Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2004

Using IS2002 to Assess the Impact of a Proposed Curriculum Change

Jeffrey Landry
University of South Alabama

J. Harold Pardue
University of South Alabama

Roy Daigle
University of South Alabama

Herbert Longenecker
University of South Alabama

Using IS2002 to Assess the Impact of a Proposed Curriculum Change

Jeffrey P. Landry
University of South Alabama
jlandry@usouthal.edu

J. Harold Pardue
University of South Alabama
hpardue@jaguar1.usouthal.edu

Roy J. Daigle
University of South Alabama
rdaigle@usouthal.edu

Herbert E. Longecker, Jr.
University of South Alabama
hlongenecker@usouthal.edu

ABSTRACT

This paper describes how a process for mapping the curriculum of a four-year IS degree program (Daigle et al. 2004), using the IS2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (Gorgone et al. 2002), is useful for analyzing the impact of a proposed curriculum change. An example, based on an actual use of IS2002 software tools, is used to illustrate the process. The example is based on an actual case of comparing two courses, Software Engineering Principles and Information Technology Project Management, to determine which was a more appropriate required course for IS majors. The example illustrates the value of IS2002 software tools for identifying areas of value-added breadth and depth, determining areas of overlap, and following the threads of how important concepts are developed over the course of the degree program.

Keywords

IS2002, model curriculum, curriculum development, IS accreditation

INTRODUCTION

Recent trends towards accountability in IS education have lead to a process of accreditation for four-year degree programs in information systems (ABET CAC 2002). As more and more faculty become interested in accreditation, increasing use of the IS2002 model curriculum as a means of curriculum self-study should follow. The purpose of this paper is to describe a specific example of a use of the IS2002 model curriculum to demonstrate the power inherent in the model curriculum, especially when it is used with software, for evaluating and improving IS courses and curricula.

This paper utilizes a process described in Daigle et al. (2004). The process, called "mapping a curriculum," involves defining which local courses—the courses that IS students at a university are required to take—cover which learning units in the IS2002 model curriculum. For each local course, faculty working in an IS department must define a depth level and local objective for each learning unit covered by a course they teach. The depth metric is an integer from 1 to 4 in increasing depth, and the local objective is a statement which reflects how that learning unit is covered in the local course. It should be written like a syllabus objective and contain a learned capability verb. See Figure 1 below for examples of local objectives.

BACKGROUND—THE CURRICULUM UPDATE SITUATION

Recently, the information systems faculty at the University of South Alabama School of Computer and Information Sciences (CIS) met to discuss and propose changes to its 4-year information systems degree program's curriculum. Among the changes considered was replacing the CIS 231 course, Software Engineering Principles, with ITE 475, Information Technology Management, as a required course for IS majors. Software Engineering Principles, taught by a member of the computer science faculty, had long been a required course for IS majors, and a prerequisite to the Systems Analysis and Design Sequence. The IT Project Management course had been recently added as a new course required of its new information technology specialization. Because of the perception that the IT Project Management course covered material fundamental to and perhaps missing from the IS curriculum, the IS faculty wanted to pursue its inclusion as a required course. To make room for another required course, the faculty was considering Software Engineering Principles as the course to replace. The faculty also perceived that software engineering was an area more suited to computer science that had

significant overlap with project management and systems analysis and design and was less relevant to IS majors, also having been taught by CS faculty. Therefore, students would least miss this course.

LEARNING UNIT COMPARISON REPORT

The author prepared a report, using spreadsheet-based reports of data obtained from a query of the IS2002 database (www.iseducation.org). The report is called "learning unit comparison report" because it compares two local courses on the basis of how they cover the learning units of the IS2002 model curriculum.

Each row represents coverage of a learning unit by one or both local courses. The learning unit is described in the "learning unit" column in the form <logical course>.<learning unit number> - <learning unit short title>. Missing from the report are the learning unit's goal and associated objectives, which, if seen would provide a more in-depth understanding of how the local courses relate to the learning unit.

The "levels" columns for each course, listed as "CIS-231 Lvl" and "ITE-475 Lvl," respectively, lists the educational depth of coverage of a learning unit in that course. The scale used reflects what students will be able to do, in the sense of learned capability, and is defined in the IS2002 model curriculum as follows: 1-recognize, 2-differentiate, 3-use, and 4-apply.

The "local objectives" columns for each course, appearing as the first ("CIS-231 Obj") and last ("ITE-475 Obj") columns in the report, are more than just syllabus objectives. They were written after careful consideration of each local course's relationship to a learning unit in the model curriculum, and reflect how the local course addresses the learning unit. When compared to the learning unit, they reflect both a sense of the depth of coverage of that learning unit, and how that particular learning unit is interpreted by the local course. When comparing two courses that teach the same learning unit, the local objectives can be compared to see if the two local courses are doing something similar are different, or if one course's coverage subsumes or provides prerequisite coverage of that learning unit.

The "coverage" column lists the number local courses that cover the learning unit, including the two courses being compared. Large numbers provide a measure of topic overlap. More than one coverage of a learning unit should be expected, however, both in terms of varying depths of coverage, which should roughly increase throughout the four year-curriculum, and in terms of the manner in which the learning unit is covered, in terms of specific topic content.

To provide more insight into whether courses overlapped unnecessarily in learning unit coverage, of if they provided unique coverage or allowed for spiral learning, a second report, not shown in this paper, was used. The report provides more in-depth analysis into how a learning unit is covered across the curriculum. For example, if a learning unit was said to be covered five times, the more in-depth report provided a listing of each local course, level of depth of coverage, and local objective for each of the five courses that cover the learning unit.

The complete report appears as Table 1 below.

CIS-231 Obj	CIS-231 Lvl	Learning Unit	Coverage	ITE-475 Lvl	ITE-475 Obj
Discuss the importance of Total Quality Management	1	01.6 - Information and Quality	3	-	-
Represent decision making processes visually in a decision tree (describe advantages offered by decision trees)	2	03.21 - Decision Making, Simon Model	2	-	-
Describe basic design concepts such as basic steps of design, modularity, cohesion, coupling, and refactoring	2	05.50 - Problem Solving, Modules/Cohesion/Coupling	5	-	-
List and describe various techniques of evaluation	1	08.95 - IS Database Conceptual/Logical Models	9	-	-
Describe various design techniques; Describe the characteristics of the essential and implementational views in the	1	08.96 - IS Functional Specifications	7	-	-

CIS-231 Obj	CIS-231 Lvl	Learning Unit	Coverage	ITE-475 Lvl	ITE-475 Obj
design process; Describe and construct a data dictionary					
Diagram, describe, and discuss configuration management	2	08.98 - IS Development and Conversion	5	-	-
Diagram and describe the purpose of UML class diagrams (with data, operations, cardinality, class associations), state diagrams, activity diagrams, collaboration diagrams, and use cases	3	08.99 – IS Requirements/Work-Flow Planning	6	-	-
Describe and apply the options typically available in a make versus buy decision	2	00.13.08 - Is life cycle: developing with packages	8	2	Define and distinguish between key information systems analysis and design activities relevant to IT projects and the different types of IT projects (packaged vs. in-house vs. contract/custom)
Discuss and diagram software development models	2	03.28 - IS Development Standards	6	2	Differentiate among various systems development life cycle models including waterfall, prototyping, and spiral models.
Discuss the importance of Total Quality Management and the Capability Maturity Model	1	03.30 - Personal, Performance Evaluation	6	1	Recognize the value of process improvement initiatives, such as the Capability Maturity Model (CMM) for software
Describe and discuss the process of developing project requirements	1	07.74 - IS Requirements and Specifications	7	2	Compare and contrast a variety of different IS requirements analysis techniques
Discuss and describe risk management (RMMM) versus crisis management (pro-active versus reactive strategies), and identify risks and draw up risk tables	2	07.77 - IS Development Risks/Feasibility	8	1	Identify and assess information systems project risks
Use teamwork and communication skills to support the development of a software project	3	07.79 - Interpersonal, Consensus Development	10	3	Identify group process losses and ways to manage IS project teams to overcome the limitations of groups
Describe team paradigms for software development	2	07.80 - Interpersonal, Group Dynamics	7	1	Describe common issues involving communication and conflict among groups involved in an IS development project
Define and apply concepts relating to metrics, indicators, measurements, and measures	2	07.83 - IS Software Quality Metrics	2	1	Identify basic software quality metrics, such as fault density, fault detection rate, and fault detection efficiency
Use communication skills to support the development of a software project	3	08.117 - Personal, Presentation	6	4	Make an oral and written report in the role of an IS project management consultant as part of the semester team project
Demonstrate discipline-based motivation, initiative, organizational skills, and a strong work ethic	2	08.87 - Interpersonal, Agreements and Commitment	7	3	Estimate durations and schedule tasks, using milestones to help manage their completion
Discuss the purpose of testing and how that purpose impacts the implementation of testing procedures	1	09.103 – IS Development Testing	7	1	Describe the what, who, when, and why of testing
Use teamwork and demonstrate discipline-based motivation, initiative,	3	09.113 - Interpersonal, Empathetic Listening	5	2	In the role of a project manager, differentiate among conflict resolution alternatives, applying

CIS-231 Obj	CIS-231 Lvl	Learning Unit	Coverage	ITE-475 Lvl	ITE-475 Obj
organizational skills and a strong work ethic					the concept of emotional intelligence
Describe project scope and how it is built; List and describe various techniques of evaluation; Discuss the purpose of testing and how that purpose impacts the implementation of testing procedures; Describe and discuss various types of testing: white box, black box, unit, integration, regression	2	09.116 - IS Life Cycles and Projects	7	1	Explain how software engineering techniques like testing and metrics affect IS projects and can improve quality and lead to a mature software process
Represent decision making processes visually in a decision tree (describe advantages offered by decision trees)	2	1.10 - Characteristics of an IS Professional	5	1	Recognize that behavioral issues motivation, trust, participation, and leadership affect the dynamics of IS projects.
Demonstrate a commitment to quality software development	2	1.5 - Systems and Quality	5	1	Recognize how systems theory can be used to explain project dynamics, and success and failure in a project management context
Discuss pros and cons of LOC and function point metrics	2	10.105 – IS Development, Project Planning	8	3	Use the function point technique to make software project estimates
Conduct and describe software reviews and Formal Technical Reviews	3	10.106 – IS Development, Project Management	7	2	Describe inspection and review meetings and the pros and cons of review meetings as forms of nonexecution-based testing
Describe what is involved in software planning including the use of tools such as the COCOMO model	1	10.107 – IS Development, Project Management	8	3	Use a project management software tool to effectively solve generic project management problems
Draw up Gantt charts for specific projects	3	10.108 – IS Development, Project Management Tools	8	4	Use the critical path method, PERT, and Gantt charts for estimating, evaluating, and scheduling the interrelating activities of an IS project
Describe and discuss the importance of coding standards	1	10.122 - IS Policies and Standards	7	1	Explain the benefits of following a standard software process, i.e. the CMM
	-	00.3 - Problem Solving, Small IS	2	3	Use electronic spreadsheet software to solve project management to solve quantitative project management problems, such as project comparison and selection and estimation of activity durations, costs, and effort.
	-	01.11 - IS Careers	5	2	Discuss the skills and abilities of project managers, the role of professionalism in project management; and distinguish between the role of functional and project managers in different project organizations (functional, project, matrix, and hybrids)
	-	03.17 - IS as a Strategic Component	3	2	Describe reasons for IS success and failure from Standish Group studies

CIS-231 Obj	CIS-231 Lvl	Learning Unit	Coverage	ITE-475 Lvl	ITE-475 Obj
	-	03.26 - IS Planning	3	3	Recognize that an IS project plan should be linked to organizational planning, and that the lack of such is a common cause of IS project failure (Standish Group reasons for IS project success and failure); apply failure approach in project management case study
	-	05.56 - Problem Solving, IS Applications, Sub-Structures	7	3	Use a spreadsheet for project cost-benefit analysis and other financial models for evaluating projects.
	-	07.72 - IS Analysis and Design Tasks	8	2	Define and distinguish between key information systems analysis and design activities relevant to IT projects and the different types of IT projects
	-	07.84 - Systems and Quality Metrics/Assessment	7	1	Explain the importance and benefits of testing throughout the life cycle, including aspects of client satisfaction in the definitions of success and quality
	-	08.86 - Interpersonal, Synergistic Solutions	7	3	Evaluate group work tasks by the degree of task interdependence, and devise management tactics accordingly for all types of projects
	-	09.112 - Personal, Proactivity, Principled Action	6	2	Describe issues with keeping a project under control, including tradeoffs among the scope, cost, and time dimensions of a project
	-	09.114 - Interpersonal, Goals, Mission, Alignment	8	1	List and describe various project control tactics to keep team member activities in alignment with project goals
	-	09.115 - IS Responsibility to Sell Designs to Management	5	3	Evaluate potential IS projects according to a multi-criteria technique, the SMART method
	-	1.9 - IT and Attaining Objectives	2	1	Recognize that IS projects should be linked to organizational strategies, goals, and objectives
	-	10.125 - IS Implementation and Outsourcing	3	1	Describe the key reasons driving the decision of whether nor not to outsource (develop and possibly operate) information systems, including large portions of the IS function
	-	10.126 - Personal, Time and Relationship Management	5	2	Describe time management paradoxes such as Brooks Law and Parkinson-s Law, and how managing time in an IS project must be balanced off against managing scope, quality, and cost.

Table 1 - Learning Unit Comparison Report

RESULTS

At a high level, the report showed that a significant number of learning units were covered by both the Software Engineering Principles course (27) and the IT Project Management course (34), and that of the learning units covered, the courses overlapped in covering 20 of the LUs. Software engineering principles covered seven learning units that were not covered by

IT Project Management, while 14 other learning units were covered by IT Project Management but not by Software Engineering Principles.

The faculty then considered each of the three sections of the report, one at a time, starting with the seven learning units addressed only by CIS-231, Software Engineering Principles. By looking at the first section of the report, one can see that its local objectives on the topics of Total Quality Management (TQM) and decision trees covered learning units touched by few other courses in the IS curriculum. The most in-depth area that would be missed if CIS-231 were replaced would be the learning unit on workflow diagramming. Although five other courses cover that learning unit, the faculty determined that only one other IS course covers this learning unit with in-depth UML diagramming, as done in CIS-231.

Next, the faculty considered the 20 overlapping learning units. Most notably, the coverage of software quality metrics occurs only in the two courses being compared. The other notable issue is that CIS-231 covers some of the learning units dealing with quality and testing in more depth than ITE-475. Almost all of the rest of the learning units in the overlapping group are covered with a similar local objective and at an identical level of depth. Almost all of these overlapping learning units are covered elsewhere in the curriculum as well, with the exception of the software quality metrics unit.

Finally, the faculty considered the third group of learning units, those covered by IT Project Management but not by Software Engineering Principles. Of striking note in this group are a set of IT management learning units that are covered in few other places in the curriculum including units on IT and attaining objectives, IS as a strategic component, IS planning, and IS implementation and outsourcing. Additionally, the project management course covers a small-IS application problem-solving learning unit covered in few other places by providing a spreadsheet analysis problem. This finding confirmed one faculty member's observation that student's spreadsheet skills, in general, appeared to be deficient.

The discussion then moved to suggesting alternatives. Some of the alternatives centered on identifying the topics that had to be covered in some depth by other courses, if CIS-231 were to be replaced by ITE-475. Other alternatives included the suggestion of different courses to replace, based on the value of what Software Engineering Principles offered. A final decision was not reached in the meeting, but there seemed to be a perception of value in the reporting process, and it was realized that any decision proposed could be analyzed and evaluated by a similar impact report.

REACTIONS FROM FACULTY PARTICIPANTS

The use of the mapping data seemed to change the nature of the curriculum debate. In some cases, the data reinforced existing opinions, and in other cases, the data changed opinions. The faculty member that taught object-oriented systems analysis and design remarked that he would be surprised if students would be very successful at learning UML diagramming at an in-depth level in the sophomore software engineering course. A consensus seemed to emerge among the meeting the OO SAD course, taught in the junior year, was a more appropriate place for IS students to learn about UML diagramming techniques, and that we should keep that course as is. This also diffused a movement by two other faculty who were going to argue that we recombine the two-course SAD sequence into one. It was argued instead, that we leave the OO SAD course alone, as it provided junior-year coverage of an important skill area.

Another faculty noted just how much overlap existed among courses, a fact brought out by the Coverage column of the report. Some learning units were covered in as many as 8 to 10 courses in the curriculum. The data reinforced the beliefs of this faculty member that there was perhaps too much overlap in the curriculum, and he was hopeful that more data would help identify major course overlaps and support efforts to streamline the curriculum, if the need arose.

A senior faculty member in the meeting was struck by the richness of IS management material covered in the IT project management course. He said that the richness of this management material was not reflected in the software engineering course, which was more about the technical detail of applications development, in his opinion. Furthermore, the IS management material, which was more fundamentally important for IS students, was not covered elsewhere in the curriculum, the faculty member noted after reviewing the report. This faculty member moved that we replace software engineering with IT project management. However, the faculty held off on making a final decision.

CONCLUSIONS

The process added objectivity to an otherwise very subjective process. The fundamental question of the proposed curriculum change was "what would the students gain and lose if we swapped the two courses?" Usually, such a question could spark arguments of turf battles and philosophical differences, and it did. The addition of the mapping data, however, provided a more objective means of answering the question, and a means of analyzing the impact of the proposed change. The objectivity was obtained by following a process that emphasized the use of the IS2002 model curriculum as a standard for expressing the content of each course, then comparing the two courses in terms of the standard.

Comparing the relative educational value of two courses is but one such use of IS2002 for mapping one's curriculum. Future work in this area will explore other course and curriculum-based uses of IS2002. In addition to tabular and numerical curriculum mapping data, the use of graphical reports is possible and is a work-in-progress for this group. Currently, software exists that is free of charge for mapping one's IS curriculum. The Windows-based application program is available on request from the authors.

Besides making faculty aware of free curriculum development software, how else is this case study useful for faculty at other institutions? We think that it provides an analytical framework for evaluating the impact of a curriculum decision by linking up course objectives against one another, by expressing objectives in terms of the model curriculum, and by counting the number of overlaps in coverage and reporting depth levels of coverage. This framework is useful for providing insights into the course comparison debate and decision.

REFERENCES

1. ABET CAC (2002) 2003-2004 Criteria for Accrediting Computing Programs, Computing Accreditation Commission, Accreditation Board for Engineering and Technology, Inc., Baltimore, MD, http://www.abet.org/criteria.html, (accessed on February 16, 2004).
2. Daigle, R., Longenecker, H., Landry, J. Pardue, J. (2004) Using the IS 2002 Model Curriculum for Mapping an IS Curriculum, *Information Systems Education Journal*, 2, 1, URL: http://isedj.org/2/1/
3. Gorgone, J., Davis, G., Valacich, J., Topi, H., Feinstein, D. Longenecker, H. (2002) IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems (DRAFT)," ACM, New York, NY, AIS, and AITP (formerly DPMA), Park Ridge, IL.