Maintenance
    d. Training
18. The name of the software life cycle activity that involves building the software and converting the design process into proper code.
    a. Implementation
    b. Interface Design
    c. Architectural Design
    d. Code Design

19. The name of the document that describes the order of tasks and necessary time planning
    a. Project Schedule
    b. Source Code
    c. Test Plan
    d. Design Plan
20. The name of the software life cycle activity during design phase that involves determining interfaces in the software system between different parts.
    a. Interface design
    b. Interface plan
    c. Architectural Design
    d. Software Design
21. It is the source of requirement where the requirements will be derived from the environment and the software product will be executed.
    a. Operational Environment
    b. Organizational Environment
    c. Domain
    d. Stakeholders
22. It is the source of requirements where the software engineer have to understand, conform and satisfy the structure and culture of the organization.
    a. Domain Knowledge
    b. Organizational Environment
    c. Operational Environment
    d. Management
23. The name of the term which is used to refer to individual who should have direct influence on the requirements.
    a. Project Manager
    b. End user
    c. Stakeholders
    d. Project Leader
24. The name of the software development process where the engineer takes role for finding conflicts between stakeholders.
    a. Conflict Resolution
    b. Requirements Checking
    c. Domain Understanding
    d. Requirements collection
25. The name of the software development process that is need to find out whether requirements are consistent with stakeholders demands or not.
    a. Requirements Checking
    b. Requirements Collection
    c. Requirements Specification
    d. All of them above
26. The name of the phase which is concerned with demonstrating that the requirements satisfies with the customers wants.
    a. Requirements Validation
    b. Requirements Collection
    c. Requirements Specification
    d. Requirements Checking

27. _____ is the first stage of understanding of the software requirements and also building relationships between development team and customer.
    a. Requirements Specification
    b. Requirements Elicitation
    c. Requirements Collection
    d. Requirements Validation
28. An individual or a group of heterogeneous stakeholders who will operate the software product in different roles and requirements.
    a. Customers
    b. Users
    c. Software Engineers
    d. Market Analyst
29. A group of stakeholders who have commissioned the software product or who represent the relevant target market.
    a. Customers
    b. Users
    c. Software Engineers
    d. Market Analyst
30. These individuals reuse the components of different software products to profit from developing software.
    a. Market Analyst
    b. Software Engineers
    c. Regulators
    d. Customers
31. The responsible job of _____ are using the requirements to plan target marketing bid for the system and planning the software system development process.
    a. System Engineers
    b. System Customers
    c. Managers
    d. System Maintenance Engineers
32. It is an International Standard that establishes a framework of software life cycle processes.
    a. ISO/IEC 12207 Software Life Cycle Processes
    b. ISO 3535:1977 Forms Design Sheet and Layout chart
    c. ISO/IEC 15026 System and Software Engineering Assurance
    d. ISO/IEC 15939:2007 Systems and Measurement Software Engineering Processes
33. It is a document which is developed as an output of user requirements.
    a. Software Requirement Specification
    b. Software Design Document
    c. Validation Testing Plan
    d. Software Testing Plan
34. What can be definition of acquisition process?
    a. The purchase of software service or product
    b. The purchase of market plan
    c. The purchase of necessary stuff
    d. The purchase of Software documentation

35. Which of followings are the job of supplier?
    a. Responding Acquirer's proposal
    b. Singing contract with acquirer
    c. Supplying Software Product
    d. All of them above
36. In which process can Implementation activity be done?
    a. Supply Process
    b. Acquisition Process
    c. Requirements Definition Process
    d. Development Process
37. _____ is the activity which involves getting units of program and combines them in an integrated program with all relations of units.
    a. Testing
    b. Integration
    c. Joint Review
    d. Problem Resolution
38. Which of the following can produce inputs for documentation process?
    a. Developers
    b. Managers
    c. Users
    d. All of them above
39. It is a process of identifying software items, controlling modifications and releases, recording and reporting the status of the items in the software project.
    a. Configuration Process
    b. Management Process
    c. Training Process
    d. Supply Process
40. _____ is about checking the fulfillment of the requirements of the end of every tasks.
    a. Verification Process
    b. Validation Process
    c. Acquisition Process
    d. Development Process
41. _____ Process check whether the final software system or software product matches its desired usage.
    a. Infrastructure Process
    b. Improvement Process
    c. Training Process
    d. Validation Process
42. If a system or software product is transferred from an old to a new operational environment, it shall be ensured that any software product or data product are in accordance with ISO/IEC 12207. This is _____.
    a. Tailoring
    b. Software Retirement
    c. Documentation
    d. Migration

43. _____ is about development and documentation of plan to remove active support by the operation and maintenance organizations.
   a. Conflict Resolution
   b. Requirements Documentation
   c. Training Plan
   d. Software retirement
44. _____ are management, Infrastructure, Improvement and Training.
   a. Organizational Life Cycle Process
   b. Supporting Life Cycle Process
   c. Primary Life Cycle Process
   d. All of them above
45. _____ is the process of arranging/modifying ISO/IEC 12207 Software Life Cycle Processes according to the needs, size, budget, risks, time etc.
   a. Configuration Management Process
   b. Operation Process
   c. Tailoring Process
   d. Development Process
46. The general name of software development methodologies that produce software in short iterations and allows enhanced changes in design.
   a. Traditional Process Models
   b. Agile Model
   c. Waterfall Model
   d. Incremental Model
47. Which of the followings is not an Organizational Processes?
   a. Management Process
   b. Improvement Process
   c. Audit Process
   d. Training Process
48. _____ is small scale organization that enters into a contract
   a. Party
   b. Operator
   c. Service
   d. Task
49. _____ is a group of people and facilities in conformance with an agreement to accomplish particular tasks.
   a. Manager
   b. Facility
   c. Implementation
   d. Organization
50. The process of the evolution of an entity from conception through the retirement.
   a. Monitoring
   b. Firmware
   c. Evaluation
   d. Life Cycle

B.3 Gameplay Scale Survey Data

P	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	SCORE
1	5	5	4	4	4	4	4	4	4	4	5	5	5	4	4	65
2	5	5	3	4	4	4	4	5	4	4	4	4	4	4	4	62
3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	75
4	4	4	3	3	4	4	4	3	3	3	3	3	5	3	3	52
5	4	4	3	4	4	4	4	4	3	4	3	4	4	3	4	56
6	5	5	3	4	4	4	3	4	3	4	3	4	4	3	3	56
7	5	5	5	5	5	4	4	4	3	4	3	3	5	5	5	65
8	5	5	3	4	4	4	4	5	4	5	5	4	5	4	4	65
9	5	5	5	5	4	4	4	4	3	4	3	3	4	4	3	60
10	4	4	4	4	3	4	4	4	3	4	3	3	4	3	3	54
11	5	5	4	4	4	4	4	4	3	4	3	4	4	4	4	60
12	5	5	4	4	4	4	4	4	3	4	5	4	5	4	4	63
13	4	4	4	4	4	5	5	1	5	4	3	4	4	4	4	59
14	4	4	4	4	4	4	4	4	4	4	3	4	4	4	4	59
15	5	5	4	5	4	4	4	5	4	4	3	4	4	4	4	63
16	4	5	4	4	4	4	4	4	4	4	4	4	4	4	4	61
17	5	5	4	4	4	4	4	4	3	4	3	4	4	4	4	60
18	4	5	5	5	4	4	4	5	3	4	4	4	5	4	4	64
19	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	64
20	5	5	4	4	3	4	4	4	3	4	3	4	3	4	4	58
21	4	4	4	4	3	3	3	4	3	4	2	3	2	4	5	52
22	4	4	3	2	3	2	2	4	2	3	2	4	3	4	4	46
23	5	4	4	1	4	1	1	4	1	4	1	4	1	4	3	42
24	5	4	3	2	3	2	2	4	2	4	3	3	4	3	3	47
25	4	3	2	2	3	3	2	2	2	3	3	3	2	4	4	42
26	5	4	3	3	4	2	2	3	3	4	2	2	2	3	4	46
27	4	3	2	3	3	2	3	3	1	4	3	4	1	4	4	44
28	4	4	4	2	3	3	2	3	3	4	2	4	4	3	3	48
29	5	4	4	2	3	2	2	3	2	4	2	4	1	4	3	45
30	5	3	3	2	4	2	2	3	2	3	2	4	2	4	4	45
31	4	4	3	3	4	2	2	2	2	4	2	3	2	4	4	45
32	4	4	4	2	4	2	2	2	4	4	2	4	2	4	4	48
33	4	4	3	2	4	2	2	3	2	4	2	4	3	4	4	47
34	4	4	4	2	4	2	2	4	2	4	3	3	2	4	4	48
35	4	4	3	2	4	2	2	2	2	3	3	3	1	4	4	43
36	2	3	2	2	2	2	2	2	1	3	1	4	1	4	4	35
37	4	4	2	2	4	2	2	2	2	3	2	4	2	3	3	41
38	4	3	3	3	4	3	2	3	2	3	2	3	1	4	4	44
39	4	4	4	2	3	2	2	3	3	4	2	3	3	4	4	47
40	3	3	4	2	4	2	2	4	3	4	2	3	2	4	4	46

B.4 Virtual Learning Session Quiz Data

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Q1	a	a	a	a	a	a	a	b	b	a	a	c	d	a	b	a	d	a	a	a
Q2	c	c	c	b	c	c	c	b	c	c	c	b	b	c	c	b	b	c	c	c
Q3	b	b	b	d	b	b	b	d	b	b	a	a	a	b	b	a	a	b	a	b
Q4	b	c	c	a	b	c	a	c	c	b	d	c	c	c	c	d	c	a	d	c
Q5	d	c	c	a	d	d	c	d	d	d	a	d	d	c	d	d	d	a	a	d
Q6	a	c	c	a	a	c	c	d	d	a	d	d	d	c	d	d	d	a	d	d
Q7	c	c	c	d	c	c	b	c	c	c	c	c	c	c	c	c	c	c	c	c
Q8	d	d	d	a	d	d	a	d	d	d	d	d	d	d	d	d	d	d	d	d
Q9	a	a	a	b	a	a	c	a	a	a	a	a	a	a	a	a	a	b	a	c
Q10	c	c	c	c	c	c	c	a	a	c	b	b	b	c	a	b	b	b	b	b
Q11	b	b	b	b	b	b	b	c	c	b	b	b	b	b	c	b	b	b	b	b
Q12	d	c	c	c	d	c	c	b	b	d	a	a	a	c	b	a	a	c	a	c
Q13	d	c	c	c	d	c	c	d	d	d	c	a	a	c	d	a	a	d	c	d
Q14	c	c	c	c	c	c	c	c	c	c	d	d	d	c	c	d	d	c	d	c
Q15	a	c	c	c	a	c	c	c	c	a	b	a	a	c	c	a	a	c	b	c
Q16	c	c	c	c	c	c	c	b	b	c	c	a	a	c	b	a	a	c	c	d
Q17	d	d	d	d	d	d	d	c	c	d	d	d	d	d	c	d	d	d	d	d
Q18	a	a	a	a	a	a	a	d	d	a	b	a	a	a	d	a	a	a	b	a
Q19	c	b	b	b	c	b	b	b	b	c	a	b	b	b	b	b	b	a	a	c
Q20	a	a	a	a	a	a	a	a	a	a	a	b	b	a	a	b	b	a	a	a
Q21	a	a	a	a	a	a	a	a	a	a	a	c	c	a	a	c	c	c	a	a
Q22	b	b	b	b	b	b	b	b	b	b	b	a	a	b	b	a	a	c	b	b
Q23	c	d	d	d	c	d	d	d	d	c	d	c	c	d	d	c	c	d	d	d
Q24	c	d	d	d	c	d	d	d	d	c	d	a	a	d	d	a	a	c	d	c
Q25	b	c	c	c	b	c	c	c	c	b	b	a	a	c	c	a	a	c	b	c
Q26	b	a	a	a	b	a	a	a	a	b	c	a	a	a	a	a	a	c	c	a
Q27	b	b	c	b	b	c	c	c	c	b	b	b	b	b	c	b	b	a	b	b
Q28	b	b	c	b	b	c	c	c	c	b	b	b	b	b	c	b	b	a	b	b
Q29	a	a	c	a	a	c	c	c	c	a	a	a	a	a	c	a	a	c	a	a
Q30	b	b	d	b	b	d	d	d	d	b	b	b	b	b	d	b	b	a	b	b
Q31	c	c	d	c	c	d	d	d	d	c	c	c	c	c	d	c	c	a	c	c
Q32	a	a	d	a	a	d	d	d	d	a	a	a	a	a	d	a	a	a	a	a
Q33	c	c	a	c	a	a	a	a	a	c	c	a	a	c	a	a	a	c	c	a
Q34	c	c	a	c	a	a	a	a	a	c	c	a	a	c	a	a	a	c	c	c
Q35	d	d	d	d	d	d	d	d	d	d	a	d	d	d	d	d	d	d	a	d
Q36	d	d	d	d	d	d	d	d	d	d	c	d	d	d	d	d	d	d	c	d
Q37	b	b	b	b	b	b	b	b	b	b	c	b	b	b	b	b	b	b	c	b
Q38	c	c	c	c	c	c	d	d	d	c	c	d	d	c	d	d	d	c	c	d
Q39	a	a	a	a	a	a	a	a	a	a	c	a	a	a	a	a	a	c	c	a
Q40	a	a	a	a	a	a	a	b	b	a	a	b	a	a	b	b	a	a	a	a
Q41	d	d	d	d	d	d	d	c	c	d	d	b	b	d	c	b	b	d	d	d
Q42	c	c	c	c	c	c	d	a	a	c	d	b	b	c	a	b	b	d	d	d
Q43	d	c	c	c	d	c	c	c	c	d	d	a	a	c	c	a	a	d	d	d
Q44	a	a	a	a	a	a	a	a	a	a	c	d	d	a	a	d	d	c	c	a
Q45	c	c	c	c	c	c	c	c	c	c	c	d	d	c	c	d	d	c	c	c
Q46	c	c	c	c	c	c	c	c	c	c	c	d	d	c	c	d	d	c	c	c
Q47	b	b	b	b	b	b	b	b	b	b	a	b	b	b	b	b	b	d	a	d
Q48	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
Q49	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d
Q50	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d
S	66	60	52	50	70	54	52	50	54	66	54	62	64	60	54	62	64	52	54	82

B.5 Paper-based Learning Session Quiz Data

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20
Q1	a	d	d	a	d	a	a	b	a	a	a	a	a	c	a	d	d	c	a	a
Q2	c	d	d	c	c	c	c	d	d	d	c	c	c	d	c	b	b	d	c	c
Q3	b	d	d	b	a	b	b	a	b	b	b	b	b	c	b	c	c	a	b	b
Q4	d	a	a	a	c	c	d	a	d	d	c	c	c	d	c	b	c	d	c	c
Q5	a	a	a	a	c	d	a	c	d	d	d	d	d	a	d	a	d	d	d	d
Q6	a	a	a	a	d	d	b	c	c	c	d	d	d	a	d	c	d	d	d	d
Q7	a	d	d	c	d	c	d	c	c	c	c	c	c	d	c	d	c	c	c	c
Q8	c	c	c	d	d	d	a	c	a	a	d	d	d	c	d	a	d	d	d	d
Q9	b	c	c	a	b	a	c	b	a	a	a	a	a	c	a	d	a	a	a	a
Q10	a	b	d	d	b	b	d	c	c	c	b	c	b	c	b	c	b	b	b	b
Q11	b	b	c	d	c	a	c	a	b	b	b	d	b	b	b	a	c	b	b	b
Q12	a	a	c	d	a	d	c	a	d	d	a	c	a	a	a	d	d	a	a	a
Q13	d	d	c	d	c	c	c	a	d	d	d	b	d	d	d	c	a	d	d	d
Q14	d	c	d	c	d	d	d	c	b	b	c	d	c	c	c	d	a	c	c	c
Q15	d	c	d	c	d	a	c	d	c	c	c	d	c	c	c	d	a	c	c	c
Q16	a	d	a	a	a	c	d	b	d	d	d	a	d	d	d	a	c	d	d	d
Q17	a	d	a	a	a	a	d	b	d	d	d	a	d	d	d	a	c	d	d	d
Q18	c	a	c	c	c	d	a	b	a	a	a	c	a	a	a	c	b	a	a	a
Q19	c	c	c	a	c	d	a	c	a	a	a	c	a	a	a	c	b	a	a	a
Q20	c	c	c	a	c	b	a	d	a	a	c	d	a	a	a	c	a	a	a	a
Q21	a	c	c	c	d	a	a	a	c	c	d	a	c	a	a	d	c	a	a	a
Q22	b	c	c	d	a	b	b	b	d	d	c	b	d	b	b	c	d	b	b	b
Q23	d	d	c	d	b	c	c	c	a	a	b	c	a	c	c	b	a	c	c	c
Q24	d	d	a	d	a	a	a	a	d	d	c	a	c	a	a	d	c	a	a	a
Q25	c	c	a	a	b	a	a	a	c	c	d	a	c	a	a	d	c	a	a	a
Q26	c	c	a	a	c	a	a	a	c	c	c	a	b	a	d	a	c	c	a	a
Q27	b	d	b	b	d	b	b	b	c	c	d	b	c	b	c	b	a	d	b	b
Q28	b	b	b	b	c	b	b	b	d	d	c	b	c	b	a	b	d	c	b	b
Q29	a	a	a	a	d	a	a	a	c	c	d	a	d	a	c	a	a	d	a	a
Q30	b	b	b	b	c	b	b	b	c	c	c	c	c	b	d	b	c	c	b	b
Q31	c	c	c	c	d	c	c	c	d	d	c	d	d	d	a	c	c	d	c	c
Q32	c	a	a	a	d	a	a	a	d	d	a	c	c	d	c	a	a	c	a	a
Q33	c	a	a	a	d	a	a	a	b	b	a	c	c	d	c	a	a	d	a	a
Q34	c	a	a	a	d	a	a	a	c	c	a	c	d	c	c	a	a	c	a	a
Q35	c	d	d	d	d	d	d	d	d	d	d	b	c	c	a	d	d	c	d	c
Q36	a	a	d	d	d	d	d	d	d	d	c	b	b	a	a	d	d	c	d	a
Q37	a	a	b	a	b	b	b	b	b	b	a	c	c	c	a	b	b	d	c	c
Q38	d	d	d	d	d	d	d	d	d	d	c	a	a	a	c	d	d	c	c	a
Q39	a	a	a	a	a	a	a	a	a	a	b	c	d	c	d	a	a	c	c	c
Q40	a	a	a	a	a	a	a	a	a	a	b	d	c	c	c	a	a	c	c	c
Q41	d	c	c	c	d	a	c	c	a	a	c	c	a	c	b	d	d	b	b	b
Q42	c	c	c	c	d	a	a	a	c	c	c	c	c	c	b	d	d	b	c	b
Q43	c	c	c	c	a	b	c	c	a	a	a	a	a	b	b	d	d	b	b	b
Q44	c	c	c	c	c	c	d	d	b	b	c	a	d	a	c	a	a	b	c	c
Q45	d	c	d	d	a	d	b	b	c	c	b	c	d	c	b	c	c	b	a	a
Q46	b	b	d	d	d	a	d	d	b	b	c	b	d	b	c	b	b	d	d	c
Q47	c	c	d	d	a	d	b	b	c	c	d	c	d	c	d	c	c	a	b	a
Q48	a	a	d	d	d	b	c	c	a	a	b	a	c	a	b	a	a	c	c	b
Q49	d	a	a	a	b	c	c	c	c	d	a	d	c	d	b	d	d	b	a	b
Q50	d	a	a	a	b	c	c	c	b	d	a	d	c	d	b	d	d	c	a	b
S	44	42	40	48	32	60	46	60	46	50	50	54	40	54	50	50	54	42	72	68

B.6 Assessment Quiz Answer Key

1	A	11	B	21	A	31	C	41	D
2	C	12	A	22	B	32	A	42	D
3	B	13	D	23	C	33	A	43	D
4	C	14	C	24	A	34	A	44	A
5	D	15	C	25	A	35	D	45	C
6	D	16	D	26	A	36	D	46	B
7	C	17	D	27	B	37	B	47	C
8	D	18	A	28	B	38	D	48	A
9	A	19	A	29	A	39	A	49	D
10	B	20	A	30	B	40	A	50	D

Part 3

Appendix C

C.1 T Distribution Table

t table											
cum. prob	t .50	t.75	t.80	t.85	t.90	t.95	t.975	t.99	t.995	t.999	t.9995
one-tail	0.50	0.25	0.20	0.15	0.10	0.05	0.025	0.01	0.005	0.001	0.0005
two-tails	1.00	0.50	0.40	0.30	0.20	0.10	0.05	0.02	0.01	0.002	0.001
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.701	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.32	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
Confidence Level	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%

Part 4

Appendix D

Curriculum Vitae

PERSONAL INFORMATION

Surname, Name: AYDAN, Ufuk

Date and Place of Birth: 4 May 1990, Antalya

Marital Status: Single

Phone: +90 543 302 19 55

Email: aydanufuk@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya Univ., Computer Engineering	2016
B.Sc.	Çankaya Univ., Computer Engineering	2013
High School	Ömer Seyfettin Anatolian High School	2008

WORK EXPERIENCE

Year	Place	Enrollment
2015- Present	Computer Engineering Department of University of Turkish Aeronautical Association	Research Assistant

2012	SOFTEB Yazılım ve Teknoloji Tic. A.Ş.	Intern
2011	Türkiye Kalkınma Bankası A.Ş. Bilgi İşlem Müdürlüğü	Intern

SKILLS and KNOWLEDGE

Programming Languages

- C, C#, Java
- SQL, HTML, ASP.NET

Softwares

- Microsoft Visual Studio
- NetBeans IDE, Eclipse
- ArcGIS, ArcMap
- Unity3D
- Microsoft Visio, Microsoft Project, Microsoft Office

Foreign Language

- English (YDS 2014 : 83,75)

SEMINARS and CERTIFICATES

- Çankaya Üniversitesi "1. Yazılım ve Teknoloji Günleri" (12-13 Mart 2012)
- Silverlight, WPF, LINQ Seminar (4 Ocak 2010)
- "Windows 7" Seminar Bilge Adam (11 Aralık 2009)
- "E- Ticaret ve Web'in Geleceği" Seminar Bilge Adam (10 Aralık 2009)

EXPERIENCES

- 2015 - Present University of Turkish Aeronautical Association - Computer Engineering Department Research Assistant

- 2013 (Senior Project) Doğan Tel Örgü

- Stock Management, Product, Order Tracking and Customer Information System

- 2012 (Summer Intership) SOFTEB Yazılım ve Teknoloji Tic. A.Ş.
  - T.C. Sağlık Bakanlığı Yatalak Hasta Evde Bakım Projesi

- 2011 (Summer Intership) Türkiye Kalkınma Bankası A.Ş. Bilgi İşlem Müdürlüğü
  - C# Seminar
  - Personnel Phone Book Desktop Application

PUBLICATIONS

- Aydan, U., Yilmaz, M., & O'Connor, R. V. (2015, June). Towards a serious game to teach iso/iec 12207 software lifecycle process: an interactive learning approach. In *International Conference on Software Process Improvement and Capability Determination* (pp. 217-229). Springer International Publishing.

INTERESTS

- Software Engineering
- Object Oriented Programming
- Geographical Information Systems and Geoprocessing methods
- Unity3D and arcade games
- Scale model cars, Canoe - Kayak