Chinese Business Review, ISSN 1537-1506
January 2012, Vol. 11, No. 1, 128-137

Educational Software Development Life Cycle Stages*

Salah Alkhafaji, B. Sriram

Sur University College, Sur, Sultanate of Oman

Technological innovations have revolutionized the educational technology into various dimensions. Educational processes without educational technology have no value in this modern world. In education domain, the educational software has simplified the processes in greater extend. A proper development methodology has to be implemented while developing such educational software. In particular, the software developed to enrich these education processes should follow a development strategy to motivate the end users to utilize the hypermedia potentials. The software development life cycle (SDLC) has different phases in designing such educational technology and assists the end users to benefit from the modern technology. This study identifies the various factors to be considered at each phase of the SDLC while developing educational software. Also, this study proposes some suggestions to be followed in ESDLC with respect to educational processes perspectives. The core idea of this study is to identify the various issues in implementing such educational software in day to day teaching and learning processes.

Keywords: software development life cycle (SDLC), educational technology, teaching and learning processes, technology innovations, educational software

Introduction

General Introduction

Information technology and other technological innovations rationalized the education processes drastically, in past decades. Technological innovations substantially increased the educational processes into various degrees. Modernized tools and techniques eased the teaching and learning processes in some cases. Various conceptual models are developed using this educational technology.

As the role of educational technology has become an essential tool in this emerging world, there is a need to implement appropriate development methodologies while developing such educational technology. The educational software should support and enhance teaching and learning processes to meet the required outcomes. Based on the challenges faced by the educators and learners in their educational processes, these technologies should be reviewed and updated.

* This paper is updated version of Alkhafaji, S., & Sriram, B. (2011). Using SDLC in educational hypermedia development. *International Conference on Business and Information* held on July 4-6, 2011, Bangkok, Thailand. Proceedings of Business and Information, 8 (ISSN: 1729-9322).

Salah Alkhafaji, Ph.D., Assistant Dean for Students and Admin Affairs, Sur University College.

B. Sriram, M.Sc., B.Ed., D.C.A., Lecturer and Quality Assurance Officer, Sur University College.

Correspondence concerning this article should be addressed to Dr. Salah Alkhafaji, Sur University College, PO: 440, PC: 411, Sur, Sultanate of Oman. E-mail: srishan11@gmail.com.

The effectiveness of educational technology in educational processes depends on the proper and timely usage. For this, the educational technology depends on the development methodologies. Thus, the educational software should be developed systematically using software development life cycle (SDLC). The requirements' factors need to be analyzed accurately to identify the end users' necessities.

This study proposes the factors which have to be considered at each stage of SDLC while developing educational software. Also, this study identifies the major issues of using SDLC in educational technology development processes from educators and learners perspectives.

Purpose of Study

The main purpose of this study is to identify the major factors influencing day to day teaching and learning processes, those which have to be considered while developing educational software. The goal of this study is to propose some suggestions to be followed while SDLC phases in order to utilize the maximum potentials of educational technology in teaching and learning processes.

Research Questions

To conduct this study, the following research questions were formed and analyzed:

How effectively SDLC shall be applied in educational software development processes?

What are the major factors to be considered while developing educational software with respect to educators and learners perspectives?

How the major factors that influence teaching and learning processes shall be integrated into SDLC?

Does currently available educational software support teaching and learning processes in all aspects?

Limitations

Software development methodologies only have been considered for this study purpose. The educational software usage in teaching and learning has been emphasized in this study. This may further extended to the various other environmental factors that influence the educational processes. Also, this may be extended to the other educational technologies used in teaching and learning processes with respect to educators and learners perspectives.

Literature Review

Teachers cannot hope, any more than software designers can, that the design of the new medium itself will strongly influence the directions in which students put forward their effort, regardless of the assessment schemes put in place or the resource limitations under which students work. When weaving collaborative hypermedia into classroom practice, success will depend on the attention paid to the variety of countervailing forces which can work against the expressed and implied intents of the instructional designer (O'Neill, Edelson, Louis, & D'Amico, 1995).

Hypertext and hypermedia are extensively used in educational setting, as they allow varied subjects to be related in numerous ways. Different learning and teaching methodologies can be applied to reach the students easily. Such software can be developed with different strategies with various interfaces. The course syllabus, assessment methods and available resources may be included to make the online classroom environment more constructive. The course materials may also be included to assist the students in their learning processes. The software may contain the major part of the help features to make the environment understandable (Dix, Finlay, Abowed, & Russell, 2008).

Hypermedia technologies must provide maximum amount of resources and supporting materials to effectively improve teaching/learning processes. It should be interactive and attractive. As hypermedia facilitates the complex learning processes involving a large range of activities, they should be used in the classrooms to provide adequate understanding of the subjects to the students. Irrespective of the students' level, all the students should get benefits out of the hypermedia technology. Hypermedia should self-motivate the student in his learning processes. The hypermedia should provide timely, helpful information to both educators and learners (Salah & Sriram, 2010).

After feasibility is established, the software development life cycle (SDLC), a subcycle of the SPLC, begins. This subcycle typically includes phases for analysis, conceptual design, design, implementation, testing, and installation and checkout. SDLC end is signaled by delivery of an operational application (Sue Conger, 2008).

A software process model is an abstract representation of a software process. The Waterfall model, evolutionary development and component based software engineering are the three generic process models widely used in current software engineering. They are not mutually exclusive and are often used together, particularly in large system development. All sorts of variants of three generic processes have been proposed and used in some organizations (Sommerville, 2009).

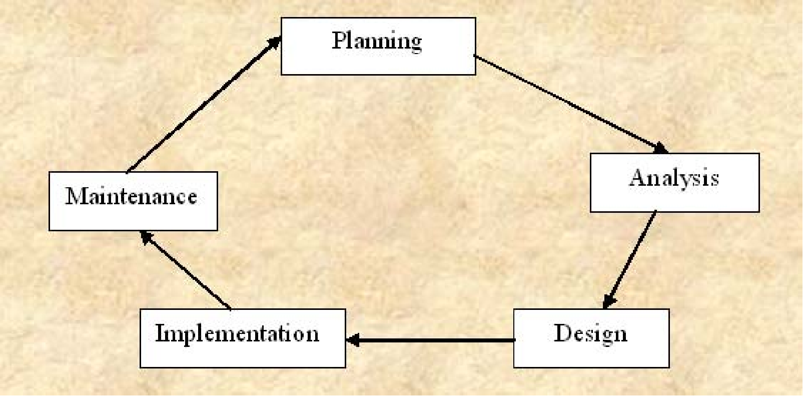

Figure 1. Systems development life cycle. Source: Modern Systems Analysis and Design (Hoffer, Joey, & Valacich, 2009).

Most organizations find it beneficial to use a standard set of steps, called a systems development methodology, to develop and support their information systems. The Systems Development Life Cycle (SDLC) is a common methodology for systems development in many organizations; it features several phases that mark the progress of the systems analysis and design effort (Hoffer et al., 2009). Figure 1 shows a sample SDLC.

A software development process, also known as a software development lifecycle, is a structure imposed on the development of a software product. Similar terms include software life cycle and software process. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Some people consider a lifecycle model as a more general term and a software development process as a more specific term. For example, there are many specific software development processes that "fit" the spiral lifecycle model (Wikipedia article).

Designing and building a scaled-down but functional version of a designed system is known as prototyping. A prototype can be built with any computer language or development tool. Using prototyping as a development technique, the analyst work with the users to determine the initial or basic requirements of the system (Hoffer et al., 2009)

Research Methodology

Data Collection

The secondary data were analyzed for this study purpose. The data collected for “educational hypermedia development methodologies” by the same authors were analyzed with respect to this study. Also, the research works conducted by various authors were also studied and analyzed.

A questionnaire was prepared with 20 closed end questions and five open end questions. Also, 10 demographic and five hypermedia knowledge related were also asked in the questionnaire. The questionnaire was distributed to the academic and administrative staff members of Sur University College, College of Applied Sciences and Nursing College, Sultanate of Oman. Seventy two responses were received and analyzed for this purpose.

Data Analysis

The data collected were analyzed with respect to various factors. It was observed that 58% of the respondents were from he educators, 24% of the respondents were students, 10% of the respondents were general users of technology and 8% of the respondents were developers. The respondents who were neither educators nor students but either used educational technology during their studies or the respondents who were using educational technology in their work were categorized as “general users”.

Table 1

Knowledge About Educational Technology

Response	Educators	Students	General user	Developers	Total	Total (%)
Excellent	34	3	1	3	41	56.9
Very Good	6	3	2	2	13	18.1
Good	2	6	1	1	10	13.9
Average	0	3	2	0	5	6.9
No Idea	0	2	1	0	3	4.2
Total	42	17	7	6	72	

The respondents were asked about their knowledge about educational software to know their background. Though the knowledge about educational technology cannot be exactly measured and rated, the basic idea about these optional questions was to have an overall opinion about the knowledge of the respondents about the educational software. This helped the researchers to relate their responses in close-end questions with their experience and knowledge. This has been shown in Table 1.

Analysis on Questionnaire

The summarized results on research question were shown in Table 2.

Interpretation:

Thirty one point nine percent and 19.4% of the respondents either disagreed or strongly disagreed that the educational technologies help in active learning with different types and levels of students.

Thirty six point one percent and 30.6% of the respondents either strongly disagreed or disagreed that the currently available educational software follows a proper development methodology.

Twenty six point four percent and 23.6% of the respondents strongly agreed and agreed on the fact that educational technologies create the collaborative learning environment.

Twenty nine point two percent of the respondents stated that constructive methodologies are incorporated in the available educational software to ensure all educators and learners get benefit.

Nearly 71% respondents agreed on the facts changes are required in teaching and learning processes to use available educational software in the classroom.

Table 2

Summarized Results on Research Questions

No.	Area	Strongly agree	Agree	No idea	Disagree	Strongly disagree
1	Educational software helps in active learning with different types and levels of students	17	12	6	23	14
2	Educational technology systems follows a proper development methodologies	8	11	5	26	22
3	Educational technologies create the collaborative learning environment	19	17	10	11	15
4	Constructive methodologies are incorporated in the available educational technology to ensure all educators and learners get benefit	9	12	4	16	31
5	Changes are required in teaching and learning processes to use available educational technologies in the classroom	19	32	5	12	9

Figure 2 depicts the questionnaire analysis graphically.

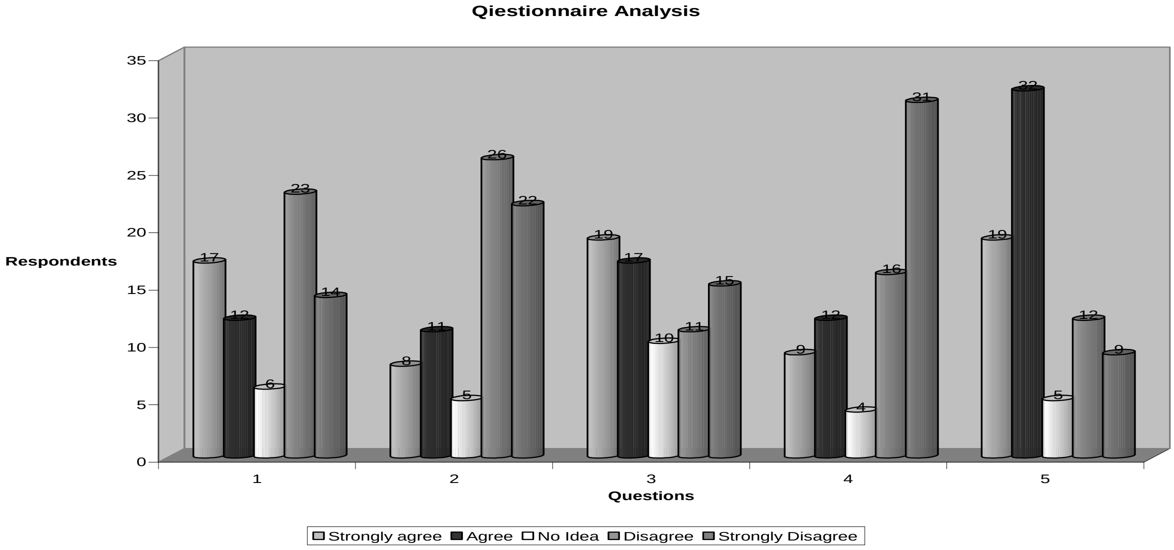

Figure 2. Questionnaire analysis.

Using SDLC in Educational Software Development

Though various models have been followed in software development life cycle (SDLC), the major phases are: analysis, design, implementation and testing and maintenance. The analysis phase requires more input from the users to find the requirements. The user requirements and system requirements need to be identified and analyzed properly in order to get the maximum potentials of educational software. As a preliminary requirement, SDLC shall begin with strategic planning in analysis phase which will help to identify the requirements. The design phase uses the requirements and the fields are defined in analysis phase. The required architectural designs and real time software designs have to be defined. The software has been developed in implementation phase. On completion of the software, it will be tested for integrity and requirements at testing phase. Later, it will be maintained and any further requirements will be suggested under maintenance phase. As a preliminary requirement, SDLC shall begin with strategic planning, as some models suggest.

Analysis Phase

The above results show the following. In analysis phase, various requirements are to be studied and analyzed for designing purposes. In education processes as the educators and learners are the main users. Their requirements have to be collected and specified. Feasibility study shall be conducted and the user requirements and system requirements shall be elicited. Some of the major requirements identified while developing the educational software is as follows.

Educator's requirements. The major requirements of the educators identified are: Course management and data transfer. The educators expect that the proper course coverage should be included in the educational software to provide better service to the students. Also, they suggested that the educational software should provide adequate tools and techniques to transfer the already available data into the systems. An analysis should be conducted to suggest how the educational software shall be designed to assist the educators in course management and data transferring. According to educator's requirements, the educational software must be "complete, reliable, timely and constructive".

Learner's requirements. The major requirements of the learners identified are: Active learning and interaction. The students should be motivated in various ways to learn the course. They need to participate actively in education processes. The educational technology should provide sufficient tools for active learning. Also, the educational software should be designed for easy interaction. According to learner's requirements the educational software should be "simple, clear, effective and comprehensive". Figure 3 shows the users' requirements that have to be considered while developing such educational software.

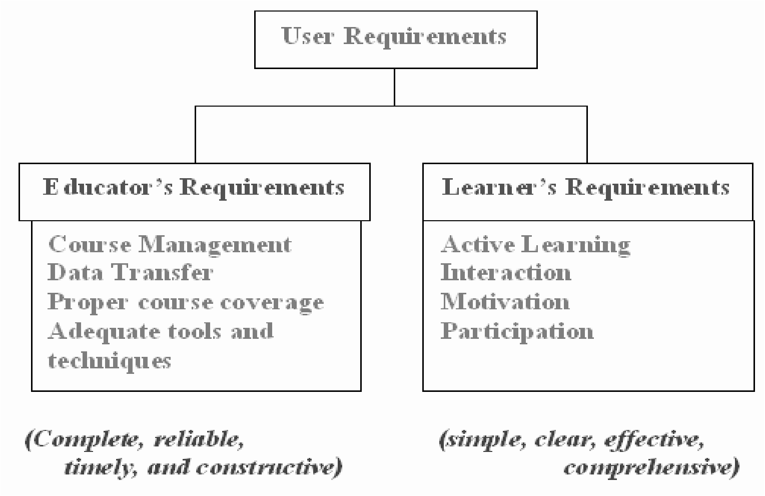

Figure 3. Users' requirements analysis.

Based on the above educational requirements, the requirement determination process shall be studied to replace or to enhance the current system. The system requirements shall be studied according to the inter-relationships and structured (Hoffer et al., 2009). The required system models such as content diagram, data flow diagrams (DFD), class diagrams, entity relationship diagram (ERD), use CASE diagram and other objects and data models shall be structured as required. Based on the above models, the systems catalog shall be developed as dictionary. The basic principle of analysis phase should concentrate on "clarifying, identifying, classifying and structuring the key people requirements". Figure 4 shows the analysis method to be followed while developing education software.

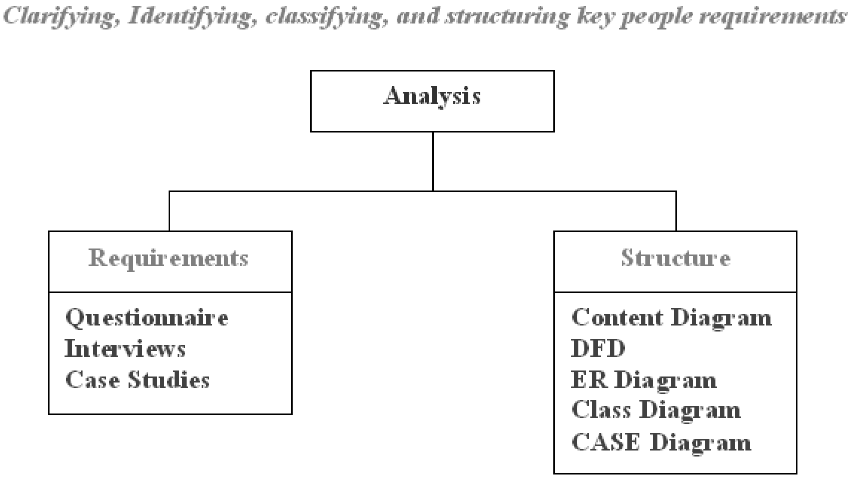

Figure 4. Systems model.

A prototyping system shall be developed and tested with the user agreement in this phase to get the clear view points of the user. As prototype augment the SDLC processes, final system shall be developed based on the prototype specifications (Hoffer et al., 2009).

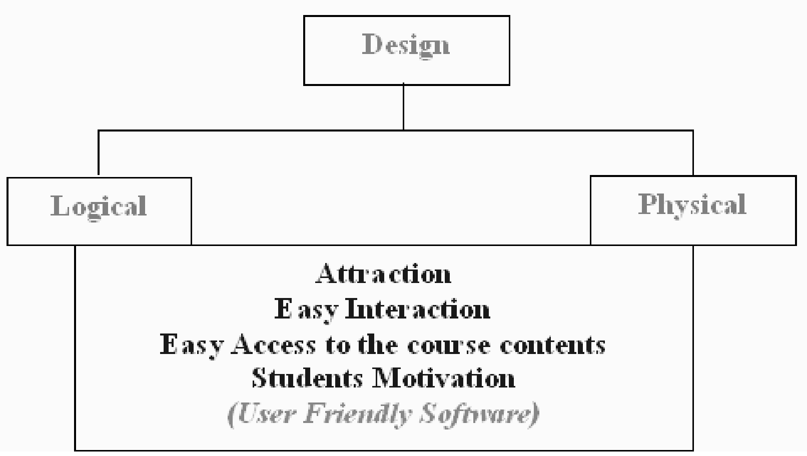

Figure 5. Design phase.

Design Phase

Using the above educators and learners requirements, suitable interface has to be designed. The major factors to be taken care in this phase are: "Attraction, easy interaction, easy access to the course contents and students motivation”. The user interface design principles such as user familiarity, consistency, minimal surprise, recoverability, user guidance and user diversity (Sommerville, 2009) have to be followed in this phase. The logical and physical design of the software should be as “user friendly”. Figure 5 shows the design phase of educational software development.

Implementation Phase

Coding and developing. As this phase concentrates on development strategies, appropriate application programming language (software) has to be selected for the design. Any development strategies may be selected based on the above proposed systems design. Though the educators and the learners are front end users, the software development should also consider their socio-economic ethics other than their general requirements. The implementation phase should consider “economic and compatibility” of the software.

Testing. This phase concentrates on verification and validation of the software developed with respect to end users perspectives. The major factors to be considered in this phase are the educator’s perceptions and learner’s adoption. The verifications have to be compiled with educator’s perceptions on educational technology. Though, all the perceptions may not be met, still maximum degrees of perceptions should be validated. In the same way, learners’ adoption psychology should be measured and validated. The educational software should tested for “reliability, accuracy, user requirement fulfillment and validity”. Figure 6 shows the implementation phase of educational software development process.

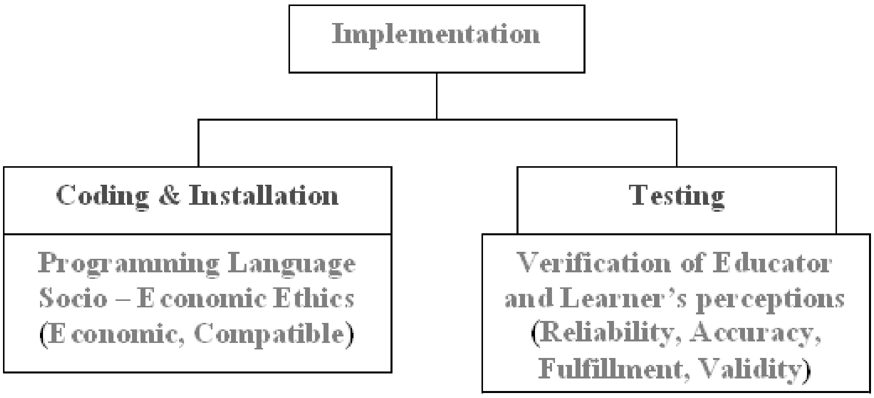

Figure 6. Implementation phase of ESDLC.

Maintenance Phase

The developed hypermedia should be maintained satisfactorily till it has been decommissioned. The end users feedbacks should be taken care in this phase. Necessary analysis shall be conducted for updates and maintenance. During the maintenance phase, the software has to be checked for errors.

Conclusion

The above discussed factors that influence the teaching and learning processes have to be intensively studied and analyzed to integrate them in SDLC. A proper collection tool has to be designed to get the educators’ and learners’ perceptions before developing educational hypermedia. Though, some of the available hypermedia does not support teaching and learning processes in all aspects, using the influencing factors they may be upgraded. Figure 7 explains how effectively SDLC shall be applied in educational software development processes.

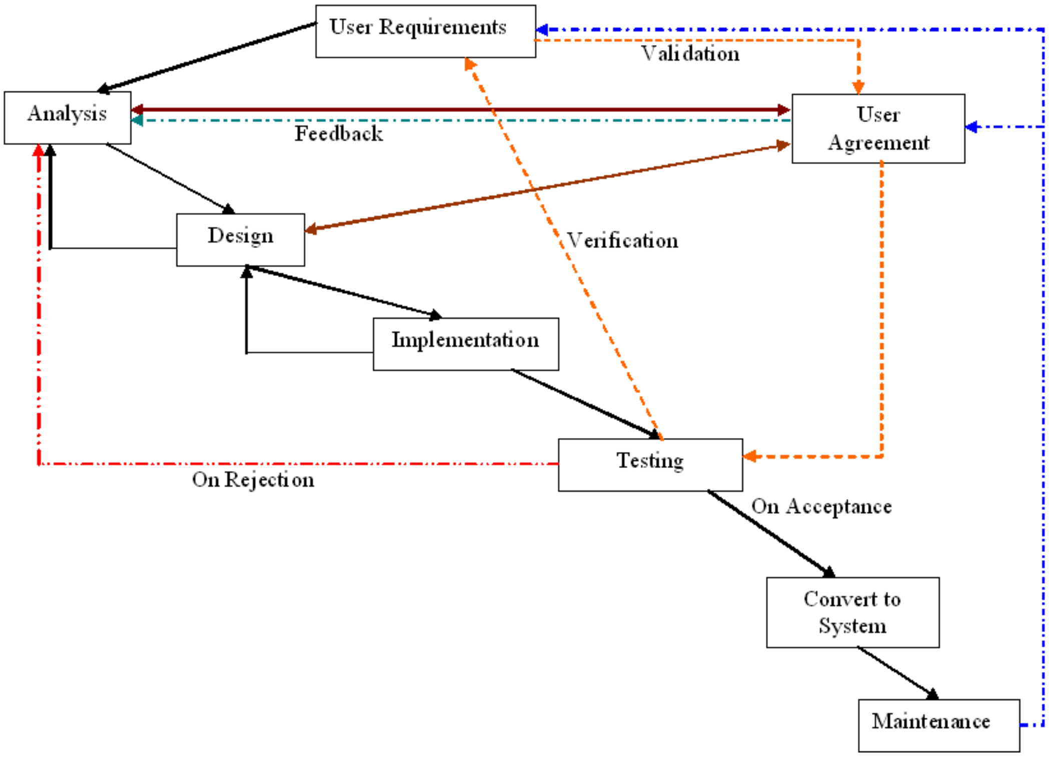

Figure 7. Educational software development life cycle (ESDLC).

Thus, the constructive and collaborative development methodologies should be followed while using SDLC in educational software development. On completion of analysis and design, at each level, users' satisfaction feedback should be collected. In testing phase, the users' requirements need to be verified and validated. During maintenance phase, when the user finds some difficulty, the feedback should collected for analysis purpose and studied for rectifications.

Future Scope

This study discusses all the phases of software development life cycle implementation in developing educational software. Each phase may be studied intensively further individually. More constructive ideas shall be discussed and demonstrated with respect to different teaching and learning processes.

References

Adaptive educational hypermedia. (2010). Retrieved December 25, 2010, from http://en.wikipedia.org/wiki/Adaptive_educational_hypermedia

Afari-Kumah, E., & Hannah, A. T. (2009). Tertiary students' view on information and communications technology usage in Ghana. *Journal of Information Technology Impact, 9*(2), 81-90.

Alkhafaji, S., & Sriram, B. (2010). Educational hypermedia development methodologies. Presented in First Joint Scientific Symposium Committee, CAS, Sultanate of Oman on April 13, 2010 in Sohar University, Sultanate of Oman, Conference Proceedings (pp. 8-24).

Braude, E. J. (2010). *Software engineering—An object oriented perspective.* Wiley.

Dix, A., Finlay, J., Abowed, G. D., & Russell, B. (2008). *Human computer interaction* (3rd ed.). Pearson Education.

Educational Technology Innovation and Impact/Why use Technology in Education/Tactile-Kinesthetic Learners. (2010). Retrieved November 15, 2010, from http://en.wikibooks.org/wiki/Educational_Technology_Innovation_and_Impact/Why_use_Technology_in_Education/Tactile-Kinesthetic_Learners

Hoffer, J. A., Joey, F. G., & Valacich, J. S. (2009). *Modern systems analysis and design* (5th ed.). Pearson Education.

Jawadekar, W. S. (2010). *Software engineering.* McGraw Hill.

Marsic, I. (2005). Software engineering. Rutgers University.

Matrai, R. (2010). *User interface.* Intech.

O'Neill, K. D., Edelson, D. C., Louis, M. G., & D'Amico, L. (1995). Learning to weave collaborative hypermedia into classroom practice. CSCL Proceedings 1995 in Bloomington, I.N..

Randalph, J. J. (2008). *Multidisciplinary methods in educational technology research and development*. HAMK University of Applied Sciences.

Riley, D. (2007). Educational technology and practice: Types and timescales of change. *Educational Technology and Society*, *10*(1), 85-93.

Software development process. (2011). Retrieved March 2, 2011, from http://www.ask.com/wiki/Software_development_process

Sommerville. (2009). *Software engineering* (8th ed.). Pearson Education.

Sue, C. (2008). *The new software engineering*. Global Text.

Technology impacts on education. (2010). *BBC News*. Retrieved November 15, 2010, from www.news.bbc.uk/2/hi/technology/8508040.stm