

# A Proposal for a Lifecycle Process for Hybrid Learning Programs

Won Kim

Sungkyunkwan University  
Suwon, S. Korea  
wonkim@skku.edu

**Abstract.** A hybrid learning program is a mixture of traditional in-class learning components and e-learning components. A learning program may be a single course or an entire curriculum. A learning program and courseware must be designed to help learners learn. As such, a hybrid learning program is a rather complex artifact. Commercial software is in general a very complex artifact that has a clearly defined lifecycle for planning, development, deployment, maintenance & upgrade, and termination. Although learning programs and courseware are not nearly as complex as commercial software, adapting the lifecycle process for the creation and managing of commercial software to hybrid learning programs and courseware should make it possible for the learning program managers and courseware developers to take a holistic and disciplined approach to the creation and management of hybrid learning programs and courseware. This paper proposes a lifecycle process for hybrid learning programs and e-learning courseware.

**Keywords:** hybrid learning, blended learning, lifecycle process, software development process, courseware development

## 1. Introduction

[1] provides rigorous definitions of hybrid learning and hybrid learning programs, and shed light on the many possible types of hybrid learning. Further, it outlines a methodology for creating and managing hybrid learning programs. One common aspect of every type of hybrid learning is that it is a mixture of traditional instructor-led in-class learning and e-learning which does not require the instructor and students to be concurrently present in the physical classroom. E-learning programs are in general more complex to create and manage than the traditional in-class learning programs, since e-learning programs must take into account not only the capabilities and limitations of various technologies that can be used in creating and delivering the learning contents, but also the fact that in general instructors do not lead the learning in real-time. Since hybrid learning programs combine e-learning and traditional in-class learning, creating and managing hybrid learning programs is inherently more

complex than either the traditional in-class learning programs or e-learning programs alone. If the learning program is not just for a single course, but for a curriculum, it obviously becomes even more complex.

There is considerable similarity between the process of creating and managing e-learning or hybrid learning programs and that of commercial software. Intuitively, they are both complex and therefore require upfront planning. After planning, they are developed. After development, they are deployed and assessed based on feedback from the customers (users and learners). They are then maintained and upgraded. Once their usefulness or value dissipates, they are retired. To be sure, there are some important differences between creating and managing learning programs and commercial software. In general, learning programs are not nearly as complex as software, with respect to the logic and logic branches. For learning programs, “learnability” is the most important objective, while it is in general not nearly as important for commercial software. Despite such differences, adopting the lifecycle process for creating and managing commercial software should help in creating and managing both e-learning and hybrid learning programs. It will allow the managers and developers of the learning programs to take a holistic and disciplined approach to creating and managing the learning programs, including the setting of the objectives, assessing the achievement of the objectives, learning program (and courseware) development (schedule, budget and resources), quality of the learning program (and courseware), usability of the courseware, learnability of the program, learning asset management, etc.

In this paper, I will propose a lifecycle process for creating and managing hybrid learning programs. The process is a sequence of steps. I will discuss considerations in each step in some detail. A major subset of the process is obviously applicable to creating and managing e-learning program and courseware, a subset of a hybrid learning program and courseware.

## **2. Lifecycle Process**

The lifecycle of commercial software consists of three primary phases: planning, development and deployment, and maintenance and upgrade. The development phase in turn consists of the upstream phase and the downstream phase [2]. The upstream phase includes three key steps: requirements specification, design specification, and test planning. The downstream phase also includes three key steps: implementation, testing, and release. The design specification includes both basic (or architecture) design, and detailed design. Often the basic design step belongs to the upstream phase, and the detailed design step is pushed to the downstream phase. Further, detailed design, implementation, and testing are often done concurrently.

During the planning and development phases, documents are produced and they are reviewed by the stakeholders (i.e., people who need to know their contents). There are two types of document: one is for internal use, and another is for release to the customers. The internal-use documents include a plan document, a requirements specification, a basic design specification, a detailed design specification, a test plan, source code with block comments. The “for-customers” documents include

executable code, release notes, user manuals and references. During the maintenance and upgrade phase, most of these documents undergo changes. Further, during this phase, a defect database is maintained to keep track of defects reported and resolved.

**Table 1.** First-level Comparison of the Lifecycle Process for Commercial Software and Hybrid Learning Programs

	<b>Commercial Software</b>	<b>Hybrid Learning Programs</b>
<i>Planning</i>	Product planning (plan document) & review	Program planning (plan document) & review
<i>Upstream development</i>	Requirements gathering & analysis (requirements spec) & review	Requirements gathering & analysis (requirements spec) & review
	Basic design (basic design spec) & review	Basic design (basic design spec) & review
	Test planning (test plan) & review	Test planning (test plan) & review
<i>Downstream development</i>	Detailed design (detailed design spec) & review	Detailed design & implementation (courseware, courseware notes) & review
	Implementation (source code, source code block comments) & code review	
	Testing: unit testing, integration testing, system testing, acceptance testing (test suite, test script) & test suite review, test results review	Testing: unit testing, integration testing, courseware testing & test results review
	Release (executable code, release notes, user manuals & references) & review	Release (courseware, program guide, learners guide to e-courseware)
<i>Deployment &amp; assessment</i>	Deployment & customer training	Deployment & customer training
	Customer support & assessment	Customer support & assessment
<i>Maintenance &amp; upgrade</i>	Maintenance & maintenance release	Maintenance & maintenance release
	Upgrade & upgrade release	Upgrade & upgrade release
<i>Retirement</i>	Retire the product	Retire the program

Table 1 provides the highest-level outline of the lifecycle processes for commercial software and hybrid learning programs. At this level, the two processes seem almost

the same, although the details are considerably different, as I will show in the remainder of this paper. Each process consists of several phases, and each phase in turn consists of one or more steps. In Table 1, the items in parentheses after each step are the deliverables (output) of the step. Each of the deliverables needs to go through a review process by the stakeholders.

Note that in Table 1, the basic design is placed in the upstream development phase, while the detailed design is in the downstream phase. Most commercial software development is done under very compressed schedule, and implementation and detailed design are done concurrently, along with unit testing and integration testing. The implementation and detailed design of e-learning courseware can also proceed concurrently.

There are 7 steps before the release step. These are planning, requirements analysis, basic design, test planning, implementation, detailed design, and testing. I note that these 7 steps in general need to be iterated at least a few times, to allow for changes and improvements to “earlier” steps as shortcomings in them are discovered while the “later” steps expand and flesh out the deliverables of the “earlier” steps. For example, during implementation, inadequacy in basic design may be discovered, forcing changes to the basic design, and possibly the test plan as well. During implementation, problems with the development schedule may be discovered, requiring changes in the product development plan or the requirements specification, which in turn may force changes in the basic design.

Further, all of the deliverables of the development steps before the deployment step need to be synchronized shortly before or shortly after the release step. In other words, all of the documents (the plan document, requirements specification, basic design specification, detailed design specification, test plan, user manuals and references) and product assets (source code and source code block comments, test suite, test script) produced must be made consistent among themselves. For example, all the features included in the requirements specification should be reflected in the design specification, source code, user manuals and references; while those features not in the design specification, source code, user manuals and references should not be in the requirements specification.

### **3. Lifecycle Process for Hybrid Learning Programs**

Despite the similarity at the highest level between the lifecycle processes for commercial software and hybrid learning programs, the details of most of the steps the lifecycle processes are very different. In this section, I will describe in some detail each step of the lifecycle process for the creation and managing of hybrid learning programs. Although documents and/or program assets are generated in each step of the process, and they are to be reviewed before proceeding to the subsequent step, for expository convenience, I will discuss them in a separate subsection at the end of each phase.

### **3.1 Program & Courseware Planning Phase**

The learning program manager, or someone he designates, leads the planning efforts. The leader forms a planning task force to plan a learning program, with some people assigned on a full-time basis and others on a part-time basis as needed. Whom he should bring into the task force depends on the nature of the learning program, the structure and business model of the organization that is to create and manage the learning program. In general, a program planner, some courseware designers and developers, some instructors, a technical support leader, a quality assurance leader, and a program assessor should be included.

The planning phase for hybrid learning programs needs to define the program, set objectives and strategy; stipulate the schedule, resources, budget, and finance; and program assessment. I will explain these below.

1. The definition of the program includes a description of what the target learners will learn (including a list of the major subjects that program will cover), duration of the program, purpose of the program (i.e., certification, academic credit, self-satisfaction), and the profile of the target learners. The profile of the learners includes the knowledge prerequisites for taking the program.
2. The objective of the program may be a combination of increased learning effectiveness for the learners; and any combination of revenue generation, cost savings, space (classroom) savings, reduced congestion on or near the campus, relief in parking space, etc. for the organizations that create and/or manage the program [1]. The strategy is the strategic means of achieving the objectives.
3. The schedule includes the dates for all key milestones in the planning, development, deployment, and assessing the learning program; and securing the resources and budget.
4. The resources include people, computers and equipment, software tools, etc. that will be available for creating, delivering, managing and supporting the courseware; administering, managing and assessing the learning program; and marketing and selling the programs. People resources include program planners, courseware designers and developers, instructors, tutors, teaching assistants, program administrators, system administrators, program assessors, marketing and sales people, etc.
5. The budget is to pay for all the resources, and other expenses.
6. Finance includes the means of providing the budget, and also revenue projection for the learning program.
7. The assessment is to assess how well the program's objectives will have been met, and should specify the means to be used.

#### **Documentation and review**

Before the planning phase concludes, a plan document needs to be written, which will serve as a guide for the development phase. The document should be reviewed by all members of the planning task force, and the key people who will be involved in the upstream development phase.

### **3.2 Upstream Development Phase**

This phase includes 3 steps: requirements gathering & analysis, basic design, and test planning. The requirements gathering & analysis step defines, based on the plan document generated in the planning phase, “what to” develop, and what the constraints are. The basic design step defines “how to” develop, largely, the e-learning courseware, and, to some extent, the overall hybrid learning program. The test planning step specifies how the e-learning courseware, and the overall hybrid learning program, will be tested before deployment.

The learning program manager, or someone he designates, leads the requirements gathering & analysis and basic design efforts. The leader forms a requirements analysis and basic design task force, with some people assigned on a full-time basis and others on a part-time basis as needed. Whom he should bring into the task force depends on the nature of the learning program, the structure and business model of the organization that is to create and manage the learning program. In general, a learning program planner, some courseware designers and developers, some instructors, a technical support leader, a quality assurance leader, and a learning program assessor should be included.

The quality assurance leader, or someone the learning program manager designates, forms a test plan task force, with some people assigned on a full-time basis and others on a part-time basis as needed. In general, a learning program planner, some courseware designers and developers, some instructors, a technical support leader, and a learning program assessor should be included.

In the remainder of this section, I will discuss each of the 3 steps in some detail.

#### **Requirements gathering & analysis step**

This is the most important step, as it serves as the basis for all subsequent steps, especially the basic design and test planning steps. This step needs to bring out, and document, all the requirements associated with the contents of the courseware; and the development, deployment, and management of the learning program and the courseware. The first set of the requirements to analyze, above all else, is the plan document created in the learning program planning phase, particularly the definition, objectives and strategy of the program; resources, and schedule. The requirements fall into at least 8 categories. These include the learning program architecture, learning program contents, learnability, e-learning usability, e-learning look & feel, e-learning management system architecture quality attributes, e-learning tools and environment, and learning program assessment.

#### *Learning program architecture*

As a hybrid learning program is a mixture of traditional in-class learning and e-learning, the “contents” will be some sequences of the two types of learning, that is, the use of in-class learning for some topics, and the use of e-learning for other topics. This sequence, or the learning program architecture, needs to be specified.

#### *Learning program contents*

All major topics the learning program is to cover, along with the scope of coverage for each topic, the level of difficulty for each, and the duration for each major component of the program should be specified, guided by the definition of the program in the plan document.

Exercises and exams in general need to be included in a learning program and courseware. Exercises help the learner to absorb the concepts effectively, and exams give the learners an opportunity to focus on learning and to put various concepts together into a coherent bigger picture. There are various types of questions for the exercises and exams, including true-false questions, multiple choice questions, filling in blanks, subjective questions, individual projects, and team projects. Appropriate types of questions should be selected that will cover the full range of topics covered in courseware or learning program.

Learning program contents can often be augmented by such learning aids as the table of contents, summarization, (in the case of e-learning courseware) links to related Web pages, etc. A guideline for including such learning aids may be set forth as well.

Further, new learning program contents may be created by reusing parts of existing learning programs. Reusable parts of existing learning programs should be specified for each topic of the new learning program. Some of the contents may be reusable with no changes at all. Some of the contents may be reusable with some changes, while others may only be usable as base references.

#### *Learnability*

The biggest problem with many of the e-learning programs has been the result of the courseware developers' focusing on wrapping the contents with technology, and losing focus on helping learners learn [3]. There are at least 4 types of techniques that can be used to help learners learn. Some or all of these may specified as requirements for the courseware developers.

1. The courseware content developers must invest the time necessary to properly organize the contents, present concepts progressively from small and simple to large and complex, illustrate concepts with examples, summarize concepts with figures and tables, etc. – all the standard techniques people use (or should use) in technical writing and preparing lectures and seminars, etc.
2. Various learning-enhancement techniques, not based on the use of technologies, may be applied, including the use of interactivity to engage the learners [4], themes (relevant to the topics of learning) [5], problem-solving challenges found in online games [6], etc.
3. Technologies may be used to engage the learners and/or to convey concepts better. Relevant technologies include audio (voice, music, sound) [7], motion graphics, animation [8], 3-d images [9], podcasts [10], video [11], simulations [12], etc..
4. Additional relevant results from learning theory may be applied.

#### *E-learning usability*

The “usability” here refers only to the usability for the learners, not for the courseware developers or designers or the learning program support staff. The usability of a hybrid learning program refers only to the e-learning component. The e-

learning component includes not only the courseware, but also the learning management system, if the learners are to use one. An e-learning Web site combines the courseware with a learning management system.

The “ease of use” of the learning management system is about how easy it is for the learners to operate it [13]. It includes a “non-technical” aspect such as clear and intuitive operating instructions for non-technical learners, mechanisms for accident prevention (e.g., unintended deletion of the answers to an exam), etc. A technical aspect includes facilities to register for a program, take exams and submit answers, check grades, navigate the contents forward and backward, keyword-based search for topics, scroll a Web page, post a note or annotation to courseware, communicate with the instructor or classmates, etc.

The usability of e-learning courseware has two aspects. One is accessibility, and another is “extensional” quality. Accessibility becomes problematic if the learner cannot access the courseware in his learning environment. There may be many reasons, including software and software version mismatch, the use of a firewall, capacity of memory and hard disk drive on a computer, Internet access speed, etc. [3] Extensional quality includes avoidance of undefined terms and acronyms, typographical errors, inconsistent presentation style and look & feel.

#### *E-learning look & feel*

The “look & feel” of a hybrid learning program refers to the e-learning component, that is, both the e-learning management system and the e-learning courseware. The look & feel, along with the usability, should also be a key criterion for selecting a learning management system.

The look & feel of e-learning courseware is concerned with the style, including the layout, background design; font style, font size; colors for the letters and drawings, background colors; the use of upper case and lower case letters; the positioning of visual aids and their captions; types and design of icons; etc. The courseware content developers must invest the time necessary to follow widely accepted industry practices, where innovation and differentiation are not needed, use them consistently throughout the courseware, and introduce some deviations, where innovation and differentiation will help. It would be best to provide a style template for courseware developers to use.

#### *E-learning management system architecture quality attributes*

The requirements specification for commercial software has a long list of architecture quality attributes. They include performance, scalability, reliability, modifiability, extensibility, portability, testability, security, availability, etc. All of these are applicable to the e-learning management system, including e-learning Web sites. However, only those whose consequences are visible to the learners need to be included in the requirements specification. The following architecture quality attributes, and the goals for each, should be included:

- Performance refers to the response time.
- Scalability refers to performance as the number of learners increases who concurrently access the courseware, and as the volume of course contents or the number of Web pages increases.



- Reliability refers to the ability of the learning management system to recover from system crashes to provide integrity of stored data. For example, the learners' registration data, exam answers, grades, etc. should never be lost or corrupted.
- Portability refers to the porting of the learning management system to different hardware and software operating environment, including different multimedia formats & media players, different plug in technology, etc.
- Security refers to the safeguarding of the learners' private data from unauthorized access or unintended deletions or changes.
- Availability refers to the e-learning courseware and the learning management system being available all the time or at least most of the time.

Other architecture quality attributes, such as modifiability and testability, are meaningful only to the developers of learning management system itself, and are not relevant to the requirements specification.

I further note that such architecture quality attributes as performance, scalability, reliability and availability are met by the facilities of the database systems used by the learning management systems, and, as such, they depend on a proper selection and administration of the database systems (often by the data centers that host the e-learning management system).

#### *E-learning tools & environment*

The tools & environment refers to the e-learning component of hybrid learning programs. The requirements for the following 3 types of tools and environment should be specified:

- for designing and developing e-learning courseware.
- for managing e-learning courseware and the learning program.
- for the learners to access the e-learning courseware and learning management system.

#### *Learning program assessment*

The assessment of a hybrid learning program is to quantitatively establish the extent to which the objectives of the program have been met. Key requirements related to assessment need to be specified, including a methodology to use, the scheduling of assessment, resources and budget.

- The methodology may test a focus group that did not take the learning program, and compare the results against those of tests given to the learners who took the learning program; or compare the results of tests given to the learners before they took the learning program and after. The choice of a methodology depends to a large extent on the nature of the learning program.
- The scheduling of the assessment includes when the assessment is done, and how many times it is done.
- The resources for the assessment include people resources to design and develop the assessment, the equipment and tools needed to conduct the assessment, and the facilities needed for the assessment.
- The budget for the assessment is the cost of paying for all the resources and other related expenses.

**Basic design step**

I note that for developing an e-learning Web site, rather than just e-learning course contents, the basic design needs to specify all key aspects of designing a Web site. It is beyond the scope of this paper. For developing e-learning course contents, the basic design necessary is fairly limited, as shown below. Only some of the elements specified or defined in the requirements gathering & analysis step need to be elaborated into guides for the courseware developers.

*Learning program architecture*

N/A

*Learning program contents*

Key subtopics of each of the major topics in the learning program architecture specified in the requirements specification should be listed and described. Guidelines for the number and types exercises and exams should be provided for each subtopic. Further, if parts of an existing learning program contents are to be reused with changes, brief descriptions of the extent of the changes need to be provided.

*Learnability*

The use of technologies should be specified for each topic.

*E-learning usability*

N/A

*E-learning look & feel*

If a style template is not available, create one for use by the e-learning courseware developers.

*E-learning management system architecture quality attributes*

N/A

*E-learning tools & environment*

N/A

*Learning program assessment*

A basic outline of the assessment should be specified, including the types of questions to ask or tests to be given.

**Test planning step**

Here “test” refers to the quality assurance of the learning program and the e-learning course contents, not the examinations that the learners will take. Test planning includes people resources and budget, testing schedule, and testing tools and environment, similar to the assessment planning and the overall program planning.

Test planning also includes checking the quality of the learning program and the learning contents. The quality check is to be done by the courseware developers and quality assurance people, and cover the following aspects of the courseware and learning management system:

- Intensional quality: includes inadequate materials, superfluous materials, inadequate visual aids, superfluous visual aids, etc.
- Learnability of e-learning courseware: includes consistent use of a theme; proper use of game-like challenges, multimedia data, simulations; adequate exercises, exams; etc.
- Errors: includes typos, conceptual errors, incorrect answers to exercises and exam questions, etc.
- Content flow: the smoothness of transition between separate component of the courseware, and the extent of overlap among the components (A little overlap is good and even desirable sometimes, but much overlap is not).
- Extensional quality: includes undefined terms, undefined acronyms, etc.
- Usability of e-learning courseware: includes consideration of the learner's environment, intuitive operating instructions, error prevention, etc.
- Look & feel: includes the layout, font style, font size, colors, icons, etc.
- Style: includes overall consistency of presentation style and look & feel.

Commercial software vendors often make a beta release of their software to a limited number of customers ahead of a general release. This is to receive feedback on the quality and features of the software. A hybrid learning program and/or e-learning courseware may similarly be beta-tested before general release. The actual learners are often better judges of the quality, learnability and usability of learning programs and e-learning courseware than the courseware developers and the quality assurance staff. If beta testing is to be included, it should be included in the overall program development and release schedule, and the beta-testing learner group should be identified and specified in the test plan document.

#### **Documentation and review**

Before the upstream development phase concludes, a requirements specification, a basic design specification, and a test plan need to be written, which serve as guides for the downstream development phase. The documents should be reviewed by all members of the task force, and the key people who will be involved in the downstream development phase.

### **3.3 Downstream Development Phase**

The downstream development phase basically includes the implementation step and testing step. As with commercial software, implementation and testing should proceed concurrently.

#### **Detailed design & implementation step**

In the case of e-learning courseware development, there is little distinction between detailed design and implementation, while the distinction is clear in the case of e-

learning Web site development or e-learning management system. However, considerations of the development of an e-learning Web site or e-learning management system is beyond the scope of this paper.

The learning program manager needs to plan the courseware development efforts, and subsequently work with the courseware development leaders to ensure that the development efforts proceed on schedule.

### **Testing step**

The testing of commercial software is best done by dividing the testing work between the developers and the quality assurance staff [2]. The developers test the software roughly in 3 granules: unit testing (of single functions), integration testing (of modules, or subsystem, of multiple functions), and full system testing. Further, they inspect their own code, and review other developers' code periodically. The quality assurance staff performs performance and stress testing on the full system, and the release staff performs acceptance testing of the full system.

The same testing discipline may be applied to the testing of e-learning courseware (and e-learning management system).

The testing granules for e-learning courseware would include a unit (a subsection), a module (a section), and full courseware. The e-learning courseware developers should perform quality assurance at each granule as the development efforts proceed. The quality assurance staff should put on the learners' "hat" and test the full courseware and full learning program.

The testing must cover every quality aspect of the courseware outlined in the test planning step.

### **Release step**

After the e-learning courseware and the full learning program have been tested, either the learning program "release" staff, or the learning program manager, should approve the release of the program and e-learning courseware.

When commercial software is released, often the release staff write "release notes" that explain how to get around some serious defects that have not been fixed. If circumstances require the release of e-learning courseware or learning program with some defects that have not been fixed, the release staff, or their equivalent, should write release notes.

If the release is a beta release, the beta testing learners should be managed, and the results should be incorporated in the general release. In effect, a beta release may be viewed as a "real" release, and the learning program and courseware may be considered to have entered into the maintenance & upgrade phase for the "general" release.

### **Documentation and review**

The deliverables of the downstream development phase include the full e-learning courseware and full learning program, a guide to the learning program (as an overview to the learners), release notes, and learners guide for operating the courseware or learning management system.

### **3.4 Deployment & Assessment Phase**

Once the learning program and e-learning courseware have been released, feedback from the learners flows in through various channels, including comments on the e-learning Web site, emails, instructors, tutors, learning support staff, etc. The feedback includes compliments, justifiable misunderstanding on the part of the learners, suggestions for improving the learning program and courseware contents, complaints about usability and look & feel, etc. Some of the compliments from satisfied learners can be used as testimonials in promoting the learning program. Some of the feedback or trends seen in the feedback can serve as a part of the requirements for subsequent changes or upgrades to the learning program and courseware.

The feedback should be segmented and saved in separate databases, including a “feedback database,” a “compliments database,” a “defect database,” and “requirements database.” Such databases are valuable learning program assets.

### **3.5 Maintenance & Upgrade Phase**

It is best to plan the maintenance and upgrade of a learning program and courseware. The plan includes the people resources and budget, and the management of a defect database and a requirements database. The resources are the people who will make the changes to the learning program and courseware, record learner feedback to the defect database and requirements database. The budget is to pay for the people, tools and equipment, and related expenses. The management of a defect database and a requirements database includes the selection of a database system and application for recording and managing defect reports and change requirements from the learners.

### **3.6 Termination Phase**

Commercial software is terminated when it becomes better to terminate it than to continue support the customer base that uses it. Similarly, a learning program and courseware need to be terminated if it makes no business sense to further support the learner base that uses them. The termination decision can have financial and legal consequences, and the organizations that provides the program and courseware must weigh them carefully and plan to address such consequences before taking the termination action.

## **4. Concluding Remarks**

I proposed a lifecycle process for planning, developing, and managing hybrid learning programs and courseware. The process was adapted from one developed for commercial software. Because the learning programs and courseware are in general far less complex than commercial software, the process is simpler. However, because of the nature of the learning programs and courseware, each of the steps within the lifecycle process involves very different issues to consider. The process may serve as a

guide to help learning program managers and courseware developers take a holistic and disciplined approach to the creation and management of the learning programs and courseware.

## Acknowledgments

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2008-(C1090-0801-0046))

## References

1. Won Kim, "Towards a Definition and Methodology for Blended Learning," in Proc. of the First International Workshop on Blended Learning, August 2008, Edinburgh.
2. Won Kim, "On Software Quality Assurance and Curbing Development Cost," Journal of Object Technology, July 2006
3. Won Kim, "Directions for Web-Based Learning", Proc. 5<sup>th</sup> Intl. Conf. on Web-Based Learning (ICWL-06), July 2006, Penang, Malaysia
4. Jim Elsenheimer, "E-Learning 1.0 Terms of Engagement: Keeping Learners Online," <http://www.learningcircuits.org/2003/feb2003/elearn.html>
5. Debbie Vogel, "E-Learning 1.0 Themes Add Creative Spark to Online Classes," <http://www.learningcircuits.org/2002/sep2002/elearn.html>
6. Davis Klaila, "Game-Based E-Learning Gets Real," <http://www.learningcircuits.org/2001/jan2001/klaila.html>
7. Lenn Millbower, "The Auditory Advantage," <http://www.learningcircuits.org/NR/exeres/6AF8D013-30DC-4CBA-BA15-09DBFD9B0E68.htm>
8. Thomas Toth, "Animation – Just Enough, Never Too Much," <http://www.learningcircuits.org/2003/aug2003/toth.html>
9. Eva Kaplan-Leiserson, "Trend: 3D Training," [http://www.learningcircuits.org/2005/nov2005/0511\\_trends.htm](http://www.learningcircuits.org/2005/nov2005/0511_trends.htm)
10. Eva Kaplan-Leiserson, "Trend: Podcasting in Academic and Corporate Learning," [http://www.learningcircuits.org/2005/jun2005/0506\\_trends.htm](http://www.learningcircuits.org/2005/jun2005/0506_trends.htm)
11. Ed Mayberry, "New Territory: Adding Video to Online Learning Offerings," <http://www.learningcircuits.org/2005/jul2005/mayberry.htm>
12. Jackie Dobrovoly, "Effective – and Ineffective – Instructional Strategies," <http://www.learningcircuits.org/2004/jan2004/dobrovoly.htm>
13. Dave Smulders, "E-Learning 1.0 Web Course Usability," <http://www.learningcircuits.org/2001/aug2001/elearn.html>