Synchronizing Development Life-Cycles: Rationale and Integrative Approach to Construction of eLearning Systems and Online Courses at a Tier 1 Research Institution

Samuel S. Conn
ssconn@vt.edu

Peter Macedo
pmacedo@vt.edu

Jackie Meese
jmeese@vt.edu
Institute for Distance and Distributed Learning
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061, USA

Abstract

Universities involved in distance learning face challenges in information and communication technology systems and online course development such as limited resources, time constraints, problems and errors associated with ad-hoc development, sustainable and useful eLearning systems, acceptable levels of user satisfaction, and support and maintenance issues. Through a qualitative case study of eLearning systems and online course construction initiatives at a tier 1 research university, the authors examine a strategy to synchronize life-cycle development methodologies. In this study the authors provide a case study of structured development life-cycle methodology for eLearning system development and online course development where each phase of the methodologies are synchronized and work in concert to optimize life-cycle development.

Keywords: development life-cycle, eLearning, methodology, course, system, information, communication

1. INTRODUCTION

Virginia Polytechnic Institute and State University (Virginia Tech) is home to the Institute for Distance and Distributed Learning (IDDL). Since 1999, IDDL has provided leadership, coordination, management, and support to the university's distance and distributed activities, including online learning and interactive video conferencing. In so doing, the need to optimize resources and minimize constraints requires a methodology for eLearning course development and eLearning system development that work in concert to support university distance and distributed learning. According to Royce (1972), a structured development life-cycle model can be used as the accommodating solution for initiatives involving multiple, complex requirements, such as eLearning course and system development. Instructional designers have long adapted life-cycle development approaches (e.g., the ADDIE model). As eLearning course development continues to depend on underlying information and communication technology (ICT) systems, the need to synchronize course development with systems development to achieve timely and efficacious outcomes increases.

The benefits of following a structured life-cycle methodology for system development include: (a) lack of problems and reduced errors associated with ad-hoc system development, (b) construction of sustainable and useful eLearning and information systems, (c) systems that perform within the IDDL enterprise architecture, (d) higher levels of user satisfaction, and (e) fewer support and maintenance issues. Similarly, benefits of following a structured life-cycle methodology for eLearning course development: (a) lack of problems and reduced errors associated with ad-hoc or haphazard instructional design and development, (b) construction of sustainable and efficacious eLearning systems and courses, (c) online courses that meet specified quality standards, (d) higher levels of learning and eLearner satisfaction, (e) fewer support and maintenance issues, (f) clear understanding of project scope and resources needed, and (g) improved understanding of instructional design and online teaching among faculty. These benefits support and reflect the justification for the study.

In this study the authors apply a qualitative research approach using a case study methodology of the construction and delivery of an undergraduate online course, supported by eLearning course and system development. The foci of the study include: (a) an examination of the life-cycle development methodology, (b) application of the methodology in eLearning course and system development, and (c) integration and synchronization of the methodology in eLearning course and system development.

2. LITERATURE REVIEW

The development life-cycle can be used as a methodology for application and system development projects (Marakas, 2006). Similarly, the life-cycle methodology can be effectively utilized to facilitate development in e-learning course and system initiatives such as those at Virginia Tech (Neville, Heavin, & Walsh, 2005). Use of structured methodology in eLearning course and system development has increased since linear, structured methodologies have been adopted by eLearning project managers (Horton & Horton, 2003). According to Horton (2003), eLearning course and system development must work in concert to achieve technology transparency, efficacious instruction, and an optimal learning experience for the online learner.

Satzinger, Jackson, and Burd (2004) and Horton and Horton (2003) concluded that the system life-cycle development (SDLC) methodology can function as a general framework for managing the eLearning course and system development process. When applying the SDLC methodology, project managers also rely on models, tools, and processes to build the eLearning system. In traditional approaches to system development, project managers utilize the SDLC as a framework for project management (Satzinger et al.). The SDLC methodology includes phases and activities required to develop a system (Whitten, Bentley, & Dittman, 2004). Dennis and Wixom (2003) indicate that the SDLC methodology involves comprehending how a system is used to support organizational needs. The methodology also includes designing, building, and delivering the system. Marakas (2006) conveys that the SDLC is a methodology that has evolved over the last few decades into a widely accepted structured approach to development.

According to Dennis and Wixom (2003), the SDLC is composed of four distinct phases: (a) system planning, (b) system analysis, (c) system design, and (d) system implementation. Avison and Fitzgerald (2003a; 2003b) identify the sequential stages as: (a) a feasibility study, (b) system investigation, (c) system analysis, (d) system design, (e) system development, (f) system implementation, and (g) system maintenance. Satzinger et al. (2004) identify five distinct SDLC phases: (a) system planning, (b) system analysis, (c) system design, (d) system implementation, and (e) system support. According to Marakas (2006), there are six distinct SDLC phases: (a) preliminary investigation, (b) system analysis, (c) logical design, (d) physical design, (e) system implementation, and (f) system maintenance. Stair and Reynolds (2006) describe five SDLC phases: (a) system investigation, (b) system analysis, (c) system design, (d) system implementation, and (e) system maintenance and review. Marakas, Satzinger et al., and Whitten et al. (2004) all describe a consistent view of the SDLC phases when logical and physical designs are grouped into a single design phase.

According to Whitten et al. (2004), the SDLC methodology is composed of the following five sequential phases: (a) Phase 1, the System Initiation Phase; (b) Phase 2, the System Analysis Phase; (c) Phase 3, the System Design Phase; (d) Phase 4, the System Implementation Phase; and (e) Phase 5, the System Support and Continuous Improvement phase. These five phases are examined and applied in this investigation.

Phase 1: System Initiation

eLearning course and system project managers should begin with the System Initiation Phase of the SDLC methodology to investigate and plan activities required to organize and start development (Whitten et al., 2004). Considerations in this phase include the project scope, goals, related schedules, and budget (Whitten et al.). Stair and Reynolds (2006) note that the first SDLC phase should be focused on investigation, the appropriate time to identify potential problems and opportunities. Investigation includes definition of the problem. According to Dennis and Wixom (2003), this phase includes: (a) determining why the course or system needs to be constructed, and (b) deciding how a project team should proceed to construct it. The value of the system to the organization is established at this point. A project plan and schedule are initiated in this phase as well as any feasibility studies that may be required (Dennis & Wixom). The project staff is identified and recruited, and the project operation is commenced.

Marakas (2006) comments that the system initiation phase begins with a feasibility study to determine whether or not to continue with the project. A primary deliverable for this SDLC phase is a formal project plan. According to Dennis and Wixom (2003), the project oversight committee reviews the system request and feasibility study and decides whether or not to proceed. During this phase, the project manager puts in place the tools and techniques to help manage subsequent SDLC phases (Dennis & Wixom).

Phase 2: System Analysis

The System Analysis Phase involves a detailed examination of the related and current processes, procedures, and systems of the organization (Marakas, 2006). Marakas notes it is important to document system requirements in this SDLC phase. Whitten et al. (2004) describes the process of analysis as factoring or decomposing a complex problem down into smaller, understandable components. Stair and Reynolds (2006) recommend that this phase include development of a list of requirements and priorities as a main deliverable. According to Satzinger et al. (2004), the activities associated with this phase include: (a) information gathering, (b) definition of system requirements, (c) prioritization of requirements, (d) prototyping, (e) creation and evaluation of alternatives, and (f) recommendations from system stakeholders. Dennis and Wixom (2003) note that the project team should define and document all system users, the function the system will perform, and when and where the system will be used.

During the System Analysis Phase, the project team attempts to gain in-depth comprehension of the problem and desired outcomes. System users are interviewed, and requirements and expectations are documented. Information gathering occurs through a variety of tasks (Valacich et al., 2004): (a) stakeholder input through surveys and interviews, (b) observation of the environment in which the system will be deployed, and (c) technical information regarding current infrastructure (Satzinger et al., 2004). Requirements information may be gathered through a variety of methods: (a) use of focus groups and interviews (Marakas, 2006), (b) questionnaires and observation techniques (Dennis & Wixom, 2003), and/or (c) reviews of existing documentation, observations of procedures and workflows, and investigations of commercial vendor solutions (Whitten et al., 2004).

Typically, the logical model is associated with system analysis, whereas the physical model is associated with system design (Satzinger et al., 2004). In logical models, system requirements are depicted without commitment to specific technology. Physical models are created to illustrate how the system will be implemented technologically. Requirements are prioritized to maximize resources and eliminate a phenomenon known as scope creep, which occurs when system stakeholders and users continue to add requirements and functionality to the system after the System Analysis Phase.

In the SDLC methodology, the need for requirements documentation in the System Analysis Phase is explicit. A requirement also can be described as a system feature or capability through which the purpose of the system is fulfilled (Sommerville, 2004; Valacich et al., 2004). A central approach common to all development methodologies is to analyze requirements to prioritize and order them for logical expression in a requirements document (Whitten et al., 2004).

Phase 3: System Design

Transition from the System Analysis Phase to the System Design Phase involves examining a solution in more technical and specific terms. According to Marakas (2006), while creating the logical model for the new system the project team must ensure problems in the old system are not recreated and that logical design meets all requirements identified in the System Analysis Phase. Models constructed in that phase are given greater detail as they are changed from high-level generic models to low-level implementation models. According to Satzinger et al. (2004), less user involvement is needed in this phase.

Architectural blueprints and metamodels are created to facilitate implementation, and integration issues are discussed (Whitten et al., 2004). The project team creates a blueprint for the system and identifies all system components in detail (Whitten et al.). According to Satzinger et al. (2004), project managers must describe, organize, and structure system components in a detailed architectural view. Technical solutions also are considered, and the best one is chosen (Whitten et al.).

In the architectural design of the system, the overall structure and framework of the solution are described. The project team considers how the system will operate with respect to software, hardware, and network infrastructure (Dennis & Wixom, 2003). Stair and Reynolds (2006) note that this phase should result in a technical design with specific details for inputs, outputs, hardware, software, database, telecommunications, personnel, and procedure components.

Phase 4: System Implementation

According to Satzinger et al. (2004), all activities that occur prior to any end-user interaction compose the System Implementation Phase of the SDLC. A large management and technical effort is required to create an operational system that meets the documented system requirements and physical design that is integrated with any existing system. Whitten et al. (2004) classify SDLC activities in this phase distinctly as construction, testing, installation, and delivery. New hardware and software are installed and tested; customization and integration work is performed (Whitten et al.). According to Stair and Reynolds (2006), related tasks include procuring various components in the system design and assembling them into an operational system. System documentation is finalized and appropriate end-user and system administration training is accomplished (Marakas, 2006; Satzinger et al., 2004).

Phase 5: System Support and Continuous Improvement

Kazman et al. (2003) define the System Support and Continuous Improvement Phase as the longest and most costly. During this time period, the system is placed in operation, maintained, and improved based on operational feedback (Whitten et al., 2004). Dennis and Wixom (2003) suggest the system support plan be implemented after system installation. According to Satzinger et al. (2004), this phase involves: (a) maintaining the system, (b) enhancing the system, and (c) supporting end-users of the system. Maintaining the system includes a range of regular activities that occur between installation and replacement or abandonment of the system. According to Stair and Reynolds (2006), the project manager should ensure successful and continued operation of the system. They suggest that a formal management review be conducted after each activity is successfully completed.

Ongoing system support is required for the useful lifetime of the system (Whitten et al., 2004). Users generally find problems that were not evident in prior SDLC phases after a few months of use (Marakas, 2006). System support activities include: (a) assisting users, (b) correcting defects, (c) making system recoveries and restorations, and (d) adapting the system to new requirements (Stair & Reynolds, 2006).

3. CASE STUDY

Virginia Tech's eLearning course enrollments now have exceeded the 100,000 mark. More than 3,500 distinct courses are offered to deliver instruction in 29 graduate degree, certificate, and licensure programs. Moreover, the university offers online 55 undergraduate core courses, and undergraduate concentration in the Humanities, and five non-credit professional development programs. As a result, the institution's need for resource optimization, scalable eLearning course development, and rapid construction of eLearning systems has continually increased. IDDL eLearning system development and online course development projects follow a structured methodology based on a sequential progression of seven phases: (a) Planning Phase; (b) Analysis Phase; (c) Design Phase; (d) Development Phase; (e) Testing Phase; (f) Implementation Phase; and (g) Evaluation, Support, and Maintenance Phase. Each phase is distinguished by activities, techniques, best practices and procedures that combine to construct viable, sustainable, efficient, and useful technology systems. System development moves from one distinct phase to the next after careful review and stakeholder consensus that all elements of the phase have been satisfactorily accomplished. The construct of the case study is to provide detail regarding the application of the methodology for eLearning system development followed by detail regarding the application of the methodology for eLearning course development.

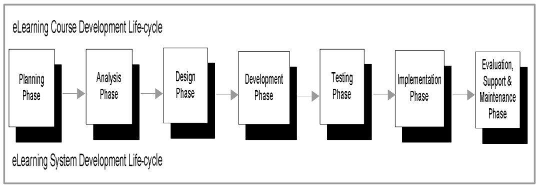
eLearning Course Development Life-cycle
Planning Phase
Analysis Phase
Design Phase
Development Phase
Testing Phase
Implementation Phase
Evaluation, Support & Maintenance Phase
eLearning System Development Life-cycle

Figure 1: Synchronized development life-cycle model

The case study system development methodology involves a life-cycle that begins with system inception and ends with system obsolescence. Figure 1 features the sequential phases of the IDDL life-cycle applied as a methodology to eLearning course and system development. Edges in the diagram represent the sequential flow from one completed phase to the next. A project team can elect to allow iterations to occur, particularly in the early phases. Iterating through previous development phases ensures errors are corrected as early in the life-cycle as possible. Final system requirements documentation is forwarded to readily construct system test plans, made available in the Testing Phase.

The case study involved construction and delivery of an undergraduate online course and followed a structured methodology based on a sequential progression through seven phases: (a) Planning Phase; (b) Analysis Phase; (c) Design Phase; (d) Development Phase; (e) Evaluation (Testing) Phase; (f) Implementation Phase; and (g) Evaluation, Support, and Maintenance Phase. Each phase was distinguished by activities, techniques, best practices, and procedures that combined to construct a viable, sustainable, efficient, and effective online course.

Planning Phase

eLearning System Development- In the case study, the Planning Phase began with a rationale and description of the business reasons for building the system. The need and value of the system was established and a detailed description of the system was created. Project sponsors were recruited and a steering committee was created to provide oversight. Resources available to the system development effort were identified and considered; constraints were identified and noted. System feasibility was studied from three perspectives: (a) technical feasibility, (b) economic feasibility, and (c) organizational feasibility. Technical feasibility described the extent to which the system could be successfully designed, developed, and implemented. A statement of technical feasibility answered the question: Can the system be constructed? Economic feasibility described the cost of the system versus the benefit(s) of the system. Total cost of ownership (TCO) was considered and established during the economic feasibility study. Organizational feasibility investigated how well the system would be accepted by the stakeholders and used as a viable system.

The Planning Phase also involved development of the project management plan. The project management plan included: (a) a GANTT chart to illustrate the development life-cycle schedule and work breakdown structure for development; (b) the critical path analysis of activities to determine the shortest time possible to complete the life-cycle development; (c) the general activities to occur in each phase of the development life-cycle; and (d) roles and staffing for the project.

eLearning Course Development- The Planning Phase began with a rationale and description of the online course. The need and value of the online course was established and a detailed description of the course was created. Project team members were identified and roles were assigned. Resources available to the instructional design and development effort were identified and considered; constraints were identified and noted. Course feasibility was examined from three perspectives: (a) technical feasibility; (b) economic feasibility; and (c) organizational feasibility. As noted, technical feasibility described the extent to which the course could be successfully designed, developed, and implemented. Economic feasibility described the cost of the course versus the benefit(s) of the course. Return on investment (ROI) was considered and established during the economic feasibility study. Organizational feasibility investigated how well the system would be accepted by the college, students, and faculty.

The project management plan for eLearning course development likewise included: (a) Statement of Work; (b) Work Breakdown Structure; (c) Project Timeline (GANTT chart); and (d) roles and staffing for the project. Documentation needs for the project were determined and assignments were made with respect to producing project documentation.

Analysis Phase

eLearning System Development- The Analysis Phase began with identification of system stakeholders. Stakeholder groups included: (a) owners of the system; (b) users of the system; (c) administrators and/or managers of the system; (d) technical or engineering supporters of the system; and (e) developers of the system. System analysis included requirements elicitation from one or more members from each stakeholder group. Two categories of requirements were documented: user and system. Within the two categories, the functional requirements were documented and the non-functional requirements were documented. Requirements were formally documented in a system requirements document. Additionally, system requirements were prioritized from most important to least important.

Functional requirements related directly to a process the system must perform or information it needed to contain. These requirements created the foundation for process models, data flows, use cases, etc. They defined the functions the system must have. Non-functional requirements referred to behavioral properties that the system must have (e.g., usability, security, reliability, availability, extensibility, etc.) These requirements also were used to design the user interface, hardware and software, and the system architecture. Review and analysis of existing system documentation (if applicable) was completed during the Analysis Phase.

The Analysis Phase involved development of the business process model or use case diagram to depict and communicate how users interact with the system, what the system must do, how the system functions. If applicable, other technical diagrams (e.g., data flow diagram) were constructed to communicate business process flows. A high level technical abstraction of the system topology was created to communicate, generally, how the system would work within IDDL's enterprise architecture. Final system requirements were forwarded to testers for completion of test plans.

eLearning Course Development- The Analysis Phase began with identification of: (a) the target audience and their characteristics including learner's existing knowledge and skills, (b) what types of learning constraints exist, (c) identification of course learning objectives/outcomes, (d) the online pedagogical considerations, (e) learning theory considerations, and (f) the learning environment. Course analysis also included pedagogical requirements elicitation from the teaching faculty. Gathering complete, accurate, and detailed course requirements was essential. Requirements were formally documented in a course requirements document. All stakeholders of the course agreed with the course requirements as stated in the course requirements document. Additionally, course requirements were categorized in terms of: (a) pedagogical requirements, (b) course materials, learning objects, and resources, and (c) technological requirements, (d) copyright requirements, (e) requirements based on relevant standards (SCORM, W3C Web Content Accessibility Standards, SACS accreditation, IMS, and Quality Matters), and (f) other as required.

Review and analysis of existing IDDL course documentation (if applicable) was completed during the Analysis Phase. New instructional design and development typically did not involve this activity. The eLearning course was reviewed and revised according to feedback from the project team members.

Design Phase

eLearning System Development- The design phase involved translating the functional and non-functional system requirements into logical models and blueprints for the system. The Business Process Model or Use Case Diagram was reviewed, refined, and approved. The goal in the design phase was to develop the implementation diagrams, metamodels, specifications, and performance criteria. In the design phase the eLearning system design was integrated with IDDL's enterprise architecture and current technology stack. The design phase also involved mapping data architectures, application programming, and user interface(s). At the end of the design phase, system engineers and technology providers met to approve the deliverables and design plan for the system.

eLearning Course Development- The Design Phase involved selecting instructional strategies, learning materials, and instructional media to achieve learning objectives and meet course requirements. The course design was reviewed, refined, and approved. The goal in the design phase was to bring course elements together to achieve learning objectives and support a well founded pedagogical flow. At the conclusion of the design phase, the project team met to approve the deliverables and design plan for the course. In this phase the project team also identified opportunities for efficient content re-use and development of learning objects.

Development Phase

eLearning System Development- In the development phase the system was constructed in a development environment. Production systems were isolated from the development environment so as not to experience any adverse effects, downtime, or operational issues. The development environment mirrored the production environment (i.e. configuration, technology stack, topology, etc.). Following the design plans, modifications to the enterprise data structure were made (if required), software programming occurred, customization of open source or existing software was completed, middleware components were created and instantiated, and user interfaces were completed. *After testing*, the components of the system were completed and integrated with the production IDDL network. IDDL system developers employed techniques such as prototyping to receive feedback from system users and other stakeholders in short, iterative cycles of development.

eLearning Course Development- In this phase, instructional designers and developers created and assembled the content and instructional materials that were defined in the design phase. Moreover, storyboards were created, content was developed and graphics were designed. Instructional designers worked closely with developers (programmers) to develop and/or integrate technologies, pedagogy, and course assets. The online course was reviewed and revised according to feedback from the project team members. In the development phase, instructional designers and developers employed techniques such as prototyping to receive feedback from course stakeholders in short, iterative cycles of development.

Testing Phase

eLearning System Development- The testing phase began with the development of test plans. Test plans were based on system requirements and created concurrently with design and development activities. Test plans for the system were made available by the end of the development phase. Test plans contained three elements: (a) the requirement to be tested, (b) how the requirement should be tested (procedure), and (c) the expected outcome of the test.

After system requirements were finalized in the Analysis Phase, all requirements documents were submitted for development of system test plans. Test plans were constructed to test usability, functional requirements and non-functional requirements at the unit and system level. Test plans were derived directly from the requirements documents generated in the Analysis Phase. IDDL system testers were not associated with the design and development phases and had little to no prior knowledge of the system.

Testing was completed prior to migration of the system from a development environment to a production environment and implementation of the system. All system tests were satisfactorily completed (passed) prior to beginning the implementation phase.

eLearning Course Development- The Evaluation Phase began with a review of a quality assessment tool (rubric) for online courses. Evaluation of the online course was conducted in concert with the rubric. Evaluation and testing of the course was based on: (a) course requirements as defined in the Analysis Phase, (b) a quality assessment rubric, and (c) functional technology testing of all hypermedia links and course technologies/tools. Test plans for the course were made available by the end of the development phase. Test plans contained three elements: (a) the requirement to be tested, (b) how the requirement should be tested (based on the rubric), and (c) the expected outcome of the test.

Evaluation and testing was completed prior to promotion of the course from development status to production status. Course evaluation and testing was satisfactorily completed (passed) prior to beginning the implementation phase.

Implementation Phase

eLearning System Development- The Implementation Phase began with discussion of the most appropriate strategy for system implementation. The appropriate strategy was based on a risk tolerance assessment, technology adoption strategy, and criticality of the system. The initial strategy for implementation was determined as either: direct cut-over, parallel implementation, or phased-in implementation. In the Implementation Phase the system “went live”. The system was migrated from development to production, systems users were informed and given access, users were oriented and trained on the system, system documentation was made available to users, and the system was supported.

eLearning Course Development- The Implementation Phase began when the online course was taught. Prior to implementation the course was scheduled and students registered for the course. During course implementation the course was monitored for issues with access, availability, and transparency of technology. Issues were corrected and faculty and students were supported by IDDL’s faculty and student support initiative. The project team solicited real-time feedback from the faculty and maintained a log for future iterations of the course development.

Support & Maintenance Phase

eLearning System Development- System documentation was maintained, the system was monitored, performance tuning occurred, and the system was operated and used in this phase. Data were collected regarding system performance and usability, and new features and functionality for the system were considered and documented for the Planning Phase in the next iteration of the system’s development.

eLearning Course Development- The course was monitored, ad-hoc quality adjustments were made, and the course was offered and taught in this phase. Data were collected regarding course performance and efficacy, and new features and functionality for the course were considered and documented for use in the Planning Phase of the next iteration of course development. A summative evaluation was conducted in this phase.

4. CONCLUSIONS

In this case study the authors examined use of the SDLC as a methodology for eLearning course and system development for an undergraduate online course. The value of synchronizing development and progressing through a life-cycle methodology in concert as eLearning courses and supporting eLearning systems are constructed was noted in the literature review and in the case study. Synchronized development yields: (a) an end result with fewer eLearning course and system interoperability issues, (b) technology transparency, (c) tighter integration of learning objects with course management systems, and (d) optimal use of development resources.

As demonstrated by the case study, eLearning system development and eLearning course development can benefit from a structured methodology and be synchronized to work in concert as both are advance through the life-cycle. The results of this study can be used by institutions of similar size and composition who are engaged in eLearning initiatives. Successful implementation of eLearning courses is dependent on the underlying eLearning technologies and systems (e.g., ICT). To overcome problems with interoperability, scale, integration, and usability, utilization of a synchronized development life-cycle methodology is the accommodating solution.

5. REFERENCES

Avison, D. and G. Fitzgerald (2003a). *Information systems development: Methodologies, techniques and tools* (3rd ed.). Maidenhead, England: McGraw-Hill.

Avison, D. and G. Fitzgerald (2003b). Where now for development methodologies? *Communications of the ACM, 46*(1), 79-82.

Dennis, A. and B. H. Wixom (2003). *Systems analysis and design* (2nd ed.). New York: John Wiley.

Horton, W. and K. Horton (2003). *E-learning tools and technologies*. New York: John Wiley.

Horton, W. (2006). *E-learning by design.* New York: Pfeiffer.

Marakas, G. (2006). *System analysis and design: An active approach* (2nd ed.). New York: McGraw-Hill Irwin.

Neville, K., C. Heavin and E. Walsh (2005). A case in customizing e-learning. *Journal of Information Technology*. *20*, 117-129.

Royce, W. (1970). Managing the development of large software systems: Concepts and techniques. In *Proceedings of the Western Electric Show and Convention (WESCon) 1970*. (pp. 328-338). Los Angeles: IEEE Conference Publication Services.

Satzinger, J., R. Jackson and S. Burd (2004). *Systems analysis and design in a changing world* (3rd ed.). Boston: Course Technology.

Sommerville, I. (2004). *Software engineering* (7th ed.). Boston: Addison Wesley.

Stair, R. and G. Reynolds (2006). *Principles of information systems: A managerial approach* (7th ed.). Boston: Thomson Course Technology.

Whitten, J., L. Bentley and K. Dittman (2004). *Systems analysis and design methods* (6th ed.). Boston: McGraw-Hill Irwin.

Valacich, J., J. George and J. Hoffer (2004). *Essentials of systems analysis and design* (2nd ed.). Upper Saddle River: Pearson Prentice-Hall.