Quantitative evaluation framework of the X-TEC model

Paula Escudeiro[1], José Bidarra[2]

[1]Departamento de Engenharia Informática, Instituto Superior de Engenharia do Porto,
[2]Departamento de Ciências Exactas e Tecnologia, Universidade Aberta

Key words: *Computer based learning, educational software development, instructional design, learning environment, educational software evaluation*

Abstract:

This paper outlines the way in which the X-TEC (Techno-Didactical Extension for Instruction/Learning Based on Computer) model is used in the development of educational software in order to strengthen the potential quality of e-Learning systems.
The aim of the present paper is to describe the deployment phases of the X-TEC model and its global evaluation. For the evaluation of our educational software systems based on X_TEC model we propose a generic quantitative evaluation framework.

1 Introduction

Modern information and communication technology renders new ways of organizing teaching and learning processes, which are independent of time and space.
E-learning has been described as the ´use of electronic technology to deliver, support and enhance teaching and learning` [Learning Technologies, 2003].

Despite the theoretical benefits that e-learning systems can offer, difficulties can often occur when systems are designed with out consideration of learner's characteristics [Frredman and Liu, 1996; Liang and McQueen, 1999].

In general, educational software systems are based on methodological approaches which are fundamentally concerned with processes or data. Their lifecycle is supported on two different and independent stages: instructional design and technical development.

To evaluate the educational software systems based on X_TEC model we propose a generic Quantitative Evaluation Framework (QEF). This framework may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between different tools.

The Quantitative Evaluation Framework (QEF) and the Techno-Didactical Extension for Instruction/Learning Based on Computer (X-TEC) model are based on the paradigms of software engineering applied to the construction of educational software.

The Quantitative Evaluation Framework evaluates the Educational software quality on a three dimensional space. Each dimension aggregates a set of factors. A given factor is a component that represents the system performance from a particular point of view.

The quality of a given system is defined in a tri-dimensional Cartesian quality space and measured, in percentage, relatively to a hypothetically ideal system, represented in our quality space.

This Quantitative Evaluation Framework may also be applied to evaluate other educational software allowing for a comparison between different tools.
This orientation is very important due to the high quality demand placed upon educational systems.

2 Model structure

The X-TEC model presents two overlapping extensions: instructional model and learning environment. This model promotes an interaction between these two extensions, allowing the deployment of a common development platform, represented in fig.(1).
This platform has quality factors settled on a multifaceted conception described by a set of internal and external factors.

2.1 The X_TEC platform and structure

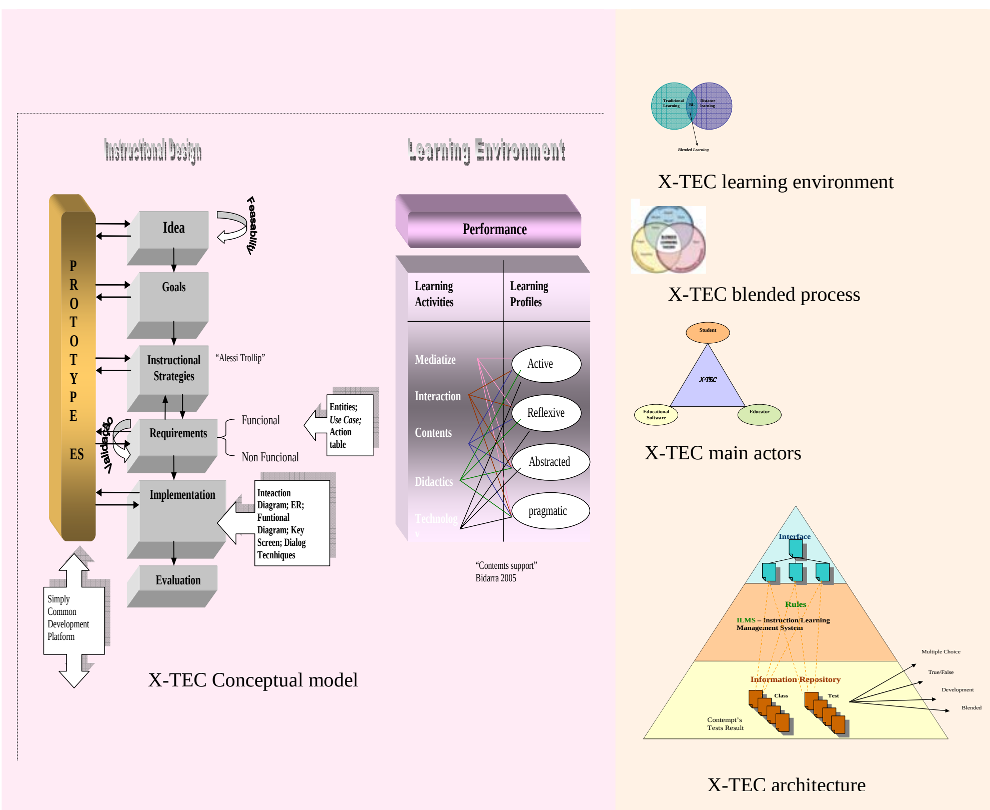
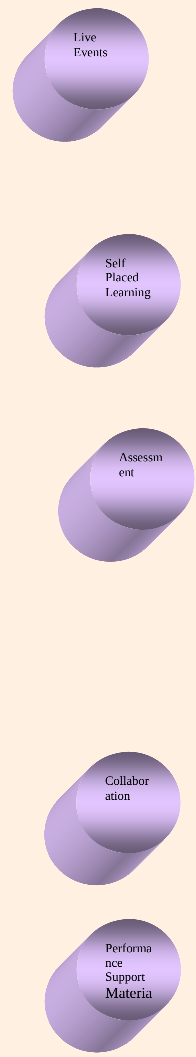


Fig. 1: X-TEC platform

X-TEC model should leverage the educator’s creativity and make possible a better vision of Information Technology on Educational Systems, guaranteeing a new related perspective which is represented in the following structure:

Idea;	Is it a good idea to be implemented?
Objectives;	target id; Support materials; Instruction; Evaluation
Didactics strategies id	(based on Allessi Trolip strategies);
Planning	activities chronogram
Cost/benefit estimate	Estimate the cost/benefit of the educational software
Requirement analysis	(functional and non functional) Implementation feasibility; Development model; Metaphor id; Actor’s id; Action table;
Educational software architecture;	Three tier architecture.
Implementation;	Data base structure id; Relational data model; Interaction diagram; Screen architecture; Key screen; Dialogue id; Educational software prototype;
Educational software evaluation	evaluate the educational software quality



2.2 The X_TEC intervenient

The development team proposed on X-TEC model is composed by two elements: Educational System Annalist (ESA)/Subject Specialist (SS) and Informatics Specialist (IS).
During the development phase a new element may come by: Designer/Audio visual Consulter (AVC)
The role of the development team is organized as follows:

ESA/CS - Prototype developer
IS - implement explicit messages from educational software scenario.

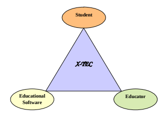
AVC - implement implicit messages from scenario.

Implicit messages: trigger student's reactions; for instance, color, object screen positioning, scenario presentation, images, sound, moving arrow around screen.

Explicit messages: sent by the program and explicitly activated by the student; for instance, menu options.

The consistent use of implicit messages as a complement of explicit messages will enable a better interaction between the student and the educator and between these and the educational software.

2.3 The X_TEC main actors



The main actors on X-TEC model are the student, the program and the educator.

Student uses the educational software for learning the subjects. The students choose the learning environment adapted to their personal characteristics. They should be in control of the system.

Educator is the educational system supervisor

Educational Software is the knowledge detainer and adviser.

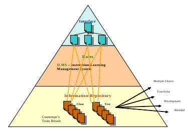
2.4 The X_TEC learning environment

The learning environment on X-TEC model is based on blended learning.The term blended learning is used to describe a solution that combines several different delivery methods, such as web-based courses and knowledge management practices. It is used to describe learning that mixes various event-based activities, including face-to-face classrooms, live e-learning, and self-placed learning. "Learning circuits- American Society for Training & Development". Learning theories of Keller, Gagné, Bloom, Merrill, Clark and Gery identify five important elements emerging from a blended process:

Live Events 1	Synchronous, instructor-led learning events in which all learners participate at the same time, such as in a live "virtual classroom."
Self-Paced Learning 2	Learning experiences that the learner completes individually, at his own space/rythm and in his own time, such as interactive, Internet-based or CD-ROM training.
Collaboration 3	Environments in which learners communicate with others, for example, e-mail, threaded discussions or online chat.

Assessment	A measure of learners' knowledge. Pre-assessments can come before live or self-paced events to determine prior knowledge, and post-assessments can occur after live or self-paced learning events to measure learning transfer.
Performance Support Materials	On-the-job reference materials that enhance learning retention and transfer, including PDA downloads and printable references, summaries and job aids.

2.5 The X_TEC architecture



The X-TEC model is supported by a three tiered architecture [Eckerson 95]: User Interface, Rules and Information Repository, according to fig.5.
The three tier architecture is used to provide increased performance, flexibility, maintainability, reusability and scalability while hiding the complexity of distributed processing from the end user.

1st Tier: Interface	It is related with the scenario identification, synchronous and asynchronous communication technologies and implicit and explicit messages. This following elements act on this tier: Educational Software; Content Specialist's and Designers.
2nd Tier: Rules	It is related with the virtual abstracted organization of the content. The intervenient on the Rules tier is: ILMS – Instruction/Learning Management System
3rd Tier: Information Repository	It will allow for all the contents, rules and interface specifications being stored on a warehousing platform. The intervenient is: Data Base Management Functionality.

The ESA has to fit the educational software in one of these didactical strategies. This identification will enable the classification of the educational software into two classes: consultation (class 1) and evaluation (class 2).

The definition of these two main groups obeying specific orientation guides conducts the ESA to choose a learning strategy (learning alone or learning by retroaction) for the educational software.
The ESA plays the role of moderator between the student and the educational software in this learning process.

The ESA should easily be able to evaluate the student learning process, appealing when needed to the information stored in the educational software database.

Three tier architectures facilitate educational software development because each tier can be built and executed on a separated platform, thus making it easier to organize the implementation.

3 Quantitative Evaluation Framework

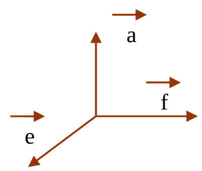
X-TEC model is supported by software engineering goals, principles and actions [Pressman, 2001], [Bates, 2000].

To evaluate the educational software systems based on X_TEC model we propose a generic Quantitative Evaluation Framework (QEF) [Paula Escudeiro, José Bidarra, 2006]. This framework may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between different tools.

Educational software quality (ISO 9126 is the standard of reference) [Scalet et al, 2000] is evaluated on a three dimensional space.

A dimension aggregates a set of factors. A factor is a component that represents the system performance from a particular point of view.

The dimensions of our Cartesian quality space are: Functionality (F); Efficiency (E) and Adaptability (A), represented in fig 1.



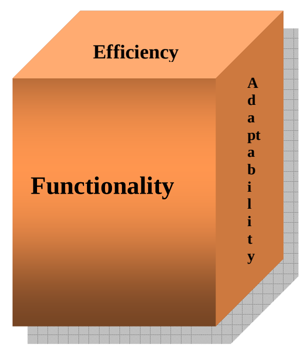


Fig 1: Cartesian quality space

For the evaluation of educational software systems based on the X_TEC model we propose a generic quantitative evaluation framework. This framework may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between distinct tools.

The quality q, of a given system is defined in our tri-dimensional Cartesian quality space, Q, and measured, in percentage, relatively to a hypothetically ideal system, I, represented

in our quality space by the coordinates (1, 1, 1).

3.1 Quality dimensions

The quality space, *Q*, aggregates, in the dimensions – *Functionality; Efficiency and Adaptability* – a set of factors that measure the relevant characteristics of an ESDM.

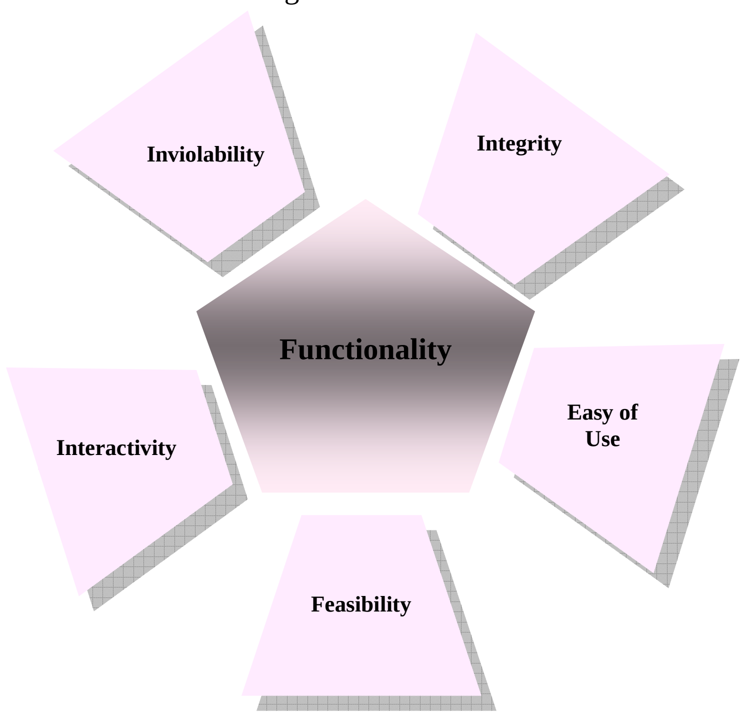
The *Functionality* dimension reflects the characteristics of the educational software related to its operational aspects. It aggregates four factors: feasibility, inviolability, easy of use and integrity

The *Efficiency* dimension aggregates four factors: data structure, programming structure, learning objects, imperfections recovery.

Through this dimension we measure the system's ability for presenting different views on the course content with minimum effort.

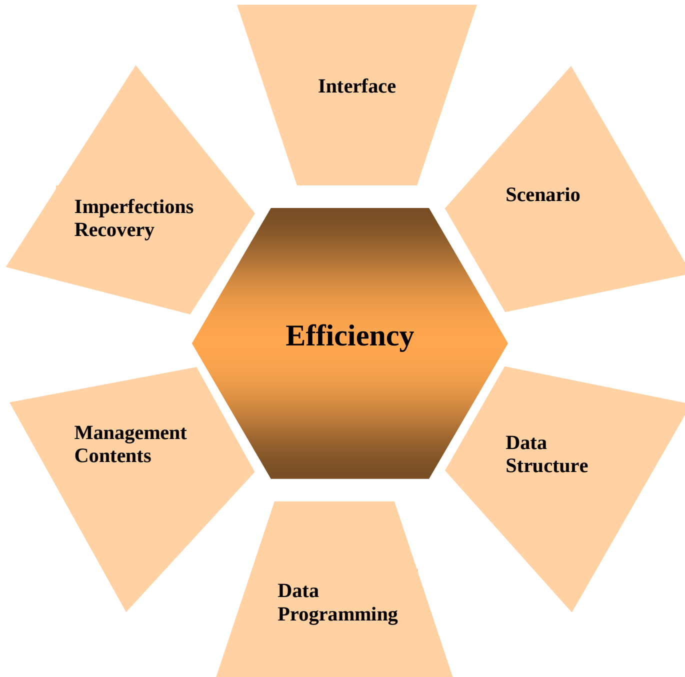
The *Adaptability* dimension is the aggregation of five factors: flexibility modularity, reusability, scalability and maintainability. Through them we can measure to what extend the scenario and course content are efficacious – whether they are focused and able to present different instructional design theories and different learning environment in a common platform.

The quality for a given system coordinates may be obtained through the application of one of several aggregation forms. We will compute these coordinates as the average of the factors that contribute to it; the average is simple and gives the same relevance to all factors. Quality dimensions are based on the following factors:

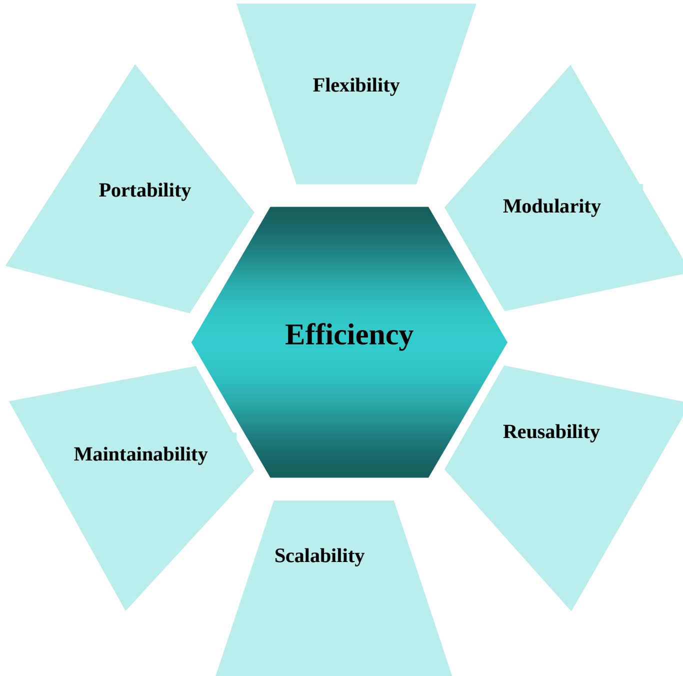


Dimension	Factor	Requirement examples
Functionality	Feasibility	What is the cost structure of each technology? What is the unit cost per student? How quickly can courses be mounted with this technology? How quickly can materials be changed?
	Easy of use	What are the institutional requirements, and barriers to be removed before this educational software can be used successfully? What changes in the institution need to be made? Does the interaction this technology enables is easy to use?
	Integrity	Conceptual integrity: Does the models remain true to the concept of "objects"?

	Inviolability	
	Interactivity	What kind of interaction does this technology enable?



Dimension	Factor	Requirement
Efficiency	Scenario	Implicitly and explicitly messages
	Data Structure	
	Programming Structure	
	Management Contents	
	Imperfections Recovery	
	Interface	Key screen; screen architecture



Dimension	Factor	Requirement
Efficiency	Scenario	Implicitly and explicitly messages
	Data Structure	
	Programming Structure	
	Management Contents	
	Imperfections Recovery	
	Interface	Key screen; screen architecture

For each system being developed we will have to identify the importance of each factor to the dimension. The dimension coordinate is them computed as the weighted mean of these factors:

$$\text{Dimension}_i = \sum_n (p_n \times factor_n), \quad \sum_n (p_n) = 1 \text{ and } \quad p_n \in [0,1]$$

Where:

n is the number of relevant factors for the dimension.

Each factor is evaluated by:

$$\text{Factor}_n = \frac{1}{\sum_m pr_m} \times \sum_m (pr_m \times pc_m)$$

Where:

M is the number of valid requirements for the factor.

pr_m is the weight of the requirement m

pc_m is the fulfillment percentage of the requirement m.

The dissimilarity between the system under evaluation and ideal system is measured by:

$$D = \sqrt{\sum_j \left(1 - \frac{Dim_j}{100}\right)^2}$$

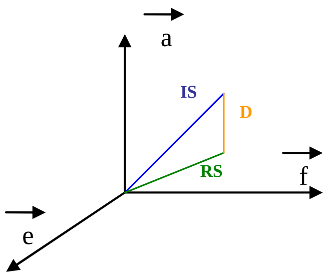
Finally the quality of the system is computed as:

$$Q = 1 - \frac{D}{\sqrt{n}}, \quad Q \in [0,1]$$

or

$$q = \left(1 - \frac{D}{\sqrt{n}}\right) * 100, \quad q \in [0,100]$$

The quality of a system is measured as the distance between the ideal system (projected system) and the real system (final system).



The system quality is in the inverse proportion of the distance between the Ideal System (IS) and the Real System (RS).

If D=0 Then Q=1

If D=maxim, $D_{max} = \sqrt{n}$ Then Q=0

The measure of the system quality is obtained from a six steps process:

1st – Requirement classification

2nd – Factor classification

3rd – Result evaluation

4th – Dimension performance

5th – Global deviation

6th – System quality

3.1.1 Requirement Classification

The ideal system has a set of requirements that indicates what the system must do.

We start by associating weights to requirements, [0,1] based on the relevance of the requirement for that particular dimension, according to:

10 – Fundamental

8 – Very Important

6 – Important

4 – Necessary

2 – Optional

0 – Irrelevant

Dimension

Requirements

Functionality	Efficiency	Adaptability
Pr n	Pr n	Pr n
Pr n	Pr n	Pr n
SUM()	SUM()	SUM()

Fig 1: matrix of the dimension requirements

3.1.2 Factor Classification

Each factor contributes to the dimension value. This contribution is represented by a real number, P_n , between 0 and 1, indicating the relevance of the factor to the dimension. The dimension value is a weighted mean the factor that contributes to that dimension

$$\text{Dimension} = \sum_n (p_n \times factor), \quad \sum_n (p_n) = 1 \text{ and } \quad p_n \in [0,1]$$

3.1.3 Result Evaluation

It is very important to validate the requirements, so that system performance can be accurately evaluated.

The matrix in fig 2 shall be fulfilled during the evaluation process. Once it is completed the system quality is automatically computed.

Dimension

Requirements

Factors		
Pcm	Pcm	Pcm
Pcm	Pcm	Pcm

Fig 2: matrix of the factors

3.1.3 Dimension performance

The performance of a dimension is obtained through, the factors of each dimension.

$$\text{Factor}_{n} = \frac{1}{\sum_{m} pr_m} \times \sum_{m} \left(pr_m \times pc_m \right)$$

And the dimension performance is given by:

$$\text{Dimension} = \sum_{n} \left(p_n \times factor \right), \quad \sum_{n} \left(p_n \right) = 1 \text{ and } p_n \in [0,1]$$

3.1.4 Global deviation

The global deviation is obtained as the Euclidean distance between our system coordinates and the ideal system, whose coordinates are (1,1,1)

$$D = \sqrt{\sum_{j} \left(1 - \frac{Dimj}{100} \right)^2}$$

Global deviation

3.1.5 System Quality

The system quality is computed by:

$$Q = 1 - \frac{D}{\sqrt{n}}, \quad Q \in [0,1]$$

$$q = \left(1 - \frac{D}{\sqrt{n}} \right) * 100 \quad q \in [0,100]$$

System Quality

We say that system quality is q% which means that the system is able to perform q% of its initial specifications.

4 Conclusions

In this work we propose a method to measure quantitatively the quality of a given educational system.
Quality evaluation frameworks, like the one we propose here, are crucial to help validating educational systems and ensure that they are adequate and follow the original specifications, before using them in the learning environment.

We are already applying X_TEC, for the development of educational software systems with our students, and using the quality evaluation framework to evaluate them. Our purpose is to realize the ability and applicability of our quantitative evaluation framework in real world solutions.

The QEF may also be applied to evaluate other Educational Software Development Models (ESDM), allowing for a direct comparison between different tools.

References:

[1] A. Dias Figueiredo (2005)	Context Engineering: And is Development Research Agenda, Universidade Coimbra, 2005
[2] Allesi, S. e Trollip (1985)	S. Computer Based Instruction: Methods and Development, Prentice Hall, Inc. Englewood Cliffs, New Jersey, USA, 1985
[3] Bates Tony, (2000)	A. W. Tony Managing Technological Change: Strategies for College and University Leaders. San Francisco, 2000
[4] BloomBertram B. Mesia, and David R. Krathwohl, (1964)	Bloom B. S. Taxonomy of Educational Objectives,: The Affective Domain & The Cognitive Domain. New York: David McKay Co Inc.
[5] Booch, (1994).	G. Booch, Object Oriented Analysis and Design With Applications, Second Edition, Benjamin/Cummings, Menlo Park, Califórnia, 1994.
[6] Clark 1994, Gery, (1994)	Gery, GJ Making CBT happen. Boston: Weingarten. Clark, RE Media will never influence learning. Educational Technology Research and Development
[7] Coad and Yordon, (1991)	P. Coad and E. Yourdon, OOA –Object Oriented Analysis, 2nd Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1991
[8] Crossley, K. e Green (1990)	Le Design des Didacticiels: Guide Pratique pour la Conception de Scénarios Pédagigiques Interactifs. ACL-Editions, Paris France 1990
[9] Eckerson (1995)	Eckerson, Wayne W. "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications." *Open Information Systems 10,* 1 (January 1995): 3(20).
[10] Escudeiro, Paula; Bidarra José, (2006)	X-TEC: Techno Didactical Extension for Instruction/Learning Based on Computer, Orlando, Florida, SITE 2006. Educational Software Engineering Models, Florida, EISTA 2006
[11] Gagné, (1996)	Gagné, Robert M. and Medsker, Karen L. (1996). *The Conditions of Learning Training Applications.* Florida: Harcourt Brace & Company.
[12] Jacobson, (1992)	Jacobson, Object Oriented Systems Engineering, Addison-Wesley, 1992.
[13] Keller/Back (2004)	Keller, M.; Back, A.: Blended-Learning-Projekte im Unternehmen, Learning Center der Universität St. Gallen, St. Gallen, 2004
[14] Merrill, (1981)	Merrill, M.D., Kowallis, T., & Wilson,

	B.G. Instructional design in transition. In F. Farley & N. Gordon (Eds.), Psychology and education: The state of union
[15] Minken, I., Stenseth, B. E Vavik L., (1998)	Educational Software. ULTIMA-Gruppen A/S, Haden, Norway, 1998
[16] Pressman Roger S., (2001)	Pressman, Roger S. Software Engineering a Practitioner’s Approach, 5 th Edition, McGraw-Hill Companies Inc, 2001
[17] Purinima Valiathan, (2005)	ASTD-Linking People, Learning & Performance. Learning circuits- American Society for Training & Development
[18] Rumbaugh et al, (1991)	J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object Oriented Modeling and Design*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
[19] Scalet et al, 2000	ISO/IEC 9126 and 14598 integration aspects: The Second World Congress on Software Quality, Yokohama, Japan, 2000.
[20] Yourdon, (1998)	E. Yourdon, *Managing the System Life Cycle*, 2nd Edition, Yourdon Press/prentice Hall, Englewood Cliffs, New Jersey, 1998.