A METHODOLOGICAL APPROACH FOR SERIOUS GAME SOFTWARE DEVELOPMENT:

AN APPLICATION FOR LANGUAGE DISORDERS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

MEHMET ÇAĞATAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF COMPUTER ENGINEERING

OCTOBER 2012

Approval of the Graduate School of Natural and Applied Sciences, Atılım University

_____

Prof. Dr. K. İbrahim Akman

Director

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

_____

Prof. Dr. K. İbrahim Akman

Head of Department

This is to certify that we have read the thesis "A Methodological Approach For Serious Game Software Development: An Application For Language Disorders" submitted by "Mehmet ÇAĞATAY" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____	_____
Prof. Dr. Pınar Ege	Asst. Prof. Dr. Nergiz E. Çağıltay
Co-Supervisor	Supervisor

Examining Committee Members

Prof. Dr. Pınar Ege	_____
Asst. Prof. Dr. Nergiz E. Çağıltay	_____
Asst. Prof. Dr. Erol Özçelik	_____
Asst. Prof. Dr. Erhan Gökçay	_____
Dr. Gül Tokdemir	_____

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Mehmet Çağatay

Signature :

ABSTRACT

A METHODOLOGICAL APPROACH FOR SERIOUS GAME SOFTWARE DEVELOPMENT:

AN APPLICATION FOR LANGUAGE DISORDERS

Çağatay, Mehmet

M.S., Computer Engineering Department

Co- Supervisor: Prof. Dr. Pınar Ege

Supervisor: Asst. Prof. Dr. Nergiz Ercil Çağıltay

October 2012, 94 Pages

The computer software has been actively used in education area in different ways today. However, for several reasons educational institutions are failing to integrate this software to current educational environments. Educational institutions have been criticized for using technologies similar to the ones used hundred years ago. We believe that, one of the reasons for this failure, integration of educational software technologies into current educational environments, is the complexity of these systems. Hence developing efficient software by addressing the real life problems is a complex process. There are various software development methodologies especially for complex software, with regular and planned development processes. So far, these software development methodologies are appropriate for almost all software, though, in terms of unique needs and developmental process of educational software, they may be inadequate. In other words, development of the educational software process requires some other considerations, such as the domain experts that are not considered during the development process of commercial software projects.

In this thesis, a new educational software development methodology by involving the domain experts and their interactions with the end users is recommended. Additionally, this software development methodology is used in a serious game development process that supports the therapy process of children with impaired

speech and language. Primarily in this study, the contribution of the serious game on the current therapy sessions is evaluated which is developed by using the proposed educational software development methodology. It is aimed to better address the problems of current therapy sessions by developing the software according to the new methodological approach. In other words, this study is a case study to show how the proposed methodology is applied on the development process of a serious game as well as its impact on current therapy sessions.

Keywords: Educational Software, Serious Game, Healthcare Game, Software Development Methodology

ÖZ

CİDDİ OYUN YAZILIMI GELİŞTİRMEK İÇİN BİR METODOLOJİK YAKLAŞIM

KONUŞMA BOZUKLUĞU ÜZERİNE ÖRNEK BİR ÇALIŞMA

Çağatay, Mehmet

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Ortak Danışman: Prof. Dr. Pınar Ege

Tez Danışmanı: Asst. Prof. Dr. Nergiz Ercil Çağıltay

Ekim 2012, 94 Sayfa

Bilgisayar yazılımları hayatın çoğu alanında kullanıldığı gibi eğitsel alanda da aktif olarak kullanılmaktadır. Ancak gerçek hayat problemlerini ele alan yazılım geliştirmek karmaşık bir süreçtir. Bu nedenle karmaşık yazılımların düzenli ve planlı ilerleyebilmesi için birçok yazılım geliştirme metodolojisi geliştirilmiştir. Bugüne kadar bu metodolojiler hemen her yazılım sürecine uygun olsa da, eğitsel yazılımların kendine özgü ihtiyaçları ve gelişim süreçleri açısından yetersiz kalabilmektedir.

Bu tezde eğitsel yazılımların daha sistematik ve temel ihtiyacı olan uzmanlar ve son kullanıcıları içinde barındıran bir eğitsel yazılım geliştirme yaklaşımı önerilmektedir. Ayrıca bu yazılım geliştirme metodu dil ve konuşma bozukluğu olan çocukların terapi süreçlerine destek vermek amacıyla bir ciddi oyunun geliştirilmesi sürecinde kullanılmıştır. Bu tez çalışması ile öncelikle önerilen eğitsel yazılım geliştirme metodolojisinin bir oyun çalışması ile değerlendirilmesi sağlanmıştır. Ayrıca geliştirilen yazılımın konuşma bozukluklarının tedavi sürecine olan etkisi değerlendirilmiştir.

Anahtar Kelimeler: Eğitsel Yazılım, Ciddi Oyun, Sağlık Oyunu, Yazılım Geliştirme Metodolojisi

To my family

&

To my love

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Asst. Prof. Dr. Nergiz Ercil Çağıltay for the endless support of my thesis study and research, for her patience, motivation, enthusiasm, and immense knowledge. I could not have imagined having a better advisor for my thesis study.

My sincere thanks also go to Prof. Dr. Pınar Ege for her expertise, support, expert guidance and offerings for all parts of my study.

Besides, I would like to thank Dr. Gül Tokdemir for her regard, great efforts, helping to improve my thesis work.

I should also express my appreciation to examination committee members; Asst. Prof. Dr. Erol Özçelik and Asst. Prof. Dr. Erhan Gökçay for their valuable time, suggestions and comments.

I want to offer a special thanks to my love and my future wife Tuğçe for her limitless patience, toleration, strength for someone like me and the most excruciating moments she stood by me.

Last but not the least; I would like to thank my family who gave me their love, support and patience.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SDLC	: Software Development Life Cycle
RAD	: Rapid Application Methodology
ESD	: Educational Software Development
ASHA	: The American Speech-Language-Hearing Association
CBST	: Computer Based Speech Therapy
UML	: Unified Modeling Language
HCI	: Human Computer Interactions
ITS	: Intelligent Tutoring System
MDA	: Mechanics, Dynamics and Aesthetics
DPE	: Dual Phase Evolution
ISTRA	: Indiana Speech Training Aid
SIM	: The Speech Illumina Mentor
STAR	: The Speech Training, Assessment, and Remediation System
OLP	: The Ortho-Logo-Paedia
DE	: Domain Expert
EU	: End User

DEFINITIONS OF SOME TERMS

In this study the following terms has been used in the following scope of meanings:

Domain Expert is the instructor, educator, medical doctor, therapist or the one who has the control over the educational or medical treatment environment and is the expert on that specific domain.

Serious Game is a game primarily developed for an educational or treatment purpose.

Educational software is software that aims to help the educators / domain experts and learners / patients to improve the learning or treatment performance.

Childhood Language Disorder: A significant deficit in the child's level of development of the form, content and use of language [1]; a significant deficit in learning to talk, understand, or use any aspect of language appropriately relative to both environmental and norm-referenced expectations for children of similar developmental level [2].

Language Therapy: Efforts to change or eliminate the underlying language problem, rendering the child a normal language learner, one who will not need any further intervention, or attempting to improve the child's discrete aspects of language function by teaching specific behaviors or compensatory strategies [3].

CHAPTER 1

INTRODUCTION

For more than 60 years, software has become an indispensable part of the human life that produces convenient products. Software development processes has continued to be as 'code & fix'. This system works for small size software development projects, however, it is not always that easy to produce complex software systems without any plan and software development methodology. Therefore, today, several software development methodologies are being used for software development process.

Recently, software is created and used for almost all areas such as commercial, social, educational and medical life of human beings. Among those areas software developed to improve cognitive processes in the field of educational and medical environments are relatively new areas. On the other hand, the current software development methodologies are mostly developed by considering the dynamics of the commercial software systems. However the dynamics of the educational and medical software development process is not the same as the commercial ones.

In those systems the domain expert (such as the educators, doctors and therapists) input and control over the software product is so important. In addition to this, the end users of those software products usually require the software to behave according to their individual situations. For example, in educational systems, since each student's educational requirements, background, learning preferences and environment differs in order to improve the level of support of these systems in educational environments; the software should consider those individual situations of the learners. Similarly, medical software that is planned to be used for a therapy session should be adapted according to specific therapy requirements of each patient. Hence the main difference between these two groups of software systems is the need for 'domain expert(s)' and 'special end user' involvement and requirements. Traditional methodologies usually do not consider these requirements. Hence, a new

methodological software development process is required for these software products. 'Domain Experts' and 'special end users', who do not included in traditional software development methodologies, are the most important participants for educational and medical software development process that need to consider cognitive processes of the end users.

Since last a few years, people play games not only for entertainment, but also for certain 'serious' topics; education, training or health treatments. These games are named with oxymoron words; 'Serious Games'. General usage of these games is to untie problems in human life. A serious game is a computer game that uses fun as a tool and its main purpose is training, educating or exercising [4].

As McLuhan says, "Anyone who makes a distinction between games and learning doesn't know the first thing about either." ~Marshall McLuhan. One of the major problems of human life is health problems. Exclusively, serious games developed for medical purposes have been successfully using in medical training and therapy. Fun and repeated features of games increase motivation and participation, decrease risk. However, the problem is designing process of these games; in order to better address the treatment purposes these games should be well considered and created with accompaniment of a professional.

In this study, a new software development methodology is proposed for educational and medical informatics software development processes and as a case study the proposed software development methodology is used for development of a 3D healthcare game. Proposed methodology involves 'domain expert(s)' and 'special end users' to develop software as different from other methodologies. The results of this study are conducted from two dimensions. First, the results are conducted from the data collected by the application of the proposed software development methodology in the case study. Secondly, the results are conducted according to the impact of the developed game-based software environment on the support of therapy sessions for speech disorder children.

This thesis is organized as follows: Chapter 2 covers background study of the thesis parts, Chapter 3 describes the research methodologies for this study, in Chapter 4 it is clarified the case study system is described by considering the 3D healthcare game for speech disorder children and used tools in detail, Chapter 5 details the results

provided from the patients after applying the system with therapists. Lastly, in discussions and conclusions of the study is provided.

1.1 Scope and Purpose

Technological developments allow looking to sectors from different perspectives. This point of view reverberates to health sector, too. Technological solutions are providing several advantages of serious games nowadays in the health sector. Expanded improvement has started to help in treatment, training and therapy. For instance; it is hard to motivate children who have speech and language problem during therapy sessions. Therapists use many different tools such as; toys, pictures and stories to handle this problem. In the literature, there are not many studies using serious games approach to support the therapy process of speech and language disordered children. With this serious game project, we intended to increase motivation and make therapy more enjoyable and also relieve the therapist from paper based therapy séances and record keeping. This thesis focuses on healthcare serious games which can also be considered as educational software for speech and language disordered children. To develop the game, a new methodology for software development life cycle (SDLC) is followed which includes necessary participants, domain experts, learners, and patients, to create educational software.

1.2 Limitations and Assumptions

Effective speech and language therapy takes weeks, months and sometimes years. With this limitation, this study only looks at few therapy sessions' data set, to better understand the objective functions of developed product from the domain experts' and patients' perspective. Besides, finding appropriate learners or patients was not easy to test the system. Furthermore, the serious game environment is developed as a prototype hence it has some limited functionality and graphical support.

Despite these deficiencies are be in evidence, this thesis is based on the assumption that the proposed educational software development methodology may beneficial for many educational software development process. Additionally, results of this study show that, even the above limitation of this study, there is evidence that, the

developed serious game environment for speech disorder children can be used and advance in therapy sessions. Further empirical studies need to be conducted to better understand the effect of the developed environment on the performance of therapy sessions and the effect of the proposed methodology on the success of this achievement.

CHAPTER 2

BACKGROUND OF THE STUDY

The purpose of this study is two-fold. The first aim is to address speech and language therapy problems by using serious games technologies. Second, it proposes a methodological approach for the game design and development process specific to such software environments. In that concern, this chapter first summarizes speech and language disorders. The second part of the chapter summarizes serious games applications and technologies which are also used to address some medical problems such as speech disorder. In the last part of the chapter, classical software development methodologies are discussed. Additionally, their limitations for developing such a software development process are discussed.

2.1 Communication, Language and Speech

Like vital requirements such as food, shelter and protection, human beings need communicate with each other. This need is observed from birth to death. For instance; a baby cries when it wants to say 'I am hungry' or 'I am not comfortable because of my clothes'. This shows that communication has many forms [1].

Many times, 'language' and 'speech' are used instead of 'communication'. This mistake is based on relationship between communication, language and speech. In fact, all of these three notions are different from each other and each one has different features. For instance; a person who has a cleft lip or palate might have no problems about using verbal language. The main problem would be in transferring the sounds of the language to acceptable acoustic forms. To show the differences, it would be appropriate to start from meaning of these words in the literature:

Communication: "the various methods of sending information between people and places, especially official systems such as post systems, radio, telephone, etc." [5]

Language: “a system of communication consisting of sounds, words and grammar, or the system of communication used by the people of a particular country or profession” [5]

Speech: “the ability to talk, the activity of talking, or a piece of spoken language” [5]

Communication is a general term which includes speech and language and also nonverbal others such as sign language, gestures and facial expressions. Language and speech are not the same thing. Language is a set of rules for different aspects of language learned by the speakers of the language. While languages share some universal standards, each language also has its own, specific set of rules. Speech, on the other hand, is auditory (acoustic) or visual manifestation of language, which requires an intact vocal mechanism. A person may know (understand) the rules of the language but may not be able to transfer it to acoustic messages due to neurological and/or structural anomalies.

Language and speech are important tools for inter-individual communication. In some cases, the use of language and speech defects occurs. These defects which have various types and sizes affect daily lives of individuals. This is called a communication disorder. Communication disorders are classified into several forms.

2.2 Communication Disorders and Classification

Sometimes people, especially children, have difficulties speaking because they do not know the language rules. And sometimes, even they know the rules, because they have problems about speech organs, they have problems about speech. All types of communication disorders are shown in Table 2.1. Generally communication disorders can be classified into two main types: Language Disorders and Speech Disorders. And further classification is made regarding problems of reception (understanding) and expression (speaking).

Table 2.1 - Classification of Communication Disorders [6]

<table>
<tr><th>Reception</th><th>Expression</th><th>Etiology</th><th>Time of Onset</th></tr>
<tr><td rowspan="2">Hearing Acuity
• Conductive
• Sensorineural
• Mixed

Central Auditory
• Processing:
• Decoding
• Integration
• Organization
• Understanding speech under adverse conditions
• Short-term memory
• Multiple categories</td><td rowspan="2">Speech:
• Articulation
• Fluency
• Voice

Language:
Form
• Phonology
• Morphology
• Syntax
Content
• Lexicon
Use
• Pragmatics</td><td rowspan="2">Neuromotor abnormalities
Hearing impairment
Environmental / learning factors
Cognitive deficits
Anatomical or physiological impairments</td><td>Congenital
Acquired</td></tr>
<tr><td>Severity
Borderline
Mild
Moderate
Severe
Profound</td></tr>
</table>

2.2.1 Speech Disorder

A person with a speech disorder may have difficulty with articulation, fluency, voice or any combination of these. Anatomic problems in the organs which are about speech cause the speech disorder. For instance: cleft palate/lips and structural defects on teeth. There are three types of speech disorders:

Articulation Disorder: Although current literature differentiates between phonological (developmental difficulty in learning the sounds of the language - a language problem) and articulation problems (difficulty in producing sounds due to organic reasons - a speech problem) both have traditionally been included under the term ‘articulation disorders’. Articulation problems are mainly associated with problems involving the speech system and motor aspects of speech production [7, p. 144]. Otto defended that articulation problems can also reflect specific physical destruction, such as cleft lip, cleft palate, or tongue-tie, which is known as limiting lingual frenulum. All these problems come from abnormalities that occurred during prenatal development [8].

Some common causes of articulation disorders

- Tooth alignment and missing teeth
- Impaired oral-motor skills
- Eating problems
- Tongue thrust swallow after 6 years of age
- Neurological disorders
- Mental retardation
- Cleft lip and/or palate

Fluency Disorder: Abnormally stoppages that interrupt the flow of communication and are inappropriate for a person's age, culture, and linguistic background. The best known fluency disorder is "stuttering". While speaking, stutterers use repetitions, prolongations and blocks. There are many theories about the cause of fluency disorder, but the exact cause is still unidentified. It may be genetic or might involve brain functions that control speech production. Some facts about fluency disorders are that:

- it generally begins between the ages of 2-5
- it may start suddenly or gradually
- involve as more girls than boys
- it can vary in severity over time

Voice Disorder: It can be identified as the lack of voice or presence of unusual pitch loudness, quality or period. Voice Disorder as commonly grouped based on their cause, an etiological classification. It can be divided to four categories:

- Voice abuse
- Neurogenic disorders
- Psychogenic disorders
- Alaryngeal communication

2.2.2 Language Disorder

As mentioned above, language consists of rules regarding its five components. These components are grouped by Bloom and Lahey [9], in three main categories: form, content and use. These five components are:

Form

- **Phonology** is the component about the sound system of the language. It deals with how speech organized [10, p. 3].
- **Morphology** is a sub-field of linguistics focusing on how words are composed [10, p. 61].
- **Syntax** is the study of sentence phrase structure that makes a sentence [10, p. 163].

Content

- **Semantics** is the study of the meaning of words or sentences of the language [11].

Use

- **Pragmatics** is the study of the use of language utterances with respect to their contexts, which also includes rules of discourse [12].

A language disorder is described by ASHA as "impaired comprehension and/or use of a spoken, written, and/or other symbol system. A language disorder may involve all of the components of language mentioned above, namely: (1) the form of language (phonology, morphology, and syntax), (2) the content of language (semantics), and/or (3) the function of language in communication" (ASHA, 1993, p.40):

Three important features of this disorder are: etiology, manifestation and severity. Some common causes of language disorders are [13]:

- Specific language impairment (SLI)
- Autism spectrum disorders
- Mental retardation and other congenital syndromes
- Hearing impairment
- Traumatic brain injury and other accidents
- Others

2.2.3 Children with Language Disorders

A child starts acquiring the structure and fluency of the language from birth. Children have usually acquired the rules for the components of language (Phonology, Syntax, Semantics, Pragmatics, and Morphology) by age five. Acquisition research has determined which rules are acquired at which age and at what order. If this process is delayed or distorted, a language disorder might be taking place [14].

In Turkish is an agglutinative language belonging to the Altaic language family in which meanings and grammatical relations are expressed by attaching bound morphemes to the roots of words, mostly nouns and verbs. For example, case morphemes (dative, accusative, locative and ablative) are attached to nouns to indicate direction (eve), position (evde), etc. Or, verbs take morphemes to indicate negative, tense, question, and the like (ol-ma-dı-mı). This brings to fore the importance of morphology and the concept of morpheme in Turkish [15][16].

Morphology is the study of the fundamental structural blocks of meaning in language. These blocks are called morphemes. A morpheme is the smallest semantically meaningful and grammatical functional piece in a language. A morpheme does not essentially have to be a word. Some morphemes can stand alone as a word without another morpheme. Those are called free morphemes, such as cat, car, boy and. Bound morphemes, on the other hand, are always attached to free morphemes and are used as prefixes, infixes and suffixes, as in the Turkish examples mentioned above [17].

Turkish children start learning the morphemes of the language at an early age [18]. Language learning problems in Turkish quite often manifest themselves as difficulties in learning to use the suffixes, which are crucial in conveying meaning in Turkish. This study, therefore, it is focused on developing a prototype computer program to teach cases of nouns and past, present continuous and future verb suffixes to children with language disorders.

2.3 Speech and Language Therapy Materials

Therapists use several therapy methods and materials for treatment. These materials are sometimes paper-based, sometimes material based and more recently computer-based approaches are being used in therapy.

2.3.1 Paper-Based Therapy Materials

Speech and language disorders generally occur at early ages. This rate is high between 3-11 years [19, p. 204]. When doing speech therapy with children, motivation can be an impediment. Thus, therapists use many kinds of techniques during sessions (Figure 2.1). As each patient with a speech disorder is unique, therapy will have to be tailored to the needs of the individual.

Figure 2.1 - Sentences for Easter Egg Basket

During therapy time, therapist uses some paper-based equipment commonly such as pictures, printed papers, real objects, games, flash cards, toys, puppets and reinforcers of all sorts to keep sessions active and comfort children to work harder (Figure 2.2). For each session, therapists need to prepare the therapy environment; find materials, tests, even toys to not to lose motivation of patient. This process is hard and onerous [20, p. 2].

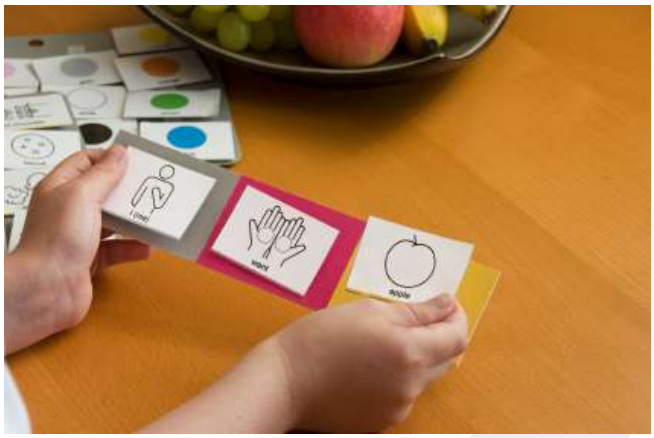

Figure 2.2 - A Paper-Based Speech Therapy Material

2.3.2 Computer Based Therapy Materials

Games are significant tools for children. At play times children feel relaxed as and free. During the last decade, these tools have begun to be used in speech and language therapy sessions. The technological progress affected speech and language therapy in two major ways [21, p. 52]:

- Computers and communication tools have positive effects on the development of speech disorder treatments,
- Computer-based software helps speech therapy sessions.

Researchers defend software usage in speech and language therapy because it [22, pp. 53-59]:

- facilities to speech and language disorders during sessions,
- provides feedback as audiovisual,
- provides various types of practical exercises for patients

Computer Based Speech Therapy Materials

Until recently, therapy games were generally basic software applications. These applications have few or no interactions, no successful visuals to influence patients especially children (Figure 2.3). Although, these kinds of games are also beneficial for therapy session, they need to be much more attractive and require an efficient usage.

Figure 2.3 - A Speech Therapy Game (Stuttering)

Studies show that computer games have a major role for speech and language disordered children. This approach allows choosing different exercises for different levels of problems. In general, research indicates a helpful game for therapy session should include feedback, controllability, curiosity and a feeling of competence [23, p. 25].

As to a study on stutters, a prototype game is created by researchers. The game supplies speech motor training. The patient controls the motion of an antagonist figure using his/her voice. The aim is hitting targets on the screen. The patient must use the correct speech pattern in terms of rhythm and continuousness to achieve (Figure 2.4). As a result, it is shown that computer based therapy is more effective than classical therapy [24, p. 3] (Figure 2.5).

Figure 2.4 - A Prototype Game for Speech Therapy for Stutters [24, p. 3]

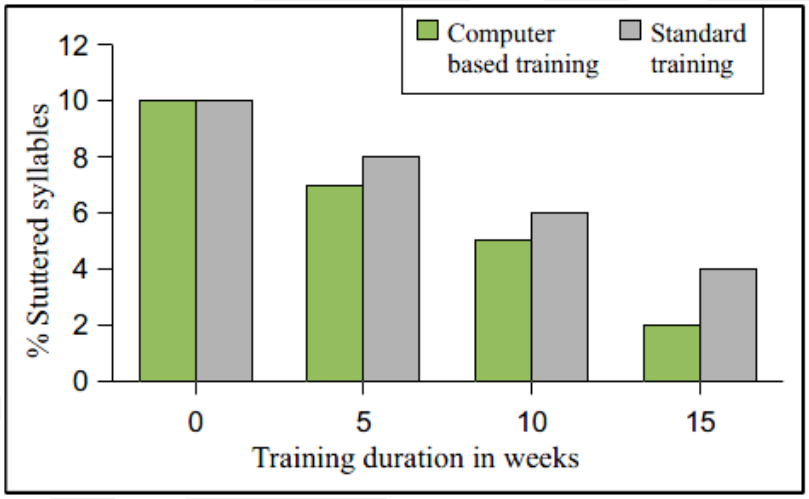

Figure 2.5 - Exemplary Fluency Measurements: Percentage of Stuttered Syllables over Time [24, p. 3]

All produced therapy games are not assumed to replace therapy session or not to be a substitute for therapist or clinicians. Speech problems need therapists' view because there are different types of problems and each problem does not have the same treatment method. It is the therapist that needs choose the method that will be performed. Therapy games can only be supportive tools for therapists. One of the best accomplished examples is the Logomon system [25]. In this system, classical modules and management system for therapist are used (Figure 2.6).

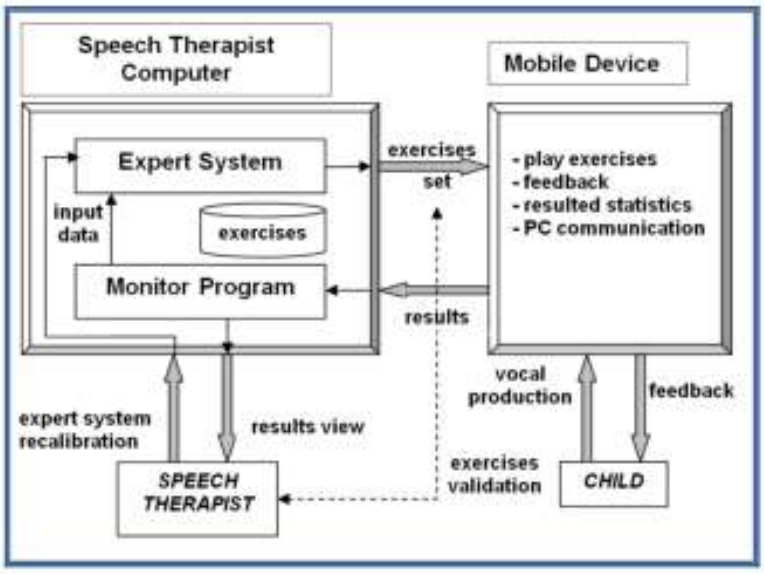

Figure 2.6 - Logomon System Architecture [25, p. 2]

Logomon system is conceived in order to assist the physical therapist. It has been proved to be useful in speech therapy during treatments. There are many software and speech therapy games that are effective which are created on Logomon system.

Computer Based Language Therapy Materials

As opposed to the computer based speech therapy materials, computer based language therapy materials are scarce. Available ones mostly focus on teaching vocabulary (Figure 2.7), or remediating cognitive functions (Figure 2.8). One reason for this scarcity is that language -specific systems are needed. None exist for Turkish.

Figure 2.7 - *Wild, Woolly, Wonderful Critters* game from LocuTour's *Basic Words for Children* [26]

Figure 2.8 – A game which includes choice exercise using animals, verbs, and contrast words such as, over/under, happy/sad [26]

2.4 Educational Software

Educational software can be developed based on different pedagogical approaches in the form of computer based instructions; computer based educational materials or serious games. As educational software, serious games addresses motivation problem of the learning process. Hence studies show that by better motivating the learners the learning performance can be improved through educational games. Therefore, in this study, serious games are researched.

2.4.1 Serious Games

The term serious game used first time 40 years ago by Clark Abt. His book is called Serious Game and it is published in 1970 [27]. He described his work in which he examined war games, simulations to train managers, students, and teachers in the educational program development, industrial management and technological planning and forecasting [27]. His definition of such games was that they "have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. This does not mean that serious games are not, or should not be, entertaining." [27, p. 9]. The Abt's definition was not for computer games, but it can be extended to serious games on computer as well. With this advance, 'serious games' term has acquired many definitions:

"[a serious game is] a mental contest, played with a computer in accordance with specific rules, that uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives." [28, p. 26].
Another definition is:

"games that do not have entertainment, enjoyment, or fun as their primary purpose" [4, p. 21].

Like these, 'serious games' has had many other definitions. However, almost all have common grounds; education, training, advertisement, simulation, investigation, informing and supporting [4].

Serious games have more than just story, art, and software. A serious game involves pedagogy: activities that educate or instruct, thereby imparting knowledge or skill. This addition makes games serious. Pedagogy must, however, be subordinate to story; the entertainment component comes first [28, p. 26].

Serious games have not come up extensively until Serious Games Initiative was constituted in 2002 at the Woodrow Wilson Center for International Scholars in Washington, D.C. With this organization, serious games expanded to many different sectors, even it has started to be offered in academic programs; Michigan State University – Serious Game Design.

Because of the popularity of computer games among young people, educators have started to approach computer games as educational settings. Actually, using games for learning, training, education dates back a long time. Fred Jane produced a tool for military in 1903, named as 'Naval Wargame', in the book 'Little Wars' was about using miniature tokens for military and education, was published by H. George Wells in 1913. Training with board and card games continues with digital games for commercial purposes. Internet was the growth in the 1990s that provided opportunity to spread the games in the digital environment. Computer video games for non-entertainment purposes were developed long before the edutainment, combination of education and entertainment, era, however, and as edutainment failed to prove profitable – and technical advancements in providing realistic settings grew, and multiplayer gaming developed – the concept of serious games was re-examined during the late 1990s [29, p. 2]. Although the current wave of serious games appears to begin in 2002, many games were designed for serious purposes before this date [30, p. 28].

Games can provide more efficient learning environment within class activities [31]. Games are not only fun tools, but also they supply defiance, an aim to finish. If people, not only children, carry out an activity that feels like playing a game, they enjoy it much more and also the activity ensures more permanent learning [31]. However, design and creation of serious games are not as simple as a normal computer game. In common games, fun is the main aim of the game design. On the other hand, serious games do not use fun for their main aim; they use it as a tool to

reach their target population. The simplest definition of serious games is games that do not have entertainment as their primary purpose [4].

Difference between Entertainment Games and Serious Games

Entertainment games and serious games are similar from many points of view. They use the same technology and sometimes the same avatars. Since they have such similarities, what could separate 'serious game' term from entertainment game?

Since the creation of digital games, there has been much discussion trying to separate serious games from leisure games. Zyda described game, video game, and serious game in his article [28, pp. 25-26]. Actually these definitions explain the difference between serious game and leisure game.

Game: "a physical or mental contest, played according to specific rules, with the goal of amusing or rewarding the participant." [27, p. 5].

Video Game: "a mental contest, played with a computer according to certain rules for amusement, recreation, or winning a stake." [28, p. 25].

Serious Game: "a mental contest, played with a computer in accordance with specific rules that uses entertainment to further government or corporate training, education, health, public policy, and strategic communication objectives." [28, p. 38].

Michael and Chen argue the differences between these game types from a design and development viewpoint; the hardware for serious games is less than optimal in markets as to entertainment games, because their market is older than entertainment games, so they need wide variety of hardware and operating systems [4, p. 30]. The result of these structural differences can be summarized in Table 2.2;

Table 2.2 - Differences Between Serious Games And Entertainment Games

	Serious games	**Entertainment games**
Task vs. rich experience	Problem solving in focus	Rich experiences preferred
Focus	Important elements of learning,	To have fun
Simulations	Assumptions necessary for workable simulations	Simplified simulation processes
Communication	Should reflect natural (i.e., non-perfect) communication	Communication is often perfect
	Need learning techniques, experts about topic and technologic background	Anything can be use for fun
	No need scoring system or sometimes has for what is good what is bad	Has Scoring system to compare players

In spite of some similarities, these two game types also have significant differences from each other which designers and developers must not fail to notice [4, p. 30]. Serious games must be regarded about the accurateness of simulated and present thing for the player.

Serious Games for Speech Disordered Children

Speech disorder therapists don't use the old clinic methods anymore because in the last decade there has been an improvement with regard to using speech technology for teaching of pronunciation, speech therapy and researches. With the software and hardware developments, the principles and technologies are utilized for designing computer games. Thus, the serious games have been used for the treatment of speech disorder. As various disabilities are being treated by planned and utilized computer techniques, some issues are taken into consideration [32]:

- age and personality,
- the level of mental processes and language development,
- representation specificities,
- the sense of observation,
- generalization and abstractization possibilities,
- the children's ability to perceive and interpret visual,
- auditory stimuli

As Bogost et al. [33] reports, there are many advantages of using computer systems in the speech disorder therapy. Patients can easily access to therapy exercises by using current technology in their own homes, so at all hours of the day and night computer based therapy exercises can be used according to the patients' wishes. An interaction with therapy software is more dynamic and reactive compared with the paper-based exercises. Furthermore, the patient's performance can be scored dynamically by means of a computer-based system and with the results of performance the therapist can make inference and interpret the findings of the therapy [34].

Computer-based speech training systems (CBST) don't address all therapy requirements. An important handicap is that using CBST generally needs help of a trained staff. On the other hand, the general belief is that computer-based speech training systems (CBST) are useful tools in speech training [25].

During the game development process for speech therapy purposes, the developers need some tools to create theories and methods for [28, pp. 28-30][35, pp. 226-229]. These tools can be summarized as:

- Modeling and simulating computer characters, story, and human emotion;
- analyzing large-scale game play;
- Innovating new game genres and play styles; and
- integrating pedagogy with story in the interactive game medium

Nickerson & Stevens developed the first computer-based speech training system [36]. With the development of the micro- computer method created more advanced-aided speech training programs. Thus, speech and hearing impaired people have an increasing possibility to improve themselves. Mostly, these systems include a microphone, an amplifier, and a speaker connected a sound card. This provides opportunity to the users to input, store and analyze speech by scanning and playing it over. There are some interactive programs in the software for assisting speech and hearing impaired children in achieving awareness. It also helps to gain control over speech attributes such as voicing, timing, pitch, and loudness as well as refining articulation and prosody. All these applications are made successfully thanks to the software [36].

Computer-based speech training systems have started to be widely used in speech training for children with a hearing or speech impairment. There are commercial available systems such as; SpeechViewer and Box of Tricks. Both are widely used and accepted in speech training are good examples of system. The first one developed by IBM in the 1980s and the other one was developed in the EU-project SPECO completed in 2001 with participating universities from Sweden, England, Hungary and Slovenia. Other examples are the Indiana Speech Training Aid (ISTRA), the Speech Illumina Mentor (SIM), the Speech Training, Assessment, and Remediation system (STAR), and the OLP-method [37].

The system SpeechViewer developed by IBM in the 1980s enable people to observe their voices produce a phoneme, and they have a chance to practice speaking by matching their voices with a model phoneme. In the activity if a student pronounces a phoneme correctly, a snail can move closely to the top of hill on screen. Children move from easy to difficult activities that start individual phonemes and go on teaching two phoneme contrasts and multi phoneme chains. At the end of the activity statistical inference shows progress in children [38].

Children use seven visual voice tools in the program for adjusting finely their voices and these tools guide children in many aspects of voice by starting the activity with consciousness of easy sound. Firstly, every tool takes voice input and then it gives visual feedback showing in the graphical form.

For an interaction designer, the most important point is to focus on what is technologically possible instead of what is important for the users in the development stage of CBST systems. This understanding makes systems more advanced and useful rather than not easy enough to use. In human-computer interaction (HCI), user involvement is a significant way to be certain that the system is developed as appropriately according to the subjects, the user and his/her environment. Actually, in the designing process the user involvement is considered on a preferential basis and it should go on until the system is finished [37].

The Ortho-Logo-Paedia (OLP) project manages to provide a method to supplement speech therapy without replacing. It is applied to speech disorders at the articulation level by gathering an integrated computer-based system together with automatic

speech recognition and distance learning. The items mentioned below are key positions of this proposal [39, p. 45]:

Therapy is based on real-time audio-visual feedback of client's speech productions, while therapy sessions are designed for each speech disorder separately and tailored specifically for each client;

Speech production Evaluation and interfacing to assistive technology is provided through automatic speech recognition based on statistical models of the data collected in the training/therapy phase. Web services provide remote collaboration and data collection for analysis and evaluation in diverse conditions.

These features correspond to the major system components: Optacia, Grifos, and Telemachos. Each of them has own characteristic features. Optacia is acoustic- to-articulation kinematic mapping in 2D or 3D and the therapist designes a map. Grifos is a speaker dependent, small vocabulary, automatic speech recognition system. Data related to training is collected during optician sessions, supplementing existing databases of material from same client cases. Telemachos uses distance learning principles with web database technology to provide the system's remote tutoring and monitoring ability [39, pp. 45-46].

OLP system contains articulation disorders represented by users and it supplies with correct evaluation and validation of the materials. The goal of system is to create speech therapy for speakers with dysarthria. Producing a sound at a syllable or word level helps the continuity of production so that some speakers might approach intelligibility of speech and normal speech as well. On the other hand, the continuity of pronunciation allows others to use speech technology applications efficiently. These applications are dependent on speech recognition so it requires a consistent pronunciation. People who have other types of articulation disorders make use of the improved materials and the system can easily be adapted to other languages.

Serious Game Design

Serious Games are defined as digital games and equipment with an agenda of educational design and beyond entertainment [40, p. 559]. As seen from Figure 2.9, the domain expert input to create "meaning" and "reality" in a serious game is a very

important design issue. Serious games are designed for the purpose of solving a problem. Although serious games can be entertaining, their main purpose is to train, investigate, or advertise. Sometimes a game will deliberately sacrifice fun and entertainment in order to make a serious point [41]. Whereas video game genres are classified by game play, serious games are not a game genre but a category of games with different purposes [42]. This category includes educational games and advergames, political games, or evangelical games. The category of serious games for training is also known as "game-learning" [43].

Figure 2.9 - Serious Game Contents [44]

Fun is not the primary power of the design. Training or education becomes the main goal. In Figure 2.10, the heart of serious game design is resolved to schema:

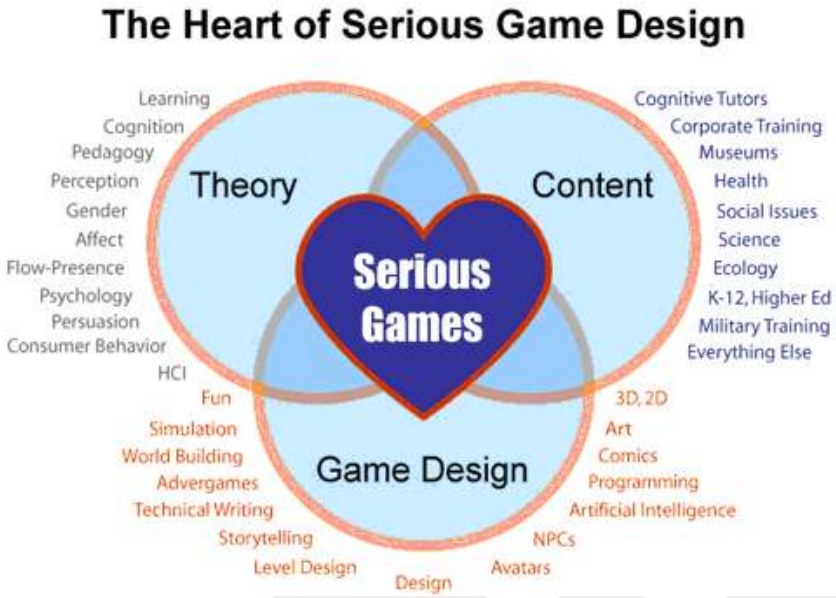

Figure 2.10 - Winn's Serious Game Design [45]

As seen from Figure 2.10, not only game design, but also theory, learning, psychology, pedagogy, gender and content of the game are needed to create a serious game. Lack of any of these factors prevents one from achieving their goal. It should also be noted that, in the Win's game design approach, the involvement of the domain expert for providing this information in the game design is critical. As seen from Figure 2.11, Wijers also shows the components of a serious game design as "play", "meaning" and "reality" [46].

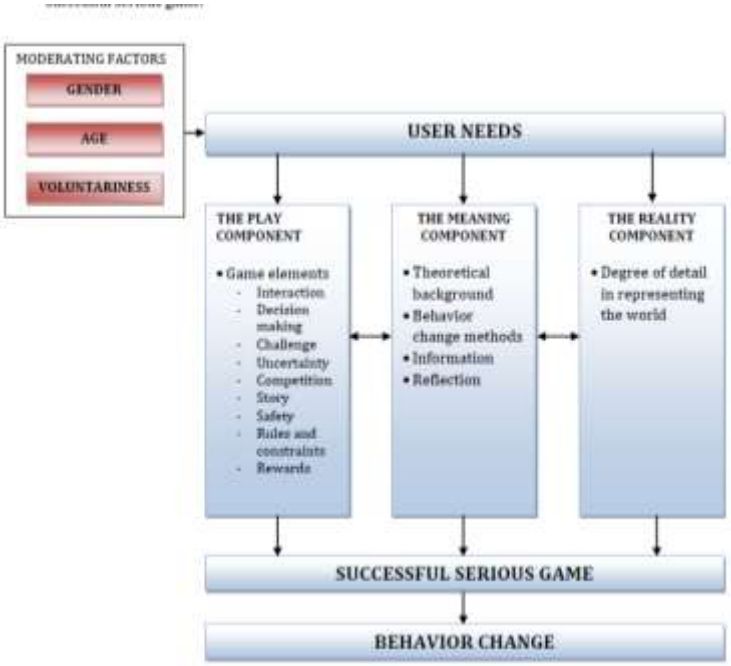

Figure 2.11 - A Model of Serious Game Design [47]

Figure 2.12 shows that serious game design stages are defined in seven levels; specifying the pedagogical objectives, choosing the serious game model, designing the scenario, searching for reusable software components, controlling the pedagogical quality, detailed scene and human computer interactions (HCI) specifications.

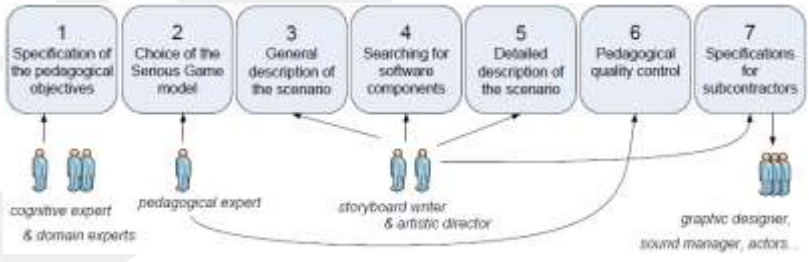

Figure 2.12 - The Seven Steps for Designing Serious Game [35, p. 4]

As seen from Figure 2.12, in this approach during the serious game design, the importance of cognitive experts and pedagogical expert, which are defined as domain experts in our study, are highlighted.

Specifying the Pedagogical Objectives: game development will provide the tools to create theories and methods for developers [28, p. 29];

- Modeling and simulating computer characters,
- Story, and human emotion,
- Analyzing large-scale game play,
- Innovating new game genres and play styles,
- Integrating pedagogy with story in the interactive,
- Game medium

Serious games present many opportunities to learn, however only if the application is designed efficiently. It is so difficult, because rather than simply trying to optimize the entertainment part of the game, or the amusing issue, one must also optimize to achieve a specific set of serious outcomes.

There are some design types to develop serious game for learning. Intelligent tutoring systems (ITS) are one of them. ITSs are computer programs that supply to help tutors that know what they teach, who they teach and how to teach. The design and development of such tutors lie at the intersection of computer science, cognitive psychology and educational research; this intersecting area is normally referred to as cognitive science [48, p. 252] (Figure 2.13).

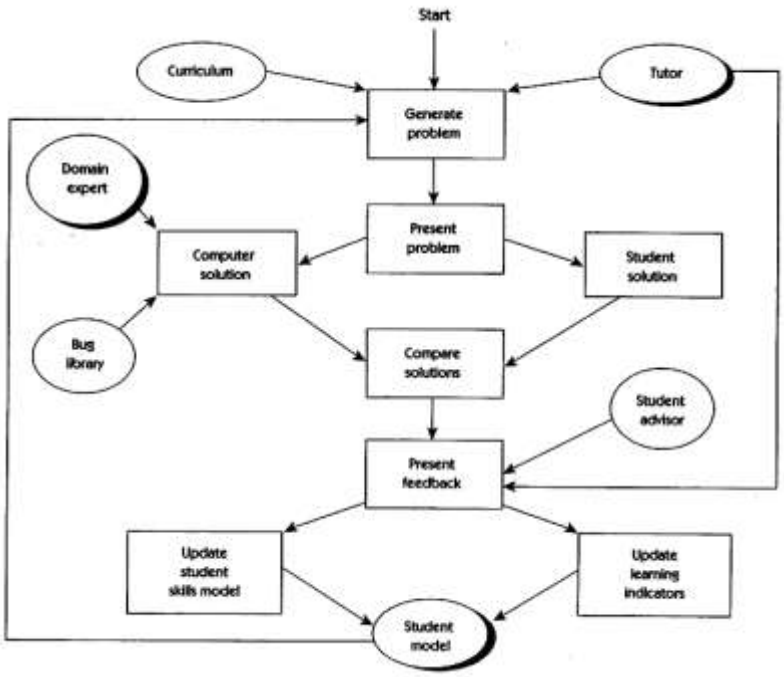

Figure 2.13 - Intelligent Tutor System Model [48, p. 252]

Intelligent tutor system model forms from five components: an expert model, a learner (student) model, a tutor model, a user interface and an authoring system (Figure 2.13). The expert model supplies the expertise part of the all intelligent tutor system [49]. The student model keeps a knowledge base about users and their action on system up [49]. Pedagogical expertise refers educational actions based on the student model [50, p. 854]. This structure is shown in Figure 2.14.

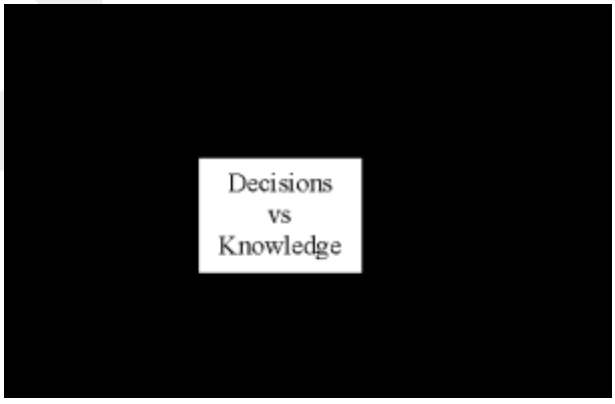

Figure 2.14 - Components of Intelligent Tutor System Architecture [50, p. 854]

There are some frameworks created to organize the game models. One of these frameworks is MDA (mechanics, dynamics and aesthetics). This framework was designed by Marc LeBlanc. The MDA framework indicates the relationship of the designer and the player. The designer designs the mechanics of the game. These rules are adduced at play time and affected by the player's inputs, forming the dynamics, or run-time behavior of the game. The emotional responses when the player is playing are also occurred in the aesthetics part of the framework [45, p. 1013].

Another framework is DPE which design, play, and experience. This framework was created as an improved version of the MDA framework. The DPE framework offers a language to discuss, a methodology to analyze, and a process to design a serious game for learning. DPE is alike to MDA framework. It delineates the relationship between the designer and the player, too [45, p. 1014].

As seen from different serious game design approaches, the contribution of the domain experts on all of them has been addressed explicitly. Now we will examine the software development methodologies from the serious game design perspective.

2.5 Software Development Life Cycle (SDLC) Methodologies

Software projects are big and complex. Thus, many approaches and designs are observed. Nowadays there are several different methods and techniques used in the life cycle of a software development project and most methodologies are adapted to the generic methodologies. Each is planned for a particular reason, most have similar goals and share many common tasks. Especially timing is a very considerable part of a software project and these methodologies help to arrange software development life from begin to end. Classic life cycle methodologies are summarized in the following sections. Software development processes are executed according to appropriate software development methodologies.

2.5.1 Waterfall Methodology

This is the earliest methodology for software development. It was developed in the early 1970's. Its name comes from a diagrammatic schema which resembles a waterfall. This methodology was accepted for years, but recently it's seen as strict and delusive. This methodology consists of with five steps (Figure 2.15):

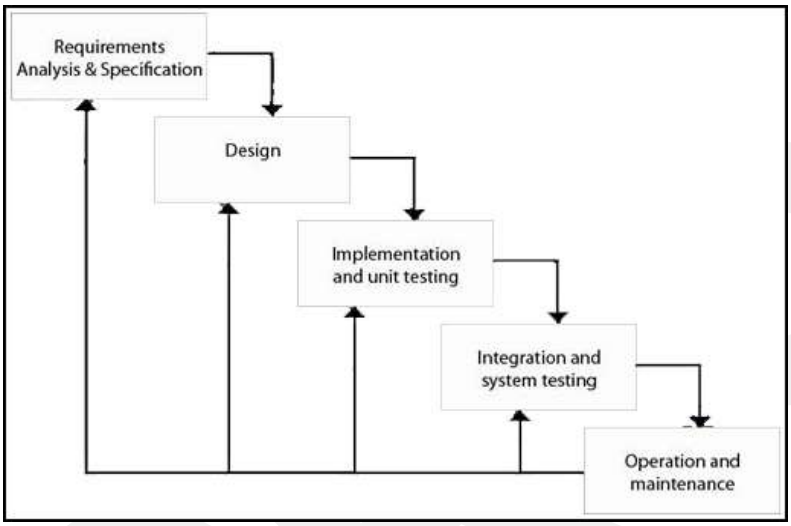

Figure 2.15 - Waterfall Methodology [51]

Requirements: System needs, information, functions, performance and interfaces features are determined in this step. Project managers and stake holders focus in this phase.

Design: System design is created from the results of the first step. Architecture, hardware and software, communication, UML are prepared in this step.

Implementation: Software is coded as to design phase deliverables. The longest step of SDLC is this phase for waterfall methodology. The main part of the cycle is here for software developers.

Integration: In this phase, all the pieces are brought together into a testing environment, then checks for errors, bugs.

Maintenance: This is the final phase of this methodology. Software is deployed for end users. Repairing, upgrading, and overhauling are applied in this part.

The advantages and disadvantages of waterfall methodology are summarized in Table 2.3.

Table 2.3 – Waterfall Methodology Advantages and Disadvantages [51]

Advantages	Disadvantages
• Easy to understand and use • Prepared structure for inexperienced programmers • Supply to prepare requirements orderly • Allows management control and departmentalization • If the quality is more important than schedule and cost, this works well • Can be adapted to any type of system easily • Gives permission to well documented software	• All requirements must be known • Inhibits flexibility • Can give a wrong idea about progress • Adjusting scope can kill a project • Risk and uncertainty are high. • Changing the software is high risky • Working version is not seen until late of the software life cycle. • Many projects are not in order

2.5.2 "V" Methodology

In fact, this is an alternative type of waterfall methodology. Main difference is verification and validation stages. In this methodology, all phases before implementation have testing phases after implementation (Figure 2.16).

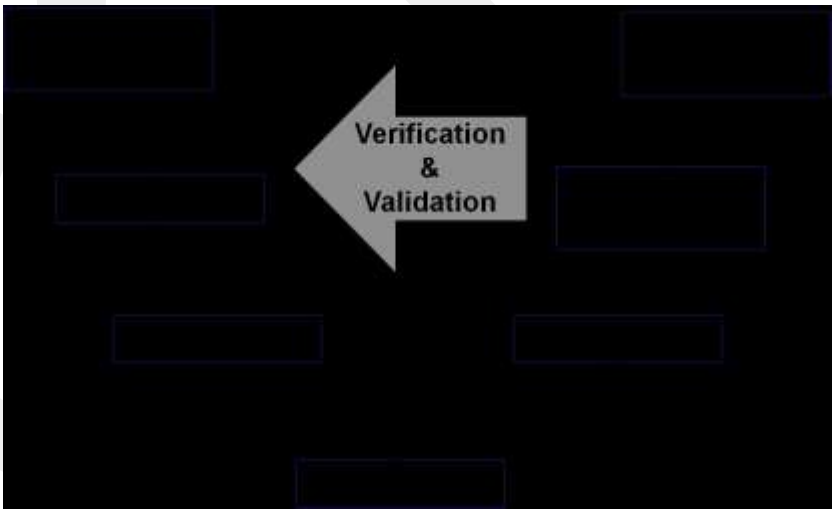

Figure 2.16 – V Methodology [51]

Disadvantages of this methodology are the same as the waterfall methodology, as well as verification advantage is that validation part has three testing phases as summarized below, which makes the testing process more reliable.

Unit Test: With this testing, each module in software system is controlled, checked that runs as expected.

Integration Test: Testing the system harmony.

Acceptance Test: Last test of the methodology. In this testing customer determines if the system satisfies its acceptance criteria.

The advantages and disadvantages of V methodology are summarized in Table 2.4.

Table 2.4 – V Methodology Advantages and Disadvantages [51]

Advantages	Disadvantages
• Accentuate planning for verification and validation of the software system • Each development phase must be testable • Development processes can be tracked easily • Easy to use	• Des not handle parallel events • Does not handle dynamic chances in requirements • Not contain risk analysis activities • Design is not authenticated • Requirements are not verified • If there is an error in system design, it occurs in integration testing, so it is lost cost and time

2.5.3 Incremental Methodology

This methodology is an advancement of waterfall methodology. All development phases are same as waterfall, but this has multiple development cycles. Thus, this methodology also called as "multi-waterfall". After each cycle is finished, a working version of software is build (Figure 2.17).

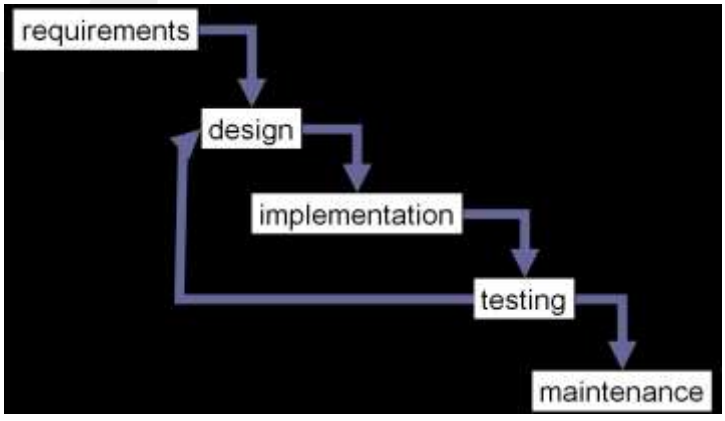

Figure 2.17 – Incremental Methodology [51]

The advantages and disadvantages of incremental methodology are summarized in Table 2.5.

Table 2.5 - Incremental Methodology Advantages and Disadvantages [51]

Advantages	Disadvantages
• Each cycle has an operational product • Customer can show and control each build • Uses "divide and conquer" breakdown of tasks • Low initial delivery cost • Fast initial product delivery • Develop important functions first, so customer get these early and he/she is happy about future • Changing requirements are reduced • Easier to test and debug during a smaller iteration	• Each phase is inflexible and do not overlap each other • All requirements are put aside front for the whole software life cycle

2.5.4 Spiral Methodology

This methodology is an evolutionary version of incremental methodology, with more emphases placed on risk analysis. It was developed by Boehm in 1988. Each iteration of the prototype represented as a cycle in the spiral (Figure 2.18).

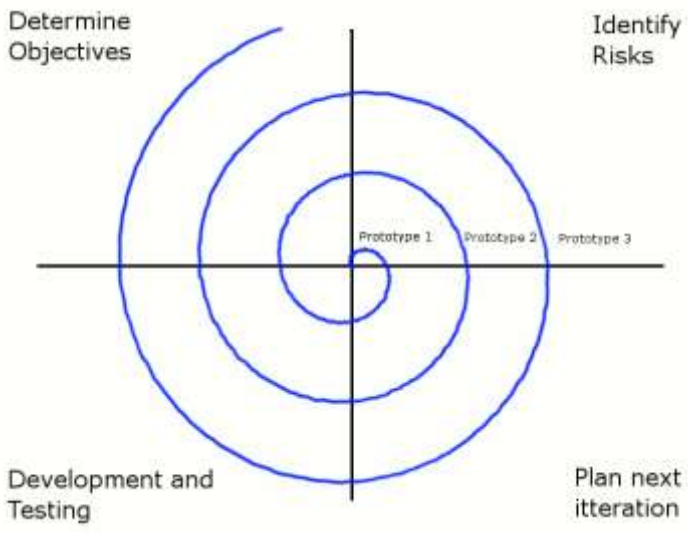

Figure 2.18 – Spiral Methodology [51]

Spiral methodology has four phases: Planning, Risk Analysis, Engineering and Evaluation.

Planning: In this phase, requirements are determined.

Risk Analysis: A process is undertaken to determine risk and alternate solutions. At the end of this phase a prototype is developed.

Engineering: Tests are put into practice for prototype.

Evaluation: The customers evaluate the output of the project to date before the project continues to the next spiral.

The advantages and disadvantages of spiral methodology are summarized in Table 2.6.

Table 2.6 - Spiral Methodology Advantages and Disadvantages [51]

Advantages	Disadvantages
• Most flexible methodology • Easy monitoring • Changing is not difficult • Cost and schedule become more realistic • Suitable high risk projects • High amount of risk analysis • Produces software early life cycle	• Can be costly methodology to use • Need specific expertise • Does not work well for small projects • Success of project depends on risk analysis phase

2.5.5 Prototyping Methodology

In this methodology, when the requirement analysis is completed, a prototype is made, so the development process gets started. After the prototype is produced, customer or end users test and evaluate. Feedbacks of the customer and end users are transmitted to the developers. After a finite number of iterations, the final version is given to the customer (Figure 2.19). This methodology is the most popular one in the software development sector.

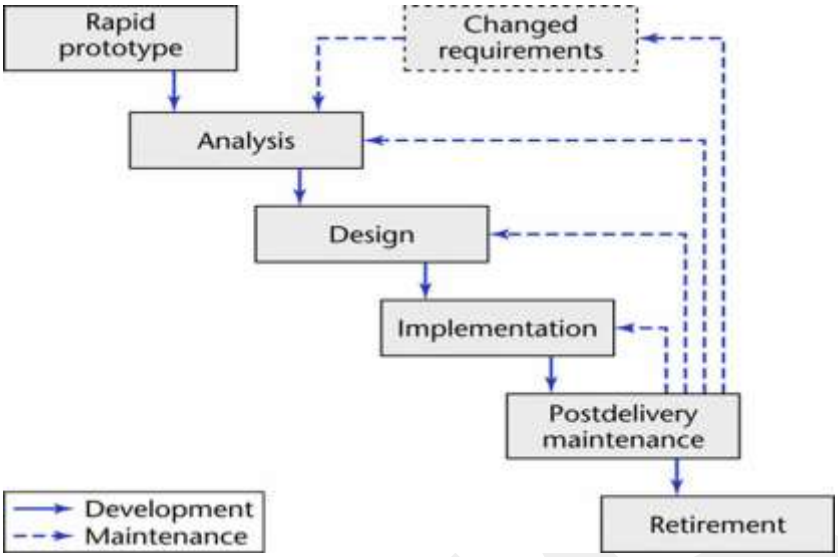

Figure 2.19 – Prototyping Methodology [51]

The advantages and disadvantages of prototyping methodology are summarized in Table 2.7.

Table 2.7 - Prototyping Methodology Advantages and Disadvantages [51]

Advantages	Disadvantages
• Customers can see the requirements • Short distance communications between developers and customers • More successful final version • Allow more flexible design • Visible progress of the development life cycle	• Bad reputation for "quick-and-dirty" methods • Maintainability may be overlooked • Customers are not satisfied, so the process can continue more than expected. • Strong management is required • Might involve extra cost

2.5.6 Rapid Application Methodology (RAD)

RAD is one of the most used methodologies. In 1980, it is developed by IBM. The aim of the RAD is overcoming the weaknesses of structured design method. The basic of this methodology is building prototype rapidly (Figure 2.20). However, compulsory participation of user constitutes the main difference of this methodology from others. Thus, software development system continues; quick building software, giving to user to evaluate and give feedback, then software is advanced.

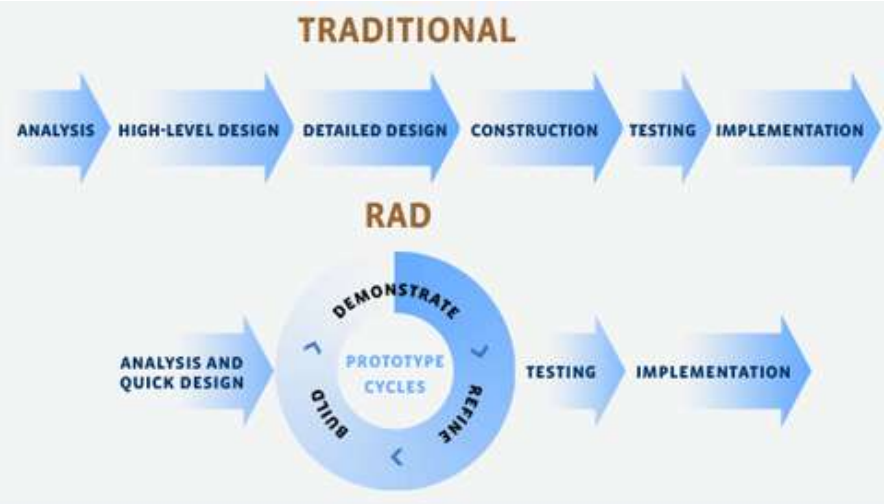

Figure 2.20 – Rapid Application Methodology (RAD) [52]

RAD has four main phases:

Planning Requirement and Analysis: Strategies for development and tools for software are also laid out in this phase.

Design: In this phase, users work together with systems analysts and develop methodologies and prototypes that represent all system processes, inputs, and outputs.

Construction: The productivity tools is used and generated. The principle of this is 'Do until done'.

Implementation: Installation of the software project, user acceptance, testing and training parts are applied.

The advantages and disadvantages of RAD methodology are summarized in Table 2.8.

Table 2.8 - RAD Methodology Advantages and Disadvantages [51]

Advantages	Disadvantages
• Reduced development time. • Increases reusability of components • High modularization achieves a more flexible and maintainable system • Quick initial reviews are possible. • Constant integration isolate problems and encourage customer feedback • Decreased cost and schedule risk because of time-box	• Need strong team and individual performances for requirement analysis • Only system that can be modularized can be built • Dependency on modeling skills is high • Not appropriate for cheaper projects • Not a suitable methodology in the absence of user. • Reusable components are requisite to decrease software development time.

2.5.7 Agile Method

In fact, agile method is combination of iterative and incremental methodologies. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change (Figure 2.21).

There are many specific agile development methods; Extreme Programming, Scrum, Crystal, Feature Driven Development, Rational Unified, Dynamic System Development, Adaptive Software Development, Open Source Software Development [53, p. 4]. Development, teamwork, collaboration, and process adaptability are promoted throughout the life-cycle of the software project.

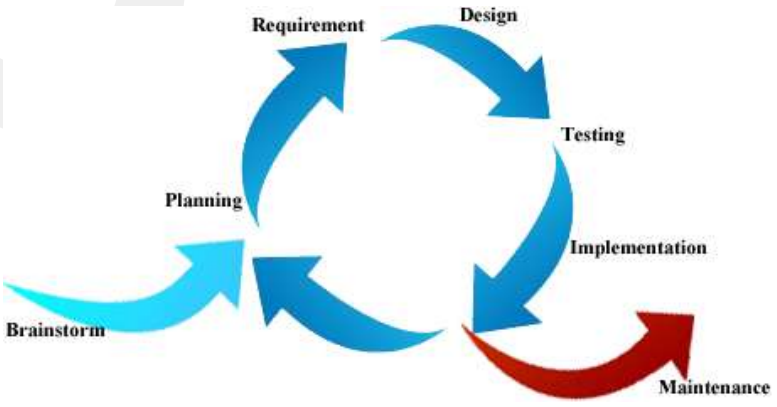

Figure 2.21 – Agile Method [51]

The advantages and disadvantages of agile methodology are summarized in Table 2.9.

Table 2.9 – Agile Methodology Advantages and Disadvantages [51][54]

Advantages	Disadvantages
• Adaptive team which is able to respond to the changing requirements • Not invest time and effort • Face to face communication • Fast documentation • End result is a high quality software • No guesswork between the development team and the customer	• Difficult to assess the effort required for some software deliverables • Lack of emphasis on necessary designing and documentation. • The project can easily get taken off track if the customer representative is not clear what final outcome that they want. • Difficult to judge the efforts and the time required for large projects

2.5.8 Comparison of Traditional Software Development Methodologies

In all of these methodologies, the technical software development and the end user requirements are considered. However, in the case of serious game development, there is another important stakeholder of the software development process named domain experts. Accordingly, there is a need to consider the communication and input of domain experts in the software life cycle in a systematical way.

In the literature, there are many comparisons of software development life cycles ([55][56][57]). As to these comparisons, level of use and success of the software methodologies are summarized in Table 2.10.

Table 2.10 – Comparison of Development Methodologies

Methodology / Feature	Waterfall	V Model	Incremental	Spiral	Prototyping	RAD	Agile
Requirement Analysis	5	5	5	5	3	4	3
Understanding	5	5	5	5	4	5	5
Cost	2	4	3	4	5	3	5
Simplicity	2	3	3	3	2	1	4
Risk	5	3	3	3	2	2	3
User Involvement	2	2	3	3	3	2	3
Flexibility	1	1	1	2	5	4	5
Maintenance	2	2	3	3	3	4	3
Reusability	2	3	4	4	1	3	3
Success	2	3	3	3	3	3	4

5 – Very High 4 – High 3 – Medium 2 – Low 1 – Very Low

As seen in Table 2.10, almost all methodologies have their own advantages. However, one of the main requirements of educational software, user involvement, does not have high point in any of these. In educational software, user includes both domain experts and end users (learners/patients). In this study, a methodology is proposed to consider both domain experts; learners expectations in a systematical way.

Hence, this study aims to better address domain needs and expectations in educational software development process. For this purpose, we proposed a new software development methodology for the educational software development process. Additionally, we have tested this methodology in developing software for speech disorder treatment process. Therefore, in Chapter 3, we have detailed our research methodology to reach this objective.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter presents the research methodologies that have been used in this study. Besides the research questions, while developing the serious game, requirement of the system and used tools are explained below.

3.1 Research Questions

In this study, a serious game is developed for children who have speech disorder. The game was developed with the collaboration of therapists and rehabilitation centers as to their requirements. Research questions of this study are:

- How the proposed methodology can be applied on the development process of a serious game?
- Can an educational game, designed for language therapy, be employed during a therapy session?

To answer the research questions, prototype educational software is developed and tested on children. According to domain experts (therapists), there is a need for such software. Main expectation is that patients and especially children may show an interest in such games and they may benefit through the game. Thus, this study is expected to make a major contribution for the treatment process of speech disorder children.

3.2 Proposed Methodology (Educational Software Development (ESD))

While the serious game is implemented, software development life cycle (SDLC) has continued different from other SDLC methodologies. In classical SDLC methodology, there are many participants; software developer and customer or end user. The software is prepared for them and they test and approve the software product. Software testing and approval take place sometimes during development or sometimes at the end of the project before installation. For instance, in waterfall development methodology, this time is at the end of the development, besides in rapid application methodology, end user tests the software during development. However, in both of the methodologies, delivery of the software is about only customer's satisfaction.

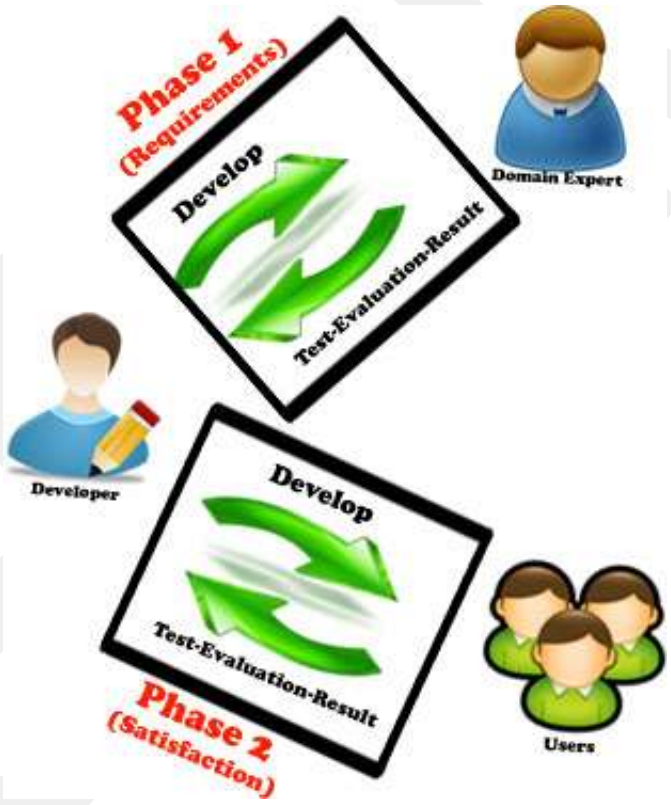

Figure 3.1 – Educational Software Development Methodology (ESD) Cycle

Even if a serious game project looks like an ordinary game project which has two participants, the developer and users that need to be pleased. This study has showed that there are three main participants for developing a serious game; developer(s), domain expert(s), and end user(s) (Figure 3.1). In the methodology, domain expert(s)

can be an educator, doctor, or therapist depending on the purpose of the developed software. Domain experts usually need to use the same software to adopt it according to the specific needs and situations of the end users. Hence, domain experts are also a different user group of this software who need a control and adopt the system to be efficiently used in the educational environments. Similarly, end user(s) also can include different types of people such as learners, students, trainees, patients. On the other hand these two groups of end users are not independent to each other. The domain experts have the control over the settings and usage of the software system. Hence without the expertise and control of the domain experts on such systems it is hard to get expected benefits through such software. In other words, domain experts are the key people on the successfully integration of the developed software systems into the current educational or treatment environment.

Accordingly, we propose to reconsider the software development methodologies by including the domain expert(s) contribution to the software development study for such software projects (Figure 3.2).

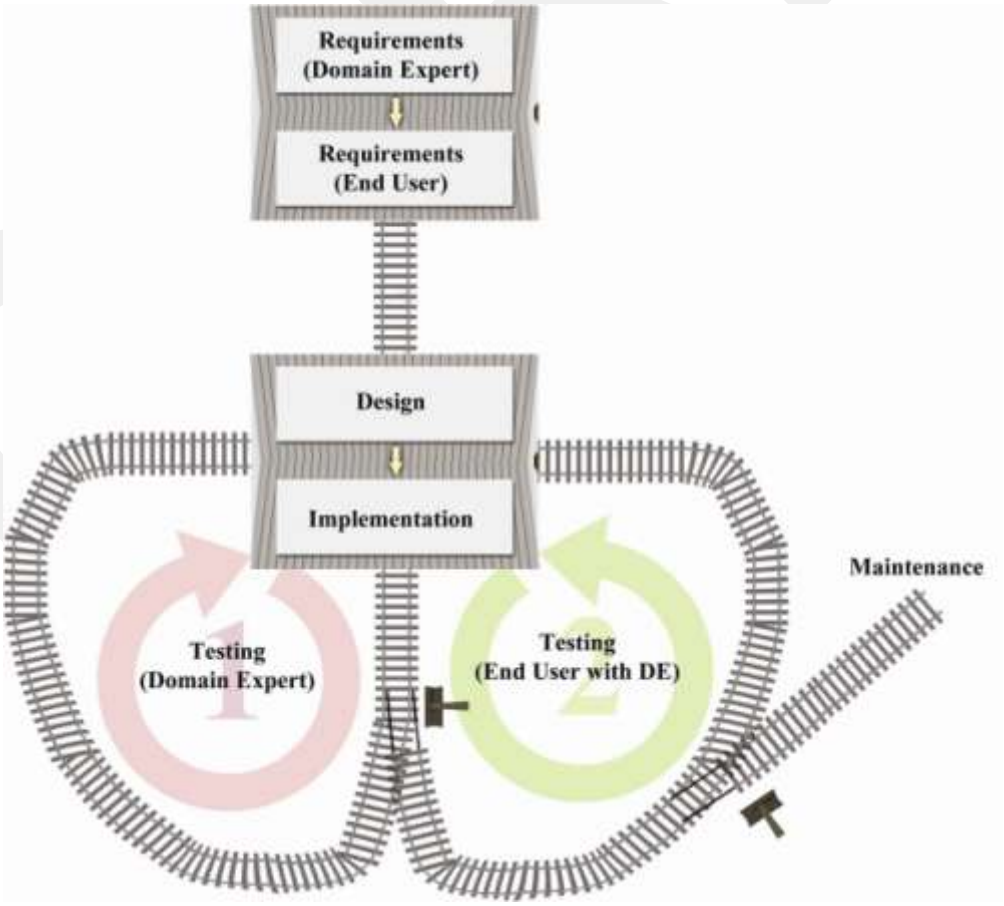

Figure 3.2 –Architecture of ESD Methodology

This software development methodology can be resembled to railway and railway stations. First station is collection and specification of the requirements. In this station, the requirements are collected from both domain experts and learners/patients as end users in turn. In this study, we followed this methodology and to understand the needs of therapist and experts, we interviewed with three domain experts which were a professor and two clinicians. Gathered data included requirements by domain expert such as "Things that they slog on while they are prepared to therapy session, during session and after session.", "Things that they need for therapy", "How do they record the sessions' data?", "How do they supply the motivation of the patient during session?". After that, the requirements of the end users were collected with domain experts. To do this, an interview was arranged with two patients who are children in this study, in trust of the domain experts. From those interviews, data needed were collected. This data was their age, gender, problems. In addition, information was obtained about whether they liked computer games and what kind of games they played. With this data first station of the ESD methodology is completed.

Second station includes design and implementation parts of the development cycle. This station can be considered as the heart of the educational software development process. Collected data is processed in this station and a prototype is the result. The 3D serious game for language disordered children was coded in this station with collected data from first station and a serious game and an administrative setup occurred which were mentioned in Chapter 3.

After this station, the prototype takes the road to be tested. In this way, it passes from the key point of the software development process. This key point is like a railway switch. This switch can provide two paths: One of them is domain expert testing and the other is end users' (learners/patients) testing ***with domain experts***. Both of the ways create cycles (Cycle 1 and 2) to the station that they started from. First, the prototype must moving toward the domain expert testing (Cycle 1). And this cycle continues until the domain experts approve the prototype to be used. After then, the railway switch is changed toward the end users' testing (Cycle 2) and new cycle starts on the end users with domain experts. This cycle can continue, if software has developmental bugs or errors. However, if software needs changing on requirements, when the prototype comes to station, the railway switch can be changed toward the

domain expert's testing with the desire of the domain expert. Thus, first cycle starts again and the prototype again needs end users testing after this testing. The end users' testing cycle also has an extra railway swich to another way, maintenance, to exit. In this cycle, the acceptance comes from domain experts again. With approval, the prototype takes the maintenance path. Therefore, ESD methodology is completed and it is shown in Figure 3.3 in detail.

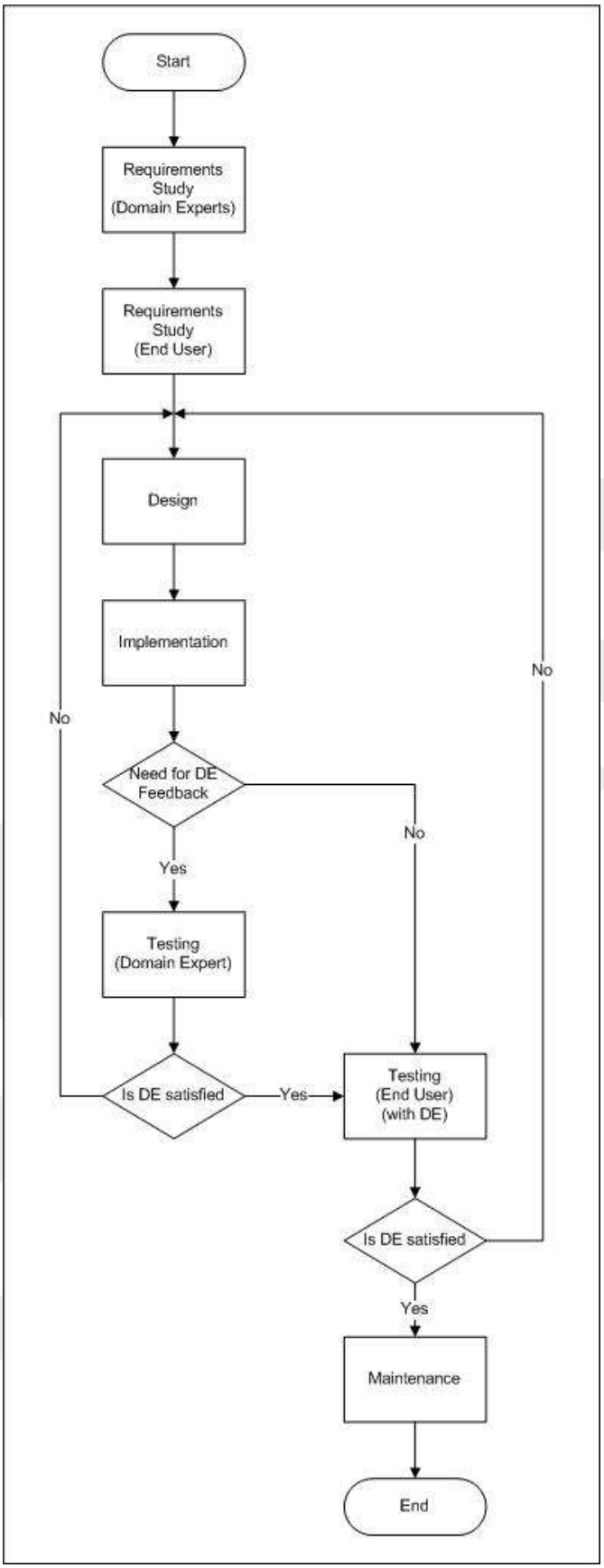

Figure 3.3 – Flowchart of ESD Methodology Process

Developing a serious game is a task that needs more attention in comparison with common game projects. Since serious games are not for entertainment primarily and they are used for more important topics such as government, military, health etc. for people who are in need of keep, the process to develop them. It must be carried out methodically and with care.

ESD methodology can be applicable to any software development methodology found in the literature. We propose a two stage testing process by considering domain experts' and end-users' view points. ESD methodology can also be applicable for the software development process of any educational software. We believe that, the classical software development methodologies cannot address main requirements of educational software development processes. By considering domain experts' viewpoints together with the end users (learners, patients) ESD methodology provides more efficient way of software development process for educational software developers.

3.3 System Requirements and Development Tools

To demonstrate a convenient serious game, there are many important issues to be considered. Examples of such requirements are: usefulness, educational, simplicity, maintainability and flexibility. To fulfill such diverse requirements, it is significant to get the bottom of needs. In this study, this matter is taken into consideration and it is interviewed with three therapists in the speech therapy and therapy centers. As the results of the interviews, it is possible to sort the requirements:

- The game should be usable for each patient,
- Because many patients are children, it should be attractive for them,
- It should be reusable,
- The system should provide some data for experts or therapists after therapy session like audio recordings, activity records,
- The environment of the game should be flexible for therapist, so he/she can create different environments for different occasions.

Based on these facts, it was decided that the structure of the serious games system originates from two main parts (Figure 3.4) and in Figure 3.11, use case diagram of the system is shown:

- 3D Game Platform (Unity3D),
- Administrative Tool (Web platform (php), Database (mysql))

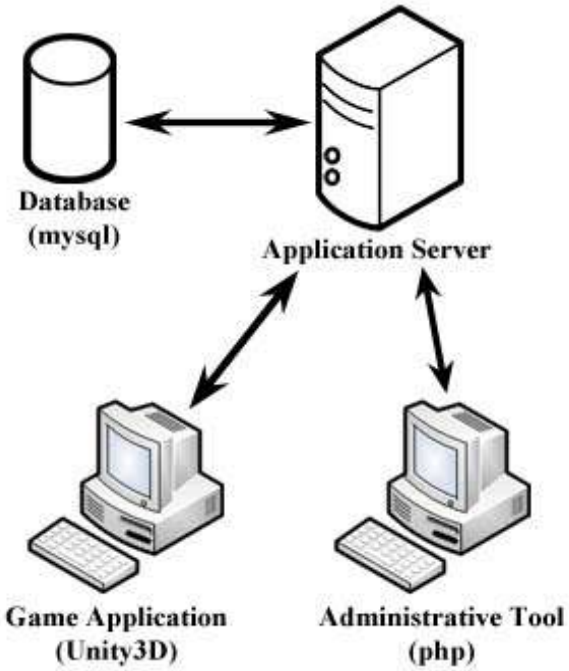

Figure 3.4 – General System Structure

3.3.1 3D Game Development Platform (Unity 3D)

The main part of the system is the game platform. This platform is the visual and effective part of the system. Hence, it is decided to create 3D environment for game platform, because this platform should be efficient for children. As game environments have more positive effects for children. Personal parameters of games are [58, pp. 117-126]:

- Affectivity,
- Mobility,
- Intelligence,
- Creativity,
- Sociability

In addition to these, 3D games are also [59]:

- Useful about instinctual achievement and progress,
- Help to learn personal
- Improve to solve problems and to think strategically,
- Interest in content of the game,

- Relaxing and motivation for students,
- Self-sufficient and provide success for lessons

To develop the 3D game platform, Unity3D game engine was used. It creates 3D environments, as well as providing robust support of programming languages, multiple platform deployment, built in networking support, and general ease of use are reason to choose (Figure 3.5).

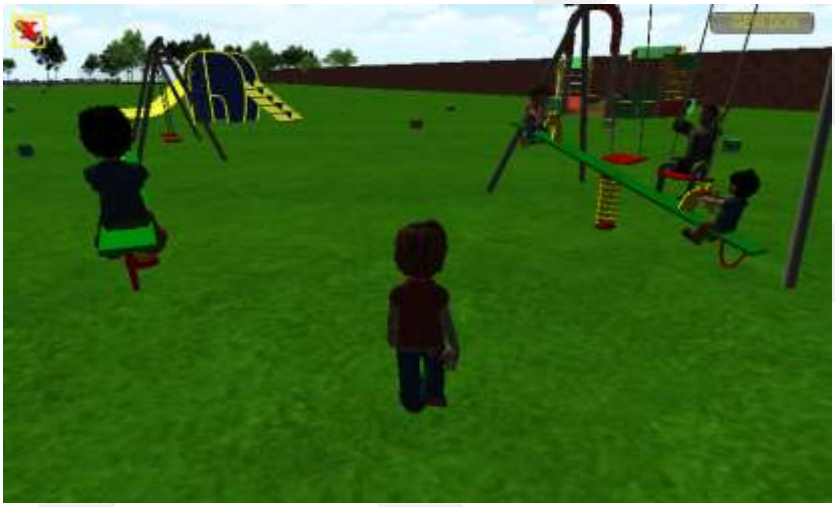

Figure 3.5 – A View from Unity3D

Flexible features of Unity3D support the system to make it a network game. Easy implementation and maintainability allow using the system everywhere such as home, clinic and hospital.

All 3D objects are created by the developer. Each object is taken up specially and new features and abilities are added. Prepared object is exported from Unity3D as special unity object; '*.unity3d'. Thus, each object can be added to game environment dynamically (Figure 3.6).

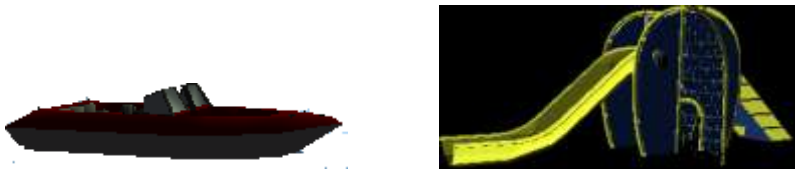

Figure 3.6 – Some of the Game Objects

The 3D environment has a structure. This structure consists of nested structures as seen in Figure 3.7:

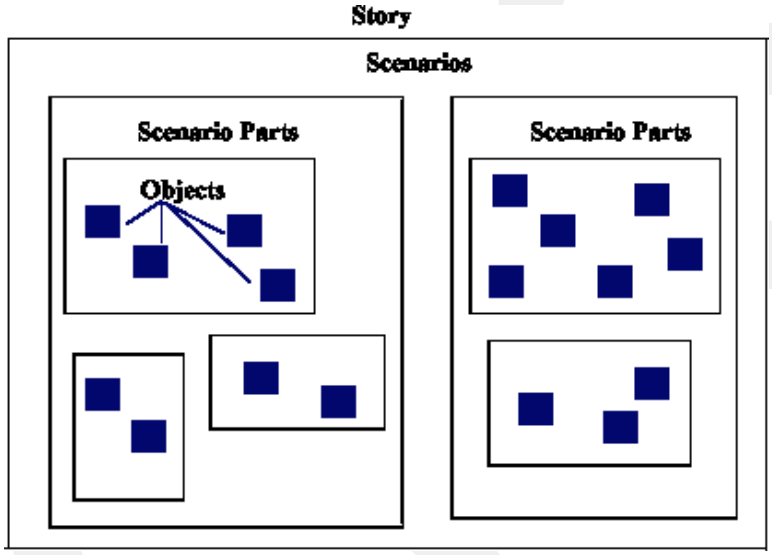

Figure 3.7 – Structure of Developed Serious Game

Object: It is the smallest part of the structure. It is created by the software developer and added to database and object directory. Each object has its own features and abilities. For instance; a ball can roll, a ship can stay on water and swing can be slipped.

Scenario Part: It includes objects that therapist want to add. This structure is a pattern for patient to develop or support to a problem of him/her. For example; therapist can put a nail, a hammer, a frame and a wall object in a scenario part to learn or advance the patients' realization, insight and use.

Scenario: This is a main part of the 3D serious game. A scenario can have one or more scenario parts. The therapist can use one or more scenarios during a therapy session.

Story: It can have scenarios. For instance; a story can be a house. Each room can be a scenario, too.

This structure allows the therapist or the expert to put any object, scenario parts and scenario to appropriate, desired place in the system. Thereby, he/she can use the same object for different reasons in different places according to the child's needs. For instance; a bench can be used in a park or in a house garden, a trash can be used on the street or in the house, in the kitchen.

In addition to all these features, a 3D game environment offers two types log data:

- 3D System Log
- File Log

3D System Log: While during therapy session, game environment records the stories which were used and also all scenarios in them, the date, for how long they were used, the score and success level (Figure 3.8).

Figure 3.8 – Administrative Setup for Expert(s)

File Log: This data is recorded in a txt file named by patient 'name+date+time'. In the file, all activities of patient are listed; used stories, scenarios, for how long the object is used (Figure 3.9).

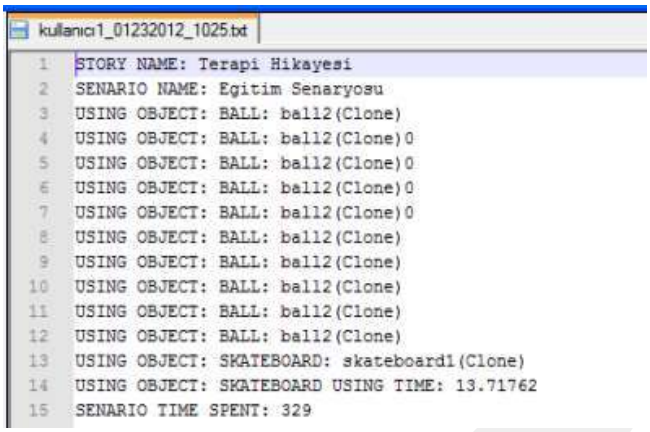

Figure 3.9 – A Patient File Log

3.3.2 Administrative Tool

The system needs an admin control panel to arrange 3D game environments and also it supplies to control patients improvement and their pass therapy data. On the other hand, 3D objects require storage and also their data needs a database. Because of these needs, an administrative tool was created as web application with PHP and as database software MYSQL was used. Administrative tool includes many features for 3D environment and therapist:

- Patients can be added, deleted, updated
- Therapist can see previous therapy data and his/her comments about sessions (Figure 3.8)
- Scenario part can be added, deleted, updated (Figure 3.10)
- Scenario can be added, deleted, updated
- Story can be added, deleted, updated

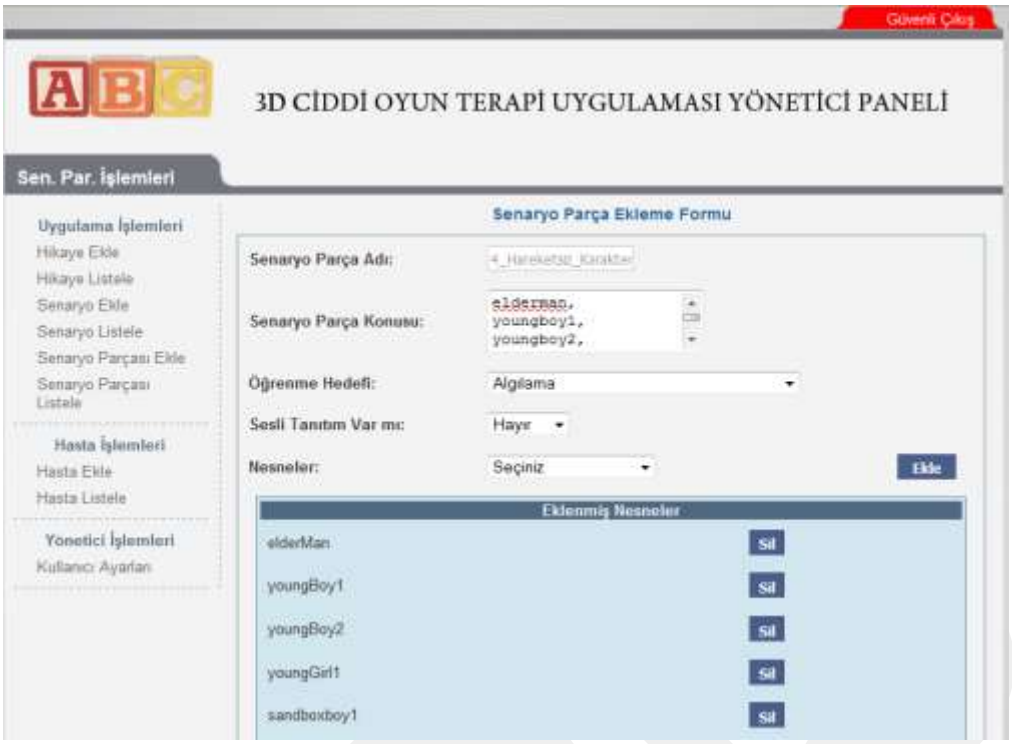

Figure 3.10 – Administrative Setup to Create Story for Patients

Web tool supplies an extra log system for therapist. He/she can add special notes for each scenario. Therefore, therapist's comments are chain and when he/she examines the old records of a patient, he/she can understand the his/her progress.

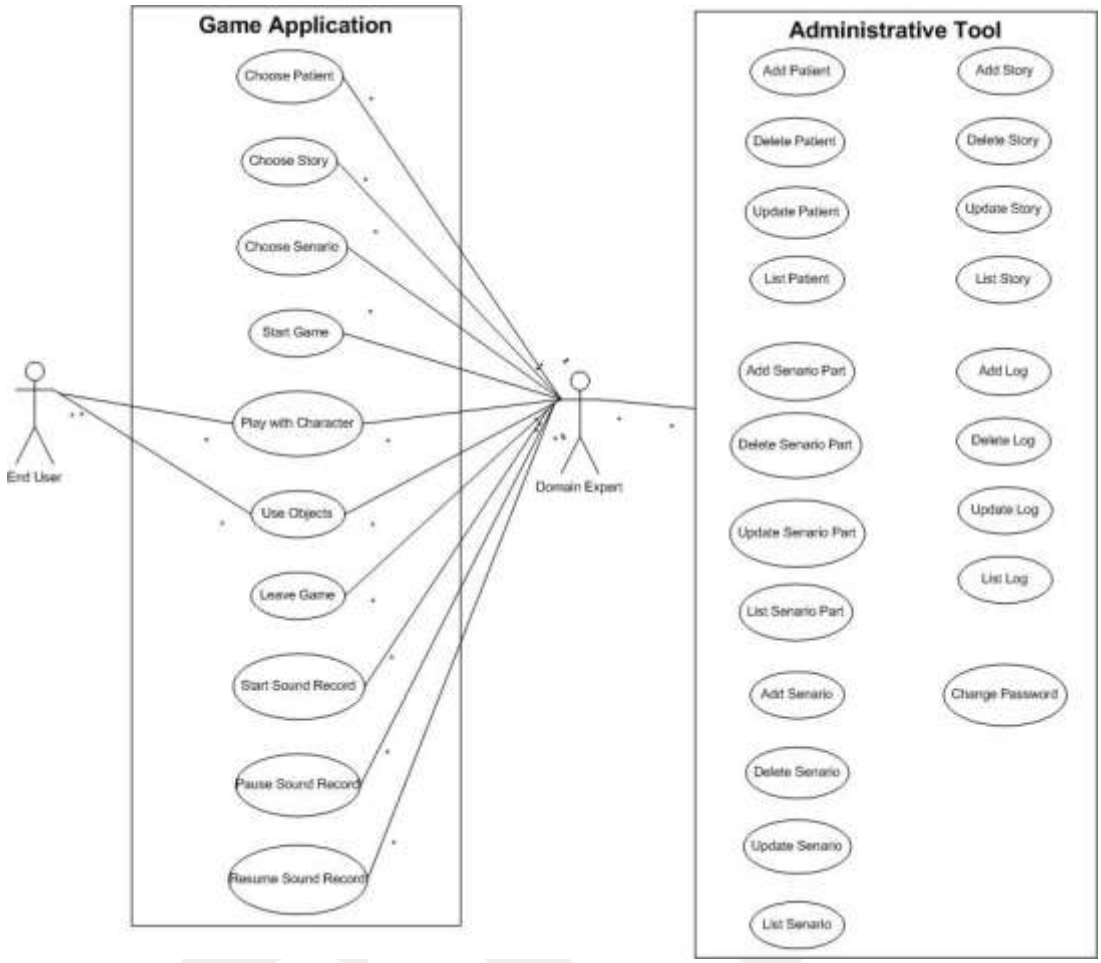

Figure 3.11 – System Use Case Diagram

CHAPTER 4

RESULTS

The resulting data and approaches of the developed software which is presented in Chapter 3 are explained in this chapter. Furthermore, the methodological approach to develop educational software which is used in this study is detailed. In the second part of this chapter, the mentioned structure of the tested software and collected data is explained.

4.1 Development Process of the System Using ESD

Development of the prototype was completed with 2 requirement studies and 6 development cycles (Design, Implementation and Testing). All stages will explain below and also it is shown in Figure 4.8.

- **Requirements Study (Domain Experts (DE))**

As the first process of the ESD methodology, interviews have been conducted with the therapist who is an expert about speech and language disorders. According to these interviews, data is collected about speech and language disorder, what methods are applied during therapies, what difficulties they encounter, what deficiencies they need and what kind of system can be useful for them. According to interview with Prof. Dr. Pınar Ege:

- Generally children have speech and language disorder (2-6)
- Children have problems about affixes
- Children have problems about tenses
- Children have problems case of nouns
- They generally omit affixes or use incorrect
- During therapy child cannot be forced to use word correct by therapists, they use different examples to treat that affix problem.

Common features of the system for prototype with guidance of Prof Dr. Pınar Ege:

- Because child cannot know the things or feature, words can be teach before the main therapy part
- It can be added many objects for one word as much as desired
- It can be possible to move in application environment by a character
- It can be possible to use objects in application environment by a character
- The character in application must be realistic.
- Prototype application environment can be a park area because it is suitable for children
- Park area can be include pool, swings, slides and sandbox
- Application environment can be stoppable by the therapist to speak with child about actions or objects.

With this data about speech and language disorder and features about application, requirements study with domain expert part of ESD methodology is finished.

- **Requirements Study (End User)**

Although during the first requirements study (with domain expert) some data can be collected about people who have speech and language disorder from therapist, to collect more detailed data, we went to a rehabilitation center to see disordered children. In the center, we interview two children and we asked questions:

- Do you have a computer in your home?
- Do you like computer games?
- What kind of computer games do you play?

As to answers, again some data comes in sight:

- Children like many kind of games
- We must use basic objects in application
- Children are too young to use mouse, so character and application must be controlled with keyboard.

1. Cycle (Design – Implementation – Testing) (Domain Expert (DE))

With data from domain expert and end users the first prototype (Figure 4.2, Figure 4.3) and also an administrative tool (Figure 4.4) were developed. Developed environment was a park that had a slide, a sandbox, some benches, some trashes, some trees, a pool and two characters (a boy and a girl) (Figure 4.1) to play. When the therapist saw and played it, she gave some comments about it.

Game application comments by domain expert:

- Characters must be suitable for group which has pre-school age children. Especially the girl character was inappropriate; she was old and so sexy.
- Character can be run more slowly, because child cannot be control easily.
- There must be more objects and actions in the environment such as swing and other people.
- All objects' names can be learnt, so the application must supply this feature.
- Application must be record sound during therapy session.
- Application environment must have much more sward.

Figure 4.1 – Girl and Boy Characters in the First Prototype

Figure 4.2 – A View From the First Prototype

Figure 4.3 - A View From the First Prototype

Figure 4.4 – A View From Administrative Tool

Administrative Tool Comments:

- A therapist should see older therapy session logs
- A therapist should add extra information about therapy.

2. Cycle (DE)

With guidance of the first cycle's comments by the domain expert, the system was improved; many new features were added and some features were changed:

- Girl character was omitted from environment
- A question mark was added to all objects. When it was clicked, its name would be vocalized (Figure 4.5).
- Some dynamic characters were added into the environment (Figure 4.6)
- Needed features for administrative tool were added.

Figure 4.5 – Question Mark to Vocalize Object Name

Figure 4.6 – A View From Second Prototype With Dynamic Characters

After the test with domain expert, again the domain expert suggested some features for game application:

- Application environment still was not big enough
- Application environment need much more objects and actions to run, because a therapy session generally takes at least 30 minutes. However, the environment was not having enough topic and action to speak with children and it was not exiting for children sufficiently.

3. Cycle (DE)

As the third cycle, a new game application was developed with extra features (Figure 4.7).

Figure 4.7 – A View From Third Prototype

Added features:

- Character was changed and made like a child.
- Park environment was made bigger and added many objects with animations and usability (Teeters, swings, a lake, a small forest, park walking area, benches, trashes, many people in the park)
- File log system to record all character actions and action time

With these features the game application was tested by domain expert and she found it acceptable to test with end user (patients).

4. Cycle (End User with DE)

The fourth cycle was the first test with patient. The test environment was prepared in the therapist's office with a normal child, a boy, (details about this test will be clarified in Chapter 5, 1st Child). The child was so active and could use the computer easily. He was curious, so he used almost all objects. During the session, the child asked there was a car in the park. Thus, it showed that if a patient is a boy, the environment needs masculine objects. Although the test had these difficulties, therapy session could take about 30 minutes.

After the test the domain expert proposed new features about game application:

- It would be better, if some objects and animations into the environment for prototype, especially for boys.

➢ The test was applied with a child who has not had speech and language disorder. He was energetic and he did not need to direction to use application and to speak much. However, if the prototype applies on a patient, actions in the environment cannot be enough for the patient.

5. Cycle (DE)

In this cycle, with new features game was presented to test:

- A usable boat and some animated boats in the lake
- Therapist can stop and run again all environment animations anytime
- A usable skateboard especially for boys
- Mini seats for character.
- Some garbage to take and put into the trashes

With these features domain expert tested and accepted that this prototype is ready to test with a child patient.

6. Cycle (End User with DE)

This cycle was with a child, a girl, who had speech and language disorder because she had hearing problem in a rehabilitation center. Details of this test will be explained in Chapter 5 (2nd Child).

Although the girl was shy and not active as much as the child who was in 4th cycle, therapy session took about 45 minutes. There was no problem to use, to record and to test with child. Even she started to use application herself towards to the end of the therapy session. Thus, this was the last cycle of ESD methodology to develop an educational application.

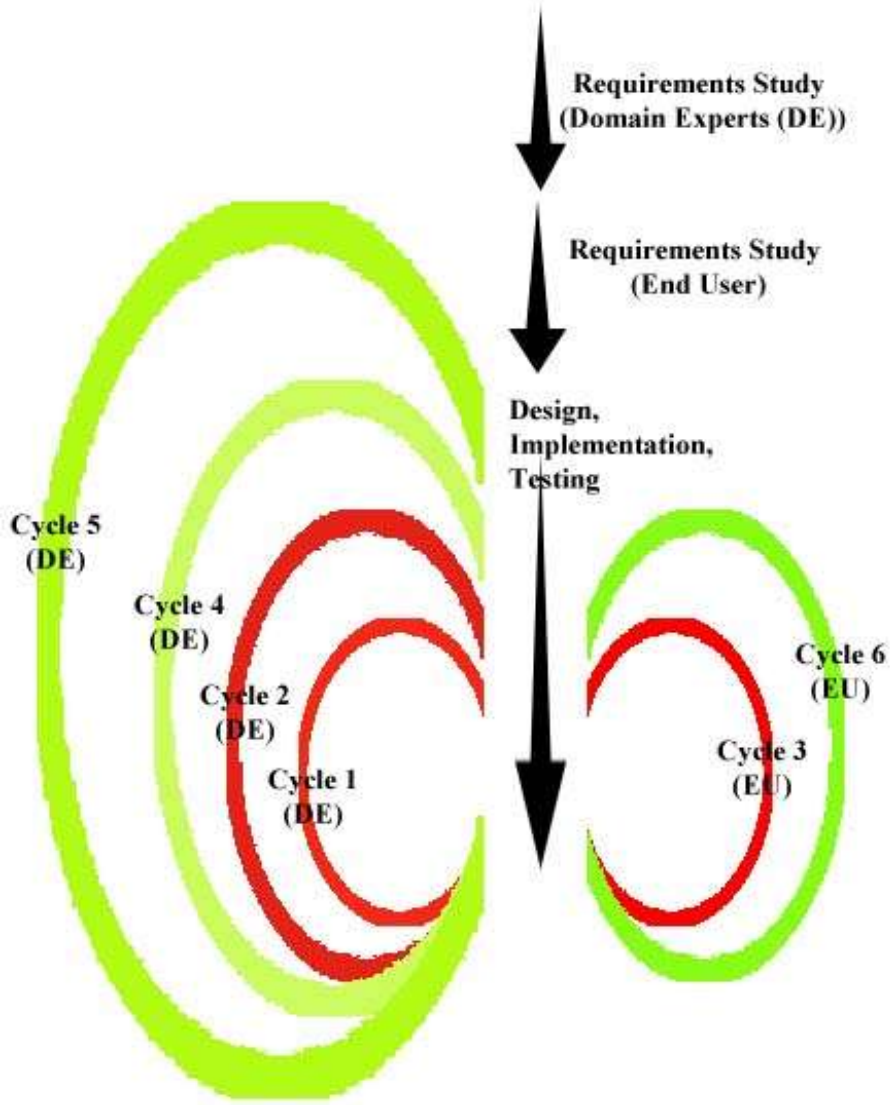

Figure 4.8 - Prototype Development with ESD

4.2 Game Environment Architecture

The game environment was constituted like a big park simulation. Many park objects were created and added, such as a swing, teeter, bench, a lake and ships. Additionally, a 3D main boy character is designed to interact with the objects in the game. While the main character is created, it was kept in mind that the patients were children. The character should not be scary or unreal for children. Hence, as seen in Figure 4.9, the software architecture was divided into two story parts (two scenarios).

Figure 4.9 – Scenario Selection

Education Scenario: The children need to learn object names to produce sentences with them. As was mentioned, in speech therapy session, therapists use objects or images to communicate with patients. However, if the patients do not know the objects or cannot name them, therapists must first teach the object names by using their images. Because of this reason, this scenario was created.

In the scenario, all objects had a 'question mark' button over or near them (Figure 4.10, Figure 4.11). When the button is clicked, recorded sound file has vocalized the object name, plays. Thus, the patient could learn the name of that object.

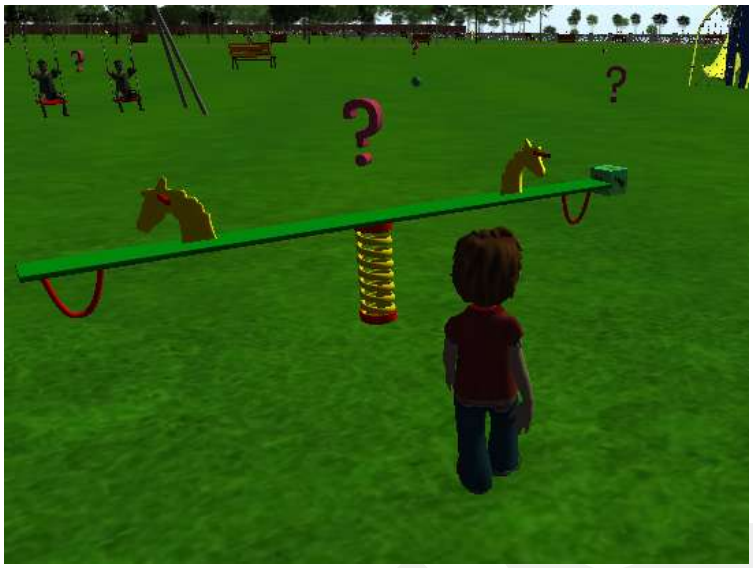

Figure 4.10 – Education Scenario of the Serious Game

Figure 4.11 - Education Scenario of the Serious Game

Because this scenario was an educational scenario, almost all objects' features and motility were disabled. Only the main character could be moved to reach the objects.

Therapy Scenario: The second and main scenario was this one. There are no educational objects in this scenario. In large part, the objects were active and useable

by the main character. The character could seesaw; climb up and down the slide, play football, swing, and ride on a ship and more (Figure 4.12 and Figure 4.13).

Figure 4.12 – Shipping Activity

Figure 4.13 – Riding Teeter Activity

Apart from these, the game environment had some additional features for the therapist. One of these features was '*sound record*'. Sometimes therapy session is long and therapists can have more than one session in a day. They have to keep a record of the sessions. The game application supplies a recording system, too. The system could be controlled by the therapist. Each sound file is saved as scenario based. Hence, if the therapist needs to listen again, s/he can evaluate it later. During the session, therapist could pause, resume and re-play the record.

Another feature of the game environment is that '*all animations can be stopped*'. Even if the application is a serious game, it is a game for children. As is known, when children play games, it is hard to stop and control the motivation. Therefore, this feature would help the therapist to control the progress of the session. With a special key, the therapist could stop the animation when s/he wanted, s/he could speak and ask questions.

4.3 Speech Therapy Usage

In an attempt to understand the affect of the newly designed 3D serious game in contrast to traditional therapy methods and sessions, the system was tested with two 5-year-old children. One of them was a normally developing child who had no language/speech problem. The improved system with his feedback from this application was applied on another child. This child had delayed language because of hearing-impairment (Figure 4.14).

Figure 4.14 – A View from Testing

1st Child

First child was a boy who had no speech or language disorder and was computer literate and had experience with computer games. Because he was active and talkative, he got the control of the computer and used the game by himself from the beginning. He was willing to play the game. At that point, therapist interfered and started to speak with him. The boy did not lose motivation with therapist. Although, the game was stodgy in as far as other game that he played, the session took almost 45 minutes. After the session, according to the therapist, domain expert, that first prototype was not sufficient for therapy with language disordered children. She indicated that more objects were needed for an effective therapy session and existent objects' activity features do not seem sufficient.

The first demonstration showed that the objects in the environment were enough, but their functions needed to be improved to keep the children much more active in the game environment and therapy session performance would be better. Therefore, some functionality features of the objects were increased.

2nd Child

Second child was a girl who had hearing problem and had a cochlear implant. She was shy and was not an active computer user. First, she could not understand what she needed to do. However, after a while, she relaxed and showed interest in the game. In the first part of the therapy session, the game was used by the therapist. In education scenario, the therapist assessed if the child knew the object names in environment. The child needed to be taught some of the object names. This step was necessary, because the purpose of the session was to teach Turkish morphology that were originally chosen as the aim of the game environment (such as –i, -a, -de/da, -den/dan suffixes attached to nouns and tense suffixes attached to verbs). To communicate with child, the therapist asked some questions about the environment such as "What is this", "What is happening now", "Where will he go next". The child repeated after the clinician's sentences and answered the questions. In below, there is a dialog that based on the video recording during test;

Domain Expert (DE) – Yes, let's start. Look, what is this place?
End User (EU) - ...
DE – I cannot use this game, well, but I believe, you will use better than me. What kind of a place is this? This is not a home, isn't it?
EU- Yes
DE – What kind of place do we have in this game? What do we have here?
EU – There is a park!
DE – Yeesss, this kid has come to a park. Let's look. What will the kid do now? Let's walk with him. What did he see? What will he do?
EU – Sliding
DE – What is this?
EU – Slide
DE – Yes, slide. Let's look what else we have? Where did we come from? What did we see?
EU – I do not know.
DE – We call this seesaw. I know it is hard to say, but let's say. Seesaw
EU – Seesaw
DE – Ok, there was also a ball here. Let's find it.

EU – There

DE – Yeesss, ball is there. Do you know what will we do now? What did he see?

EU – Ball

DE – What will he do now?

EU – Play ball

DE – Yes, What will he do with ball?

EU – Hit the ball

DE – Yes, he will hit the ball. What did he do with ball?

EU – Hit the ball

DE – He hit the ball.

EU – What happened to ball?

DE – Roll

EU – It rolled.

...

In Figure 4.14, it is shown the child is paying attention. Like first boy we tested the system on, as it is be seen in Figure 4.15, she also got the control of the game in time.

There were two domain experts in the room. While the girl was playing the game, they talked about the tool. That conversation is given below:

DE – It is taught very well with this tool. By the way, we learned that she does not know to use of noun cases in sentences. It can be taught to her with this tool. Because even though we did not use so much, this tool includes many different objects in it. And we [domain experts) can add or remove the objects that we want to. So, we can make anything with them [new objects].

Other DE – Yes, very well. Additionally, I think s/he will learn the name of the objects.

DE – Yes, s/he needs to repeat. This tool will also allow for that. Ilayda repeated the object names with me. She was a good subject for us. She did not know most of the objects' name. I think this tool can support to be taught and to be used them in sentences. In addition to these, she did not lose motivation during session. We have seen it. She needs to use this again.

ODE – I was surprised. I thought that she would know the name of the objects, because she goes to park frequently.

DE – Yes, but she needs to reinforce, because she has a hearing problem. Therefore, it is harder than other children

ODE – Yes.

DE – This was an experiment. (She returned to child) Ilayda learned to use the game. You learned, bravo. Where are you walking it? What was there?

EU – Sandbox

DE – Yes, well done

...

Figure 4.15 – A View from Testing

The therapy session took about 60 minutes for her and when she left, little girl was happy about the session. As a result, therapist considered the game as successful for a first therapy session. According to the clinician, the system usage is considerably easy and understandable. With new objects, scenarios and stories, the system can be used by therapists for speech/language therapy sessions. The domain expert declared that the game can even be used for children who have different problems such as autism and hearings problem.

CHAPTER 5

CONCLUSION AND DISCUSSIONS

This study is focused on two research approaches; a new software development methodology for educational software and a serious game development for language disorder children. The research also aimed to support the speech therapy session for the therapists. Because of these reasons, the serious game and an administrative setup were created by the new software methodology. Therefore, a prototype was produced and tested on children.

As the first part of the research, we have proposed a software development methodology for educational software based on domain expert and learners/patients. We believe that the methodology is particularly more appropriate than traditional software development methodology to develop educational software. As distinct from others, this kind of software involves domain experts and needs much more attention, because it is generally about directly human beings.

Based on the results of the tests in this study, proposed methodology can be used to develop the educational software. As the second part of the study, we have seen that, the created system plays an important role for language therapy treatment. The domain experts also agree that this educational software, serious game, is capable of sporting a faster and more effective therapy session. They are also claimed that the software can be used for other disorders such as autism and hearing problems. In addition, this software has some other advantages. One of them is supporting for communication between patients. It also provides the reporting and evaluation system of the therapy sessions, too.

A new proposed methodology is applied to develop this software project. Although it is the first experience, it supplied productive and successful result. Of course, same as classic software development life cycle methodologies, this method has also some advantages and disadvantages as shown in Table 5.1:

Table 5.1 – Advantages and Disadvantages of ESD Methodology

Advantages	Disadvantages
• High success to reach useful software • Improvement with unfaltering step • Double satisfaction about software; domain expert and end users • People who will use the software can see, test and be familiar the software because they are in the cycle during development • Flexible • Produced software product is tested and confirmed by at least one expert.	• Can be hard to find end users (patients) who are same level to test again for long term projects • Can be longer than many classical SDLC methodologies • The domain expert and technical people should work systematically; otherwise it may be barrier to realize the system.

The integration of the computer technology in speech/language therapies also made this system cost-effective and time-effective means for therapists. This kind of software supplies pre-packed therapy equipment. And also therapist can create applicable software environment for different patients and can use them repeatedly.

Despite these advantages, proposed methodology can also have some disadvantages. For instance; the educational software's authority to use depends on not only technical people but also need acceptance of domain experts. This feature can be case to misunderstanding while developing the software. Thus, development time can be more than expected. On the other hand it is necessary for educational software to be reliable.

As a result of this study, we conclude that proposed educational software development methodology meets all requirements of educational software. Beside, although developed software is a prototype for speech/language disorder therapy, it can be advanced and used for other kind of problems, too.

REFERENCES

[1] Fey, M., Language intervention with young children, San Diego, CA: College Hill Press, 1986.

[2] Paul, R., Language disorders: From infancy through adolescence, St. Louis: Mosby, 2001.

[3] Olswang, L., Bain, B., "Topics in Language Disorders," *Intervention issues for toddlers with specific language impairments.,* vol. 11, no. 4, pp. 69-86, 1991.

[4] Michael, D., Chen, S., Serious games: Games that educate, train, and inform, Boston: Thompson Course Technology, 2006.

[5] University, Cambridge, "Cambridge Dictionaries Online," 2011. [Online]. Available: http://dictionary.cambridge.org/dictionary/british. [Accessed 20 06 2012].

[6] Owens, R. E., Metz, D. E., Haas, A., Introduction to Commnication Disorders A Life Span Perspective, New York: Pearson, 2006.

[7] Angell, C.A., Language Development and Disorders: A Case Study Approach, Sudbury: Jones & Bartlett Publishers, 2009.

[8] Otto, B. W., Language Development in Early Childhood (Second Edition), Pearson Allyn Bacon Prentice Hall, 2006.

[9] Bloom, L., Lahey, M., Language development and language disorders, New York: Pearson, 1978.

[10] Yavuz, H., Balcı, A., Turan, Ü. D., Turkish Phonology,morphology and Syntax, Eskişehir: Anadolu University, 2006.

[11] Thomason, R., "Rich Thomason's Home Page," 27 03 2012. [Online]. Available: web.eecs.umich.edu/~rthomaso/documents/general/what-is-semantics.html. [Accessed 20 06 2012].

[12] Demirezen, M., "Pragmatics and Language Teaching," *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi,* vol. 6, no. Pragmatics and Language Teaching, p. 181, 1991.

[13] Justice, L. M., Communication Sciences and Disorders: An Introduction, New Jersey: Pearson Education, Inc., 2006.

[14] Ege, P., "Farklı Engel Gruplarının İletişim Özellikleri ve Öğretmenlere Öneriler," *Ankara Üniversitesi Eğitim Bilimleri Fak. Özel Eğitim Dergisi,* vol. 7, p. 2, 2006.

[15] Ege, P., "Türkçe'deki Ünsüzlerin edinimi: Bir norm çalışması," *Türk Psikoloji Dergisi,* vol. 25 (65), pp. 16-34, 2010.

[16] Ege, P., "MLU as a tool for syntactic and morphological assessment in Turkish," in *Communication Disorders in Turkish (within the series of "Communication Disorders across Languages")*, 2010.

[17] De Kuthy, K., "The Department of Linguistics at the University Tübingen," 2001. [Online]. Available: http://www.ling.ohio-state.edu//~kdk/201/autumn01/slides/morphology-4up.pdf. [Accessed 11 06 2012].

[18] Aksu-Koç, A., Slobin, D. I., The acquisition of Turkish. Dan I. Slobin, (Ed.), The Crosslinguistic Study of Child Language, V.1., Hillsdale, N.J: Lawrence Erlbaum Assoc., 1985.

[19] Topbas, S. S., Ciyiltepe, M., Dil ve Kavram Gelişimi, Ankara: Kök Yayıncılık, 2010.

[20] Cagatay, M., Ege, P., Tokdemir, G., Cagiltay, N. E., "A Serious Game for Speech Disorder Children Therapy," in *2012 7th International Symposium on Health Informatics and Bioinformatics (HIBIT 2012)*, Nevşehir, 2012.

[21] Pentiuc, S. G., Tobolcea, I., Schipor, O. A., Danubianu, M., Schipor, D. M., "Translation of the Speech Therapy Programs in the Logomon Assisted Therapy System," *Advances in Electrical and Computer Engineering,* vol. 10,

no. 2, pp. 48-52, 2010.

[22] Murray, T.G., Parker, V., "Integration of Computer-based Technology into Speech Language Therapy," Valdosta State University, Georgia, 2004.

[23] Ryan, R. M., Deci, E. L., Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being, 2000.

[24] Umanski, D., Kosters, W., Verbeek, F., Schiller, N. O., "Integrating computer games in speech therapy for children who stutter," in *1st Workshop on Child, Computer and Interaction*, Crete, 2008.

[25] Schipor, D., Pentiuc, Ş., Schipor, O., "End-User Recommendations on LOGOMON, a Computer Based Speech Therapy System for Romanian Language," *Advances in Electrical and Computer Engineering,* vol. 10, no. 4, pp. 57-60, 2010.

[26] Locutour, "Learning Fundamentals - Software For Language, Learning and Speech," LocuTour, 2012. [Online]. Available: http://www.locutour.com/. [Accessed 21 10 2012].

[27] Abt, C. C., Serious Games, New York: Viking Compass, 1986.

[28] Zyda, M., "From Visual Simulation to Virtual Reality to Games," *IEEE Computer - COMPUTER,* vol. 38, no. 9, pp. 25-32, 2005.

[29] Susi, T., Johannesson, M., Backlund, P., Serious games: An overview, 2007.

[30] Antonopolous, N., Serious Games and Edutainment Applications, New York, 2011.

[31] Gee, J. P., What video games have to teach us about learning and literacy, New York: Palgrave Macmillan, 2003.

[32] Rampnoux, O., Alvarez, J., Jessel, J. P., Lara, G. M., New advertising tools : Edumarket game, 2006.

[33] Bogost, I., Ferrari, S., Schweizer, B., Newsgames Journalism at Play, London: The MIT Press, 2010.

[34] Grawemeyer, B., Cox, R.; Lum, C., "A Knowledge-based System for speech-therapeutic auditory discrimination exercises," in *Medical Infobahn for Europe: Proceedings of MIE2000 and GMDS2000*, Amsterdam, IOS Press, 2000, pp. 568-572.

[35] Marfisi-Schottman, I., George, S., Tarpin-Bernard, F., "Tools and Methods for Efficiently Designing Serious Games," in *4th Europeen Conference on Games Based Learning ECGBL2010*, Copenhagen, 2010.

[36] Nickerson, R. S., Kalikow, D. N., Stevens, K. N., Computer-Aided Speech Training for the Deaf Bolt Beranek and Newman Inc., Massachusetts: Cambridge, 1976.

[37] Eriksson, E., Bälter O., Engwall, O., Öster, A.M., Sidenbladh-Kjellström, H., "Design Recommendations for a Computer-Based Speech Training System Based on End-User Interviews," in *SPECOM 2005*, Patras, 2005.

[38] Adams, F.R., Crepy, H., Jameson, D., Thatcher, J., "lBM Products for Persons with Disabilities," in *Proceedings of the Global Telecommunications Conference and Exhibition (GLOBECOM'89)*, New York, 1989.

[39] Öster, A. M., Presentation of a new EU project for speech therapy: OLP (Ortho-Logo-Paedia), 2002.

[40] Soransen, B. H., Meyer, B., Serious Games in language learning and teaching – a theoretical perspective, 2007.

[41] Derryberry, A., "Adobe Systems Incorporated," 2007. [Online]. Available: http://www.adobe.com/products/director/pdfs/serious_games_wp_1107.pdf. [Accessed 22 05 2012].

[42] Djaouti, D., Alvarez, J., Jessel, J.P., "Classifying Serious Games: the G/P/S model," in *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches*, Ireland, IGI

Global, 2011, pp. 118-136.

[43] Rodriguez, A. G., Gonzalez Moreno, J. C., Rodriguez Martinez, F. J., "The Role of AOSE on Serious Games," in *VII Agent-Oriented Software Engineering Technical Forum*, Paris, 2011.

[44] Harteveld, C., Triadic game design: balancing reality, meaning and play, London: Springer, 2011.

[45] Winn, B., The Design, Play, and Experience Framework, in Handbook of Research on Effective Electronic Gaming in Education, Philadelphia: IGI Global Publication, 2009.

[46] Wijers, A., Playing games is serious fun! : a serious game for type 2 diabetes patients to enhance a healthy lifestyle, 2009.

[47] Harris, J., The effects of computer games on young children – a review of the research, London: Great Britain, Home Office, Research, Development and Statistics Directorate, 2001.

[48] Nwana, H. S., "Intelligent Tutoring Systems: an overview," in *Artificial Intelligence and Education; Vol. 1: Learning Environments and Tutoring Systems*, Norwood, Ablex Publishing Corp., 1990, pp. 183-201.

[49] Ong, J., Ramachandran, S., "Intelligent tutoring systems: The what and the how," 2000. [Online]. Available: http://www.stottlerhenke.com/papers/astd_learning_circuits_2000_its_what_how.pdf. [Accessed 12 04 2012].

[50] Helander, M., Landauer, T. K., Prabhu, P., Handbook of Human-Computer Interaction, Chapter 37, North-Holland: Elsevier Science B. V., 1997.

[51] Mall, R., Fundamentals of Software Engineering, 3rd ed., Sonepat, Haryana: Rajkamal Electric Press, 2009.

[52] Pressman, R. S., Software Engineering: A Practitioner's Approach, Pressman

and Associates, 2005.

[53] Abrahamsson, P., Salo O., Ronkainen, J., Warsta, J., Agile Software development methods: Review and Analysis, Vuorimiehentie: VTT Electronics [Online] Available at: http://www.pss-europe.com/P478.pdf, 2002.

[54] Awad, M. A., "A Comparison between Agile and Traditional Software Development Methodologies," The University of Western Australia, Australia, 2005.

[55] Taya, S., Gupta, S., "Comparative Analysis of Software Development Life Cycle Models," *International Journal on Computer Science and Technology (IJCST),* vol. 2, no. 4, pp. 536-539, 2011.

[56] Aggarwal, K. K., Singh, Y., Software Engineering (3rd ed.), New Age International, 2007.

[57] Green, D., DiCaterino, A., "A Survey of System Development Process Models," University At Albany, Albany, 1998.

[58] Michelet, A., El maestro y el juego. Perspectivas V. XVI. No. 1, Paris, 1986.

[59] Bayırtepe, E., Tüzün, H., "The effects of game-based learning environments on students' achievement and self-efficacy in a computer course," *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi,* vol. 33, pp. 41-54, 2007.