Journal of Information Systems Education, Vol. 17(3)

Information Systems Development Education in the Real World – A Project Methodology and Assessment

Paul J.A. van Vliet
Leah R. Pietron
Information Systems & Quantitative Analysis
University of Nebraska at Omaha
Omaha, NE 68182, U.S.A.
pvvliet@unomaha.edu , lpietron@unomaha.edu

ABSTRACT

Based on ten years of teaching experience by the authors and on a survey of former students, two basic assumptions underlying a comprehensive systems development course are examined. First is the appropriateness of using the Systems Development Life Cycle as a systems development approach in the course. Second is the use of real-world clients for student group projects in the course. Survey results indicate that the SDLC remains a valid vehicle for systems development instruction. Additionally, former students perceive the use of a realistic, comprehensive group project to be a sound preparation for the workplace.

Keywords: Systems Development, Education, Student Survey

1. INFORMATION SYSTEMS DEVELOPMENT EDUCATION IN TRANSITION

Information systems development methodology courses have long been a staple of most universities' undergraduate Information Systems programs. Development methodologies are deemed important because they "impose a disciplined process upon software development with the aim of making software development more predictable and more efficient" (Fowler, 2002). Commonly such courses require that students execute or simulate systems development activities in a comprehensive project. The inclusion of these courses implies the expectation that these courses constitute a valuable addition to the curriculum and a relevant aspect of the training of information systems professionals.

The most commonly used approach in these courses is the Systems Development Life Cycle, which is a comprehensive and traditional method for information systems analysis, design, and maintenance and the staple content of textbooks in the field (Hoffer et al, 2002, p. 18). The usage of the SDLC in systems development practice is, however, undergoing changes. Some significant changes and challenges have begun to affect Information Systems Development educators.

Consequently, the usage of the Systems Development Life Cycle as the generally applicable process of choice is being challenged from many angles. Rapidly changing requirements, smaller systems, and turbulent business environments have called in question the comprehensive nature of the Systems Development Life Cycle (Avison and Fitzgerald, 2003; Fitzgerald and O'Kane, 1997; Yourdon, 2000).

Moreover, the emergence of the World Wide Web as a strategic business tool has lead to the need for web-oriented development methodologies and techniques. Such methodologies would need to accommodate rapidly changing technologies, short delivery times, and the inclusion of multimedia and hypermedia development (Paynter and Pearson, 1998).

The authors of this paper have for a decade taught undergraduate information systems development at an American university employing a comprehensive project approach using real-world projects. The authors are quite aware that the inclusion, design, and execution of a Systems Development course rest on a clear set of assumptions, all of which bear investigation. Based on this, two research questions have been formed which will be investigated in this paper.

- First, is the traditional Systems Development Life Cycle still appropriate to use when teaching Systems Development?
- Second, is the application of Systems Development theory into a comprehensive, realistic group project a sound preparation for the workplace?

This paper will first review the current literature on the changing nature of systems development practices and ofeducational efforts in this field. This is followed by a detailed description of the authors' approach to teaching systems development. The validity of this approach is then demonstrated using the results of a survey administered to former students. The paper finishes with some concluding comments.

2. INFORMATION TECHNOLOGY AND SYSTEMS DEVELOPMENT EDUCATION - A MOVING TARGET

2.1 The Changing Nature of the Information Technology Field

The field of information systems has undergone significant and continuous change. When Royce described what would become the Systems Development Life Cycle in 1970, the computing environment was greatly limited to large transaction processing applications running on mainframe computers in large organizations (Harrison, 2003; Royce, 1970). In the three decades since, the computing environment has become much more varied and complex.

The once monolithic physical implementation environment has become greatly varied to include a broad range of general and special-purpose computing devices. This variety of hardware devices is increasingly expected to interconnect using computer networks. High-quality user interfaces are becoming increasingly important in supporting a great variety of applications and users. All this has allowed information technology to be deployed in support of complex and strategic business applications (Barry and Lang, 2001; Fitzgerald, 1998; Hirschheim et al., 1997).

Simultaneously, employment in the information technology field has undergone related changes. According to a recent survey of Chief Information Officers, skills in Microsoft Windows administration, wireless network management, and SQL server management are in great demand – all of which are beyond traditional systems development tasks (Robert Half Technology, 2005). Additionally, research into the fastest-growing information technology positions in the U.S. shows that while traditional systems developers are still in strong demand, a substantial number of non-systems development positions experiences substantial growth (Lee, 2005).

2.2 Changes in Information Systems Development Practices

The origins of the Systems Development Life Cycle can be found in (Royce, 1970) in which a sequence of comprehensive phases guides the development of large software systems. These phases – systems requirements determination, software requirements determination, analysis, program design, coding, testing, and operations – separate activities and development skills so that a project can be forwarded linearly towards implementation. Given the SDLC's broad area of application, it rapidly became the de-facto standard for both information systems development and information systems development education (Harrison, 2003; Hoffer et al., 2002).

The aforementioned changing hardware, interfaces, and application requirements, along with a more turbulent business environment, have called into question the general applicability of the Systems Development Life Cycle. Moreover, the SDLC has been criticized as being inflexible, time-consuming, costly, bureaucratic, cumbersome, and involving too much paperwork (Avison and Fitzgerald, 2003; Barry and Lang, 2001; Fitzgerald, 1998b; Fitzgerald and O'Kane, 1997; Yourdon, 2000). Consequently, some organizations have abandoned the SDLC and replaced it with a toned-down approach customized to the development situation (Fitzgerald, 1998; Fitzgerald and O'Kane, 1997). These approaches are typified by having shorter analysis phases, fewer documentation requirements, and greater user involvement. These approaches are commonly referred to as "light" or "agile" and include Extreme Programming, Rapid Application Development, Crystal Family, and the Rational Unified Process (Fowler, 2002; Kruchten, 1998; Yourdon, 2000).

In addition, other systems development approaches have been introduced in organizational settings. Object-oriented approaches, tool-based approaches, incremental approaches, iterative and prototyping approaches, and others have been used with varying levels of success (Avison and Fitzgerald, 2003; Center for Technology in Government, 2000; Coleman and Zilora, 2003).

The development and application of these various methods and methodologies has resulted in the coining of the phrase "methodology jungle," which is described by Hirschheim et al. (1997) as "an unorganized collection of numerous methodologies which are more or less similar to each other." According to Coleman and Zilora (2003), none of these new methodologies has achieved universal applicability or received widespread acceptance.

2.3 Systems Development Education and its Relevance

Despite these changes, challenges, and critiques, the information systems development methodology course remains a common feature of undergraduate Information Systems programs. The use of development methodologies in a curriculum are deemed important because they "impose a disciplined process upon software development with the aim of making software development more predictable and more efficient" (Fowler, 2002). Indeed, the model curriculum for undergraduate degree programs in information systems provided by the Special Interest Group on Information Technology Education of the Association for Computing Machinery contains a full section of courses called "System Integration and Architecture," which heavily favors the use of the Systems Development Life Cycle phases (SIGITE, 2005, p. 97).

The question arises whether it continues to make sense to teach the Systems Development Life Cycle in today's undergraduate curricula despite the changes in the application environment and the availability of other methodologies and techniques. Educators should keep in mind the knowledge and skills their students need to obtain, especially in a field as dynamic as information systems technologies (Lopes and Morias, 2002). At the same time, the limitations of the classroom setting can't be ignored:

student work tends to be limited to either small or partial projects, at times taken directly from a textbook, and only a limited amount of time is available for the teaching of the development methodology and the application of these skills to a project.

As mentioned earlier, the profession of systems developers has undergone substantial changes. Given those changes, it might seem as if the Systems Development Life Cycle – or other comprehensive development approaches – is out of date and should be replaced in the classroom, but this is not necessarily true. According to Fitzgerald (1998, 1998b) and Fitzgerald and O'Kane (1997), inexperienced developers often use a formal methodology they have been taught as a template to follow, as a means to proceed through the development phases and tasks. Hirschheim et al. (1997) echo that "methodologies are primarily intended for beginners as the primary vehicle by which they are initiated into the field."

Once these developers gain experience, they may realize that such "blind adherence" to "universally applicable methodological prescriptions" (Fitzgerald and O'Kane, 1997) is not the most productive approach as "a methodology may not be able to recognize all situational factors" (Hirschheim et al., 1997). Over time, then, developers discover the appropriate level of methodology to apply to a given development situation (Fitzgerald and O'Kane, 1997). So rather than following a development methodology rigorously, experienced developers, drawing from their expertise and experience and based on the actual development situations, depart from a methodology "in a conscious and deliberate fashion, rather than an arbitrary one" (Fitzgerald, 1998). This "methodology-in-action" approach customizes the development process deliberately, omitting some tasks and supplementing others (Fitzgerald, 1998).

Consequently, a requirement for a systems developer to arrive at this balanced, sensible usage of a development methodology does require the first step: the learning of a comprehensive systems development methodology.

2.4 Teaching Systems Development: The Project

An important and common component of any Information Systems Development course is a student project in which theory is put to practice. A good starting point for investigating the appropriateness of such a project is in professional curriculum recommendations. The most current guidelines are the 2005 Computing Curricula recommendations provided by the Special Interest Group on Information Technology Education of the Association for Computing Machinery. Among recommended program outcomes, SIGITE includes the application of technical concepts and practices, the design of effective IT-based solutions, the demonstration of best practices and standards, the development of problem solving skills, and collaboration in goal-oriented teams (SIGITE, 2005).

Emphasizing that IT professionals require "a familiarity with the technology that goes beyond the purely theoretical," the SIGITE report describes an "Integrative Capstone Experience" consisting of students working in small groups to solve a real world project which takes a substantial amount of time to complete. The SIGITE report cites research support for these projects, but all six of its included references are in the area of engineering, not information systems (SIGITE, 2005).

Fortunately a small number of instructors have documented their experiences with the Systems Development capstone project in some detail, particularly at educational conferences such as ISECON. As can be expected, different instructors have instituted a variety of projects to match their diverse educational environments.

The first point of difference is the variety of systems development projects that are described. Some instructors have chosen for a prepared case study to be used for the students (Tan and Phillips, 2005).

Other instructors have found that case studies were not sufficiently engaging to the students. As Sherman (2000) puts it: "Students know they're not real. They don't invest themselves emotionally in solving the problems these case studies present." Consequently many instructors have selected to use real-world projects, but these projects differ greatly in organization, size, and scope. Most common is the use of multiple real-world clients, which gives each student group a unique and realistic experience. For example, Sherman describes a project in which students develop web pages for university faculty members (Sherman, 2000). Frandsen and Rhodes describe the use of real-world systems development projects and how these projects are supported by different faculty members (Frandsen and Rhodes, 2002). Scott's approach is to define a generic business problem first, then search for businesses which match this problem and can support the student groups, thereby ensuring comparability of projects (Scott, 2004).

An approach which limits the number of projects for the instructor to manage is by having all student groups do the same project for a single client (Ellen and West, 2003; Laware and Walters, 2004; Poger, Schiaffino, and Ricardo, 2005).

Aside from the variety of projects used, available documentation also varies in the amount of detail in which the projects are described. Some authors describe the systems that were developed by the students in great detail (Poger, Schiaffino, and Ricardo, 2005). Other authors provided clear listings or descriptions of the project deliverables for the different projects (Sherman, 2000; Tan and Phillips, 2003;Tan and Phillips, 2005). Yet other authors provided little information about the actual projects or their deliverables, focusing instead on describing the experiences (Ellen and West, 2003; Fox, 2002; Frandsen and Rhodes, 2002; Green, 2003; Jensen and Wee, 2000; Laware and Walters, 2004).

All these variations make it difficult to compare instructor expectations, course requirements, and educational outcomes of the documented courses. What these papers do show is that systems development instructors have made serious attempts to adopt the SIGITE recommendations for the capstone project. The various circumstances and environments of the different instructors may explain a lot of the variation in the employed project assignments.

What most of these papers also have in common is that they are written from the instructor's perspective. (This is, of course, to be expected in papers that describe the approaches to teaching the systems development courses.) From these papers, the impression arises that the more challenging projects – those involving multiple real-world clients – require substantially greater amounts of faculty effort and time. These efforts involve the recruitment of real-world clients for project assignments (Jensen and Wee, 2000); the accompaniment of the systems development projects (Fox, 2002; Laware and Walters, 2004); the management of student groups (Tan and Phillips, 2003); and a more complex grading effort when students not only work in groups but also on dissimilar project assignments (Sherman, 2000; Tan and Phillips, 2005).

What makes these additional instructor efforts worthwhile is, according to several papers, the enhanced educational value of the real-world project approach. The more realistic projects allow students "to learn what is relevant to the environment in which they will be working," (Ellen and West, 2003) and thus reduces the problems of students having "a hard time adjusting to real work jobs after graduation" (Tan and Phillips, 2003). More specifically, students were thought to have obtained better project management and time management skills (Tan and Phillips, 2005), a better appreciation of development methodologies and frameworks (Laware and Walters, 2004), improved oral and written communications skills (Fox, 2002), enhanced problem solving skills (Frandsen and Rhodes, 2002) and a better appreciation of group dynamics (Jensen and Wee, 2000).

Few of the papers, unfortunately, directly assessed the student perspective directly. Ideally, students would be asked to evaluate the utility and validity of the systems development courses a substantial time after they had entered the workplace. Only then would they be able to compare the educational opportunities offered by the real-world course projects with their professional lives.

3. A REAL-WORLD PROJECT-ORIENTED APPROACH TO SYSTEMS DEVELOPMENT EDUCATION

The authors of this paper have co-taught a set of information systems development courses for the past ten years in the Information Systems department at an American university. The content is spread out over two courses: Information Systems Analysis and Systems Development & Implementation. Students commonly take these courses in their senior year after having taken courses in Database Design, Data Communications, and other supporting subjects. This foundation prepares the students for a comprehensive systems development experience.

The objectives of the courses are to assist our students in understanding the job requirements of systems developers, both in the analysis and design stages; to explore the systems development process; to experience work on real-world projects with real clients and users; and to develop group skills, interpersonal skills, and verbal and written communications skills.

The teaching in these two information systems development courses focuses on the Systems Development Life Cycle. This approach is used to put together all the information systems foundation content in a detailed and complete process by which the information system is to be developed. The focus is on the use of a sound structured development methodology and on best practices in analysis, design, construction, implementation, and documentation.

In the Information Systems Analysis course, students are organized in groups of three or four students. At the start of this course, they are assigned a small systems development project for a real client. In recent years, such clients have included both large and small companies, non-profit organizations such schools and charities, and some university departments. The students complete the analysis portion of the project in the first course and then continue the design and development of the project in the Systems Design & Implementation course. Approximately 90 students enroll in the courses annually, and around 23 projects are completed each year.

The students' systems development projects are completed through the use of eight sequenced deliverables – commonly referred to as milestones – which guide the students through each phase of the Systems Development Life Cycle. These milestones address technical skills – such as various diagrams, designs, and code development – as well as communications skills – such as technical writing and documentation development. The following are the main deliverables of the milestones of the two courses:

- **Systems Service Request** – This is a form which includes the name and contact information of the person requesting the system, a statement of the problem, and the name and contact information of the liaison and sponsor.
- **Statement of Understanding** – This is a short document prepared for the customer that describes what the project will deliver in terms of hardware, software, and security requirements, and outlines the work required to complete the project.
- **Economic Feasibility Analysis and Project Management**– The identification of the financial costs and benefits associated with the development project. The project management lays out the project for both semesters.
- **Baseline Project Plan** – The document contains all the information collected and analyzed during project initiation and planning. It reflects the best estimate of

the project's scope, benefits, costs, risks, and resource requirements given the current understanding of the projects.

- **Systems Profile** – This is a form which focuses on collecting information about purposes, goals, and objectives of the system. In addition, it identifies the inputs and outputs for the potential system along with software and hardware requirements.
- **Statements of Scope and Purpose** – These documents identify the opportunity or statement of problem, objectives, description, benefits, deliverables, and estimated completion date.
- **Information Security Risk Analysis and Information Systems Security Policies**– This document is an assessment of the client's information security needs, covering assets, threats, vulnerabilities, losses, and safeguards. These polices are created based on the Information Security Risk Analysis.
- **Process and Logic Modeling** – This process modeling diagrams show the processes which capture, manipulate, store, and distribute data between a system and its environment and between the components within the system. The logic modeling decomposes the data structures in the data dictionary.
- **Entity Relationship Diagram** – This data modeling diagram shows the items about which data will be stored, their internal structure, and their interrelationships
- **Computer Architecture Design** – A specification of the hardware environment of the information system.
- **Systems Controls Plan** – The detailed design of specific controls to be built into the new information system.
- **Input-Output and Interface Design** – Designs of the menus, forms, and reports that make up the information system. Focus is placed on the clear presentation of information.
- **Structure Chart** – This process modeling diagram shows the hierarchical organization and interconnections the application's processes in ever-increasing detail. It is accompanied by the designs of the individual software modules and a specification of the interaction of the software modules with the database.
- **Testing Plan** – A comprehensive description and schedule of the system's testing procedures, ranging from module testing to end-user testing.
- **Implementation Plan** – A comprehensive description and schedule of all activities related to the installation and activation of the information system, including end user training and data conversion.
- **Maintenance Plan** – A description of the procedures to be followed for corrective, adaptive, perfective, and preventive maintenance activities.
- **Systems Documentation** – Following completion of the project, all development documents are gathered and organized to serve as a record of the information system as completed and as a starting point for systems maintenance.
- **End User Documentation** – Following completion of the project, a comprehensive manual is developed for the system's users.
- **Executive Summaries** – Each milestone is preceded by a one-page summary of the milestone contents for the project client, devoid of information systems jargon.
- **Communication** – Each milestone requires that groups detail the process in which they communicate and make decisions. Copies of emails, agendas, meeting notes and action items are included in this section.

Over the course of the project the students are evaluated and graded as follows:

- Each course has two examinations, which is the only time in both courses when students specifically complete individual work. The examinations make up a third of a student's grade in the course.
- The Information Systems Analysis course requires students complete tutorials on Microsoft Project 2003 and complete a case project. This course also requires homework assignments in processing and logic modeling: DFDs and IDEF models. The logic modeling includes exercises in decision tables, structured English, and algebraic notation.
- Each of the eight deliverables/milestones is graded. The student groups receive a group grade for each milestone. These milestones also make up a third of the student's grade in the course.
- In each of the two courses, the groups present their work, for which they receive a group grade.
- At the end of the second course, the final project, which includes the application, the user manual, and the systems documentation, receives a group grade. The presentation and final project jointly make up the final third of the student's grade in the Systems Design & Implementation course.

  For each deliverable, and for the final project, students submit peer evaluations in which they assess the efforts of each member of their group, including their own. These evaluations are confidential and are a good source of information for the instructor, both for grading and team management purposes. If the peer evaluations are consistent regarding the lack of effort of a group member, then that student's grade for that deliverable is reduced to an extent consistent with the group's evaluations.

4. A SURVEY OF FORMER INFORMATION SYSTEMS DEVELOPMENT STUDENTS

4.1 Introduction

Insights on the link between undergraduate systems development education and its relevance to the workplace were gathered from the results of a longitudinal survey performed by the author. This survey was addressed to former undergraduate students who had completed the set of Information Systems Analysis and Design courses described above. The survey was sent to students one year after the students completed the courses, which occurred over the period of three years. These students were asked about their perspective on the usefulness of the design of the systems

The survey was sent to these former students using e-mail. As was expected, a substantial number of surveys were returned as undeliverable, due to students having abandoned	**All 100% n=115**	**System Developers 46% n=53**	**Non-SD IT Jobs 32% n=37**	**Non-IT Jobs 22% n=25**
1. In my current job, I use a similar development methodology as used in class.	53%	79%	30%	29%
2. In my current job, I use similar development tools as used in class.	43%	62%	30%	21%
3. In my current job, I use similar diagramming techniques as used in class.	48%	67%	44%	31%
4. In my current job, I use similar documentation techniques as used in class.	61%	77%	57%	30%

Percentages indicate respondents answering positively to questions.

Table 1. Methodology, Tools, and Techniques Results

development courses from the perspective of their current jobs.

The survey was sent to these former students using e-mail. As was expected, a substantial number of surveys were returned as undeliverable, due to students having abandoned older e-mail addresses. In the end, approximately 300 surveys were actually delivered to valid student e-mail addresses and of these 115 valid responses were gathered.

Of the responding students, 60% was male, 40% female. 91% of the respondents took the courses as part of an undergraduate Management Information Systems program; with the remainder of the respondents having enrolled in undergraduate Computer Science or Business Administration programs instead. Out of the respondents, 77% had completed their undergraduate program at the time of the survey.

While they were enrolled in the Systems Development courses, 69% of the respondents were already employed in some type of Information Technology-related job. About half of respondents – 54% – worked full time while enrolled in the courses, while 46% worked part-time during this period.

At the time of the survey, 46% of the respondents were employed in Information Systems Development and/or Maintenance jobs; 32% were employed in other types of Information Technology-related jobs; 22% were employed in non-IT jobs.

Before analyzing the results an important limitation of the study must be mentioned, which is the lack of a control group in the measures. All students enrolled in the course were required to take on a real-world client for their systems development project, so no intra-course comparison could be made to book-based projects. When comparing the real-world client project to a book-based project, the respondents are drawing on experiences with book-based projects from other courses.

4.2 The Relevance of the Systems Development Life Cycle

In order to assess the relevance of the Systems Development Life Cycle in today's work environment, the former students were first asked whether the systems development approach, tools, and techniques used in the courses were perceived as relevant, given their post-course experiences. Of course, the relevance of this approach to systems development education to a student's current job relies to a great extent on the actual job the student has. As part of the survey, the student's job-type was recorded. The responses on the questions have been separated in order to check to what extent the above results were driven by the 46% of the respondents who are currently in systems development jobs. The relevant questions and responses for this section are shown in Table 1.

It is interesting to note in Table 1 that there is good support for the use of the Systems Development Life Cycle (SDLC) among those former students who work in the systems development field. This could indicate that systems development methodologies employed in practice borrow heavily from the SDLC, even if they are not specifically referred to as such

The divergence on tools and techniques was expected due to the variety of tools and techniques available. In additional comments, a number of respondents mentioned the Unified Modeling Language (UML) as a diagramming technique of choice.

Additionally, the former students were asked whether they perceived the in-course systems development deliverables as relevant given their post-course experiences. The respondents responded substantially favorably on this issue. The relevant questions and responses for this section are shown in Table 2.

Based on the results in Table 2, the activities performed in the systems development courses were relevant to the world of work. It is interesting to note the strong support for very traditional techniques – data-flow diagrams and entity-relationship diagrams – as well as for project-management deliverables – the baseline project plan, the testing plan, the implementation plan, and the statement of scope and purpose. This matches findings on the adoption of systems development tools and techniques by (Barry and Lang, 2001) as well as by Fitzgerald (1998b), who states that "those using methodologies use <u>all</u> of these tools and techniques. This lends support to the argument that methodologies provide a suitable framework to co-ordinate the purposeful application of tools and techniques" (Fitzgerald, 1998b). Finally, the

	System Developers 46%, n=53	Non-SD IT Jobs 32%, n=37	Non-IT Jobs 22%, n=25
Systems Service Request	70%	67%	68%
Statement Of Understanding	63%	61%	58%
Economic Feasibility Analysis	65%	77%	64%
Baseline Project Plan	81%	78%	72%
Systems Profile	70%	81%	68%
Statement of Scope and Purpose	83%	84%	80%
Information Security Risk Analysis	85%	81%	64%
Data Flow Diagram	85%	70%	60%
Entity Relationship Diagram	93%	70%	60%
Computer Architecture Design	79%	61%	42%
System Controls Plan	69%	73%	53%
Input-Output & Interface Design	85%	65%	59%
Structure Chart	68%	62%	72%
Testing Plan	94%	81%	70%
Implementation Plan	94%	77%	64%
Maintenance Plan	81%	73%	53%
Systems Documentation	87%	89%	64%
End User Documentation	77%	89%	64%
Executive Summaries	70%	79%	72%

Percentages indicate respondents answering positively to questions.

Table 2. Project Deliverables Results.

increased interest in information systems security is reflected in the strong support for the information security risk analysis.

4.3 The Relevance of the Real-World Project

In the survey, eight questions specifically addressed the use of a real-world client for the systems development project. The fourth question was reversely scored for validation purposes. The results of all respondents to these questions are reported in Table 3.

Two small observations about these results: first, the 'negative' question "I would rather have done a book project than a project for a real client" had the strongest 'Completely Disagree' response in the survey; 77% of all respondents chose this option. Second, the question "Working on a real client project provided more learning opportunities than a book project would have provided" had the strongest 'Completely Agree' response on the survey; again 77% of all respondents chose this option.

Three things are noticeable about this analysis. First, it is quite clear that those respondents in systems development jobs and those in other information technology jobs are generally in high agreement on most questions.

Second, out of the eight questions, there are four which were expected to score lower for those respondents not in information technology jobs; questions 1, 2, 3, and 8 specifically relate to information technology and systems development jobs. Indeed, the scores for respondents in non-IT jobs are lower for these questions. What is interesting is that for the four other questions, the scores for respondents in non-IT jobs are not greatly different from those in IT jobs. Especially the response to question 7 – "Working on a real client project provided morelearning opportunities than a book project would have provided." – indicates that the real-world project approach was perceived as valuable even by those not in IT jobs.

Third, it is interesting to see that the respondents are quite aware of one of the main drawbacks of the use of real-world projects: the additional time (and hence effort) it takes to complete real-world projects. (Question 6.) As mentioned earlier, real-world projects have less structure and higher levels of uncertainty than the more clearly defined book-based projects, and therefore require more time to complete. Despite this additional effort, the respondents were highly positive about the learning experience. This supports the assumption that real-world projects result in higher student involvement and consequently in greater student effort.

5. CONCLUDING COMMENTS

The purpose of this paper was to examine the changes in the field of information systems development, the subsequent changes in systems development education, and the appropriateness of the authors' approach to the teaching of systems development.

It was found that while systems development practices are continuously changing to accommodate changes in the information technology field, traditional approaches to information systems development remain relevant and appropriate. The teaching of systems development, therefore, should not dismiss the original principles of the systems development life cycle, as they provide the foundation of many current development approaches.

	All 100%	System Developers 46% n=53	Non-SD IT Jobs 32% n=37	Non-IT Jobs 22% n=25
1. Working on a real client project has been a useful preparation for my job.	88%	98%	95%	60%
2. Working on a real client project provided a realistic simulation of actual systems development work.	88%	94%	86%	76%
3. Working on a real client project has been useful for my activities in my current position.	74%	91%	81%	28%
4. I would rather have done a book project than a project for a real client.	6%	6%	5%	4%
5. Working on a real client project made the system development effort more predictable.	52%	51%	51%	52%
6. Working on a real client project took more time than a book project would have taken.	79%	70%	86%	84%
7. Working on a real client project provided more learning opportunities than a book project would have provided.	94%	94%	95%	92%
8. Working on a real client project was a good preparation for the demands of my current job.	76%	85%	89%	41%

Percentages indicate respondents answering positively questions.

Table 3. Real-World Client Project Results Separated by Job Type.

The authors' use of a comprehensive systems development project for a real-world client met with substantial approval from past students. While this approach requires a substantial commitment in time and effort from the instructors, the long-term benefits for the students are quite clear.

6. REFERENCES

Avison, D.E. and Fitzgerald, G. (2003), "Where now for development Methodologies?" Communications of the ACM, Vol. 46, No. 1, pp 79-82.

Barry, C. and Lang, M. (2001), "Contrasting Approaches – A Study of Traditional and Multimedia Information Systems Development in Ireland." Proceedings of the Irish Academy of Management Conference, September 6-7.

Center for Technology in Government (2000), "Models for Action: Developing Practical Approaches to Electronic Records Management and Preservation." Internet Resource.
URL: http://www.ctg.albany.edu/projects/er/sdpm2.pdf

Coleman, D.G. and Zilora, S.J. (2003), "Dynamic Enterprises Demand Advanced Curricula in Software Development and Management." Proceedings of the Fourth Conference on Information Technology Curricula, October 16-18.

Ellen, N. and West, J. (2003), "Classroom Management of Project Management: A Review of Approaches to Managing a Student's Information System Project Development." Journal of American Academy of Business, September 2003, pp. 93-97.

Fitzgerald, B. (1998), "An Empirically-Grounded Framework for the Information Systems Development Process." Proceedings of the International Conference on Information Systems, December, 1998.

Fitzgerald, B. (1998b), "An Empirical Investigation into the Adoption of Systems Development Methodologies." Information & Management, Vol. 34, pp. 317-328.

Fitzgerald, B. and O'Kane, T. (1997), "The Use of Systems Development Methodologies in Practice: A Field Study." The Information Systems Journal, Vol. 7, No. 3, pp 201-212.

Fowler, M. (2002), "The New Methodology." Internet Resource. URL: http://www.martinfowler.com/articles/newMethodology.html

Fox, T.L. (2002), "A Case Analysis of Real-World Systems Development Experiences of CIS Students." Journal of Information Systems Education, Vol. 13, No. 4, pp. 343-350.

Frandsen, M L and Rhodes, L.K. (2002), "Local Industry - Student Team Collaboration on IT Projects: Experiences with a Multi-Semester Experiential Learning Course Sequence." Proceedings of ISECON 2002, Vol. 19: §343b.

Green, L. (2003), "Projecting IT Education into the Real World." Proceedings of the Conference on Information Technology Curriculum, October 16-18.

Harrison, W. (2003), "Is Software Engineering as We Know It over the Hill?" IEEE Computer, May/June 2003.

Hirschheim, R., Iivari, J, and Klein, H.K. (1997), "A Comparison of Five Alternative Approaches to Information Systems Development." Australian Journal of Information Systems, Vol. 5, No. 1.

Hoffer, J.A., George, J.F., and Valacich, J.S. (2002), Modern Systems Analysis & Design, Third Edition. Prentice Hall, Upper Saddle River, NJ, U.S.A..

Jensen, J and Wee, L.C. (2000), "Creating Real Life Project Opportunities for Systems Analysis and Design Students." Proceedings of ISECON 2000, Vol. 17: §608.

Kruchten, P. (1998), The Rational Unified Process – An Introduction. Addison Wesley Longman, Inc. Reading, MA, U.S.A.

Laware, G.W. and Walters, A.J. (2004), "Real World Problems Bringing Life to Course Content." Proceedings of Special Interest Group on Information Technology, October 28-30.

Lee, Katherine S. (2005), "IT Compensation on the rise in '05." Computerworld, January 18, 2005.

Lopes, F. and Morais, P. (2002), "Lessons Learned from the Teaching of IS Development." Journal of Information Technology Education, Volume 1, No. 2.

Paynter, J. and Pearson, M. (1998), "A Case Study of the Web-Based Information Systems Development." Internet Resource. URL: http://www.cecil.edu/reports/www_Case_Multimedia_98.pdf

Poger, S., Schiaffino, R., and Ricardo, C. (2005), "A software development project: a student-written assessment system." Journal of Computing Sciences in Colleges, Vol. 20, No. 5, pp. 229-238.

Robert Half Technology. (2005), "Cio Survey Forecasts Stable Second-Quarter Technology Hiring. Internet Resource. URL:http://www.roberthalftechnology.com/PressRoom;jsessionid=CEr3pBRnYV1BljexMsEIeu5VwdsYHEtEl3420OiO74d0nSSmQxg0!1775009546!-877379553

Royce, W. (1970), "Managing the development of large software systems: Concepts and techniques." WESCON Technical Papers, v. 14, pages A/1-1-A/1-9, Los Angeles, August 1970. WESCON. Reprinted in Proceedings of the Ninth International Conference on Software Engineering, 1987, pp. 328-338.

Scott, E. (2004), "Systems Development Group Project: A Real World Experience." Proceedings of ISECON 2004, Vol. 21: §4134.

Sherman, C A. (2000), "The Perfect Systems Analysis Project." Proceedings of ISECON 2000, Vol. 17: §116.

SIGITE (Special Interest Group on Information Technology Education of the Association for Computing Machinery) (2005), "Computing Curricula." Internet Resource. URL: http://www.acm.org/education/curric_vols/IT_October_2005.pdf

Tan, J. and Phillips, J. (2003), "Challenges of real-world projects in team-based courses." Journal of Computing Sciences in Colleges, Vol. 19, No. 2, pp. 265-277.

Tan, J. and Phillips, J. (2005), "Real-World Project Management in the Academic Environment." Journal of Computing Sciences in Colleges, Vol. 20, No. 5, pp. 200-213.

Yourdon, E. (2000), "The Emergence of 'Light' Development Methodologies." Internet Resource. URL: http://www.spc.ca/essentials/oct1800.htm#3

AUTHOR BIOGRAPHIES

Paul J.A. van Vliet holds a bachelor in business informatics from the H.E.A.O. in Arnhem, The Netherlands, an MBA from Moorhead State University, and a Ph.D. in information systems from Oklahoma State University. He now serves as Associate Professor at the University of Nebraska at Omaha. He specializes in information systems development education and information systems architecture design.

Leah R. Pietron holds a bachelor of science from Mayville State College, MS and PhD from the University of North Dakota, and a MBA from Northwest Missouri State University. Dr. Pietron has done post-doctoral work at University of Minnesota and Indiana University. She serves as an Associate Professor at the University of Nebraska at Omaha. She specializes in information systems development education and information systems security.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096