Computer Science Curriculum and Industry Certification: Integrating Education with Practice

Dr. Ahmed El-abbassy

El-shorouk Academy, Institute of Computers and Information Technology
P.O Box, El-shorouk, Cairo, Egypt

ahmed_elabbassy@yahoo.com

Abstract: *The curriculum of a typical computer science (CS) department gives students a well-rounded, broad base with which they move into industry. However without specific product skills many employers may be reluctant to hire CS graduates. Employers often complain that job applicants and new hires are not well prepared to work for them. An emerging need for industry is the qualification for industry-based certificates. Credentialing for specific products has become predominant. Academics typically resist the demands of the industry, in part because some of them are for specific software tools, design methods, or programming languages that might fade away with time. Under market pressure, industry-based certifications are rapidly considered being complementary to and may integrate with academic degrees.*

In this paper integration issues are discussed and main integration models are presented. Finally we present and discuss a case study to integrate industry-based certifications with a CS curriculum.

Keywords: Accreditation, Computer science education, CS Curriculum, Industry certification.

1. Introduction

The research found that IT certifications are complementary to basic academic degrees, as degrees set a path for a career but are insufficient to furnish specific skills [1,2,3,4,5].

Both certifications and academic qualifications are needed for the IT industry.

The following table highlights the complementary nature of the two approaches [6,7].

Table 1 : The Complementary Nature of the Two Approaches

Academic degrees	Industry Certifications
- give IT students a well-rounded academic base to enable them to move into industry	- add competence to use specific products - establish a standard of competency in specific areas and job roles - extend credibility into areas not covered by academic degree
- deepen the learning across a range of disciplines	- quicker to attain, thereby better keeping up with market demand
- teach relevant theory and conceptual underpinnings but use only one or two technical examples due to time constraints	- provide learning around a particular technical implementation
- theory and concepts are less prone to change	- technical implementations are always changing
- don't give real-world experience	- more task and career specific

From pedagogical point of view, there are three paradigms of academic curriculum development [8], as illustrated in Figure (1).

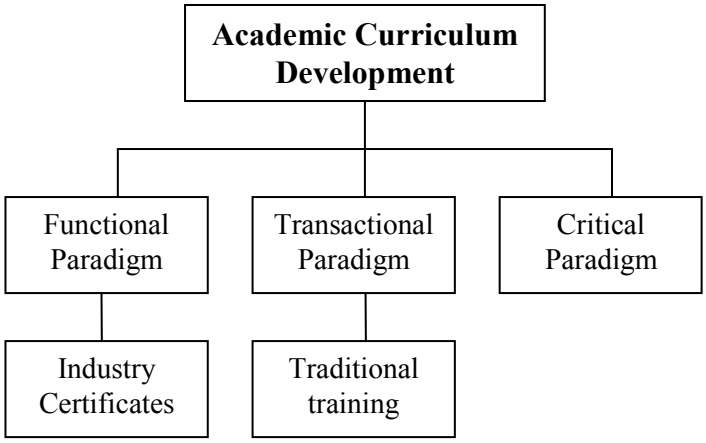

Figure 1: Paradigms of Academic Curriculum Development

- Functional paradigm, the functional paradigm is set in the present to match what industry or the person seeking a job needs now.
- Transactional paradigm is based on the needs of the learner group particular to a course of study and may involve negotiated objectives.

- Critical paradigm is based on future needs and the development of critical thinkers.

While an academic institution may adopt any or all of these paradigms within their curriculum, academic degrees traditionally conform to transactional paradigm, and industry certifications clearly conform to the functional paradigm.

With the advent of cyber-learning models the ability to certify may be the only true competitive advantage of the university [9].

2. Discussion of CS curricula.

Many undergraduate programs in CS adhere to some extent to the model curricula developed by the Association for Computing Machinery, the Association for Information Systems and the IEEE Computer Society [2]. A new report for the CS curriculum was published in 2008 [10]. It defines 13 broad areas of core knowledge and indicates the minimum time that should be spent on those areas.

CC2008 Knowledge Areas

1. DS. Discrete Structures (43 core hours)
2. PF. Programming Fundamentals (47 core hours)
3. AL. Algorithms and Complexity (31 core hours)
4. AR. Architecture and Organization (36 core hours)
5. OS. Operating Systems (18 core hours)
6. NC. Net-Centric Computing (15 core hours)
7. PL. Programming Languages (21 core hours)
8. HC. Human-Computer Interaction (8 core hours)
9. GV. Graphics and Visual Computing (3 core hours)
10. IS. Intelligent Systems (10 core hours)
11. IM. Information Management (11 core hours)
12. SP. Social and Professional Issues (16 core hours)
13. SE. Software Engineering (31 core hours)
14. CN. Computational Science (no core hours)

The areas that are directly relevant to the software profession [11] include Programming Fundamentals (16%), Net-Centric Computing (5%), Programming Languages (7%), Information Management (4%), Software Engineering (10%), for a total of 42%.

As expected, this body of core knowledge includes theoretical areas, such as Algorithms and Complexity (10%), Discrete Structures (15%), Artificial Intelligence (3.5%), as well as areas related to hardware and computer architecture (18%).

At the same time, it remains important to recognize that this core does not constitute a complete undergraduate curriculum, but must be supplemented by additional courses (electives) that may vary according to institution, degree program, or individual student.

The BS curricula of specific CS departments are based on a combination of many factors:

- Faculty preferences,
- Influence of graduate curricula,
- Traditional specializations,
- Resources,
- Requirements imposed by the university, and
- Other scheduling constraints.

The flexibility provided by the ACM model clearly permits to envisage integration of IT Certification with CS curricula.

While the Joint Task Force for Computing Curricula [2] believes that academic integrity and ethics become an issue when integrating industry certification programmes with academic curricula, others openly advocate such integration [4,12,13,14].

The advantages of integration of industry certification with academic curricula significantly outweigh drawbacks [15].

Under market pressure many academic institutions have integrated, or are in the process of integrating, industry certification into their curricula.

Due to the complementary nature of industry certification and academic qualifications, it is in the interests of the academic institution to give serious consideration to such integration if given certifications are also complementary to the objectives of the curriculum.

The CS curriculum should be upgraded regularly through surveys of skills needs, ongoing dialogue with information technology business leaders, technical professionals, and monitoring of the business environment and technological trends.

3. Integration models

There are a number of Institutes of Technology and Polytechnics who have positioned themselves to offer industry certified qualifications [15].

Two main models were proposed [16] which are:

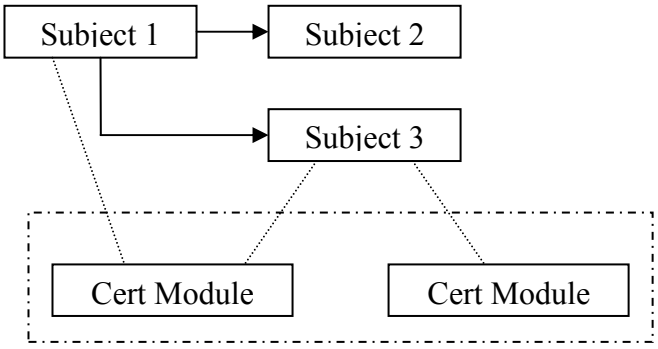

Figure 2 Course Mapping Integration Model

- Course Mapping (figure 2) This model is based on Linking of subject content with accredited programs. The certification training is embedded within the academic courses. Students upon completion of subjects receive industry accreditation that they can use to further their career prospects.

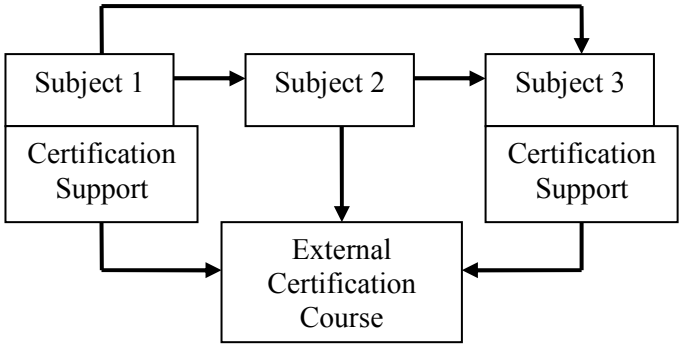

Figure 3 Curriculum Inclusive Integration Model

- Curriculum Inclusive (figure 3)
  The Institute will establish an alliance with a private provider of the industry certificate training. The provider supplies courses at a reduced cost to the students.
  The international certification exam cost is unchanged.
  Student participation is optional and will not affect their assessment in the subject.

Based on these main models, academic institutes may follow a hybrid approach. It is clear that there is no standard model for integration. Integration should be studied case by case as many factors are involved.

4. Case study

This case study is developed by the Department of Computer Science at the Institute of Computers and Information Technology - El-Shorouk.

Our CS curriculum is consistent with recommendations outlined by the Association for Computing Machinery (ACM) and emphasizes laboratory experience as a major component of courses.

Integrating industry certification possibilities into our program could not be done at the expense of educational goals, or accreditation. Instead, the approach was to identify individual courses and/or sequences of courses for which 'minimal' modification could lead to certification.

One of the goals of the Department of Computer Science is to prepare students to specialize in the development of application software. Software development is a major employment fields open to qualified programmers. It is also excellent preparation for advancement in computer systems analysis, design and management.

To integrate certificates with the curriculum, the department of computer science has adopted a hybrid approach:

- Adapt the academic courses to be more aligned with certificates.
- Establish an alliance with a private provider of the industry certificate training. The provider supplies courses at a reduced cost to the students.
- Student participation is optional.

The adaptation process consists of the steps illustrated in Figure (4) :

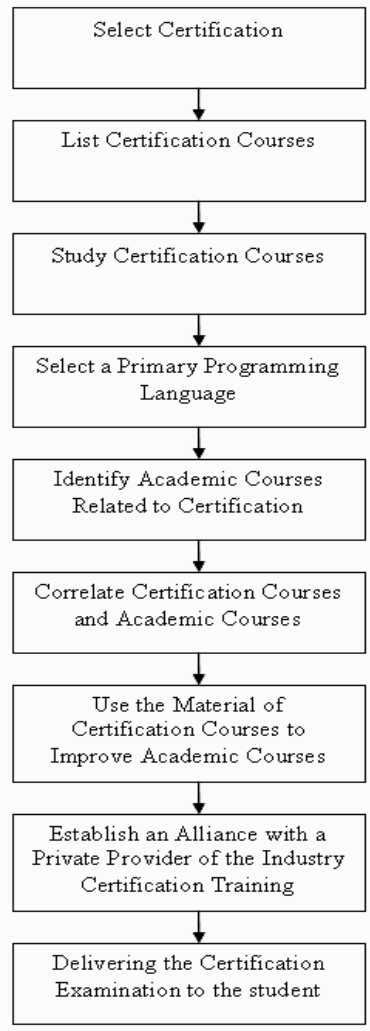

Figure 4: Aligning Academic Courses with Industry Certification

a. Select Certification

The selection of certificates depends mainly on market needs and aims of the degree.

The department has surveyed the market to select the international credentials that best suit the students, and takes the responsibility to adapt the academic courses in order to prepare the students for these credentials.

From this survey, the department selected the MCAD & MCSD Microsoft credentials (and now their successors MCTS & MCPD). The Microsoft Certified

Solution Developer (MCSD) covers various aspects of computer programming and software development.

The MCAD & MCSD Microsoft credentials are the best suiting the graduates of the academy due to the following:

- Many of the world's businesses run Microsoft Platforms and use Microsoft Applications.
- The industry recognizes the value of a hands-on qualification that prepares students for the real commercial world.
- Because certification is widely recognized both here in Egypt, in the middle east and all over the world

b. List the certification courses.

The certification courses are listed in the following table.

Table 2: The Certification Courses.

No.	**MCAD & MCSD courses**	
1	Course 2310	Developing Microsoft ASP.net Web Applications Using Visual Studio.net
2	Course 2524	Developing XML web Services Using Microsoft ASP.net
3	Course 2663	Programming With XML in the Microsoft.Net Framework
4	Course 2124	Programming with Microsoft C#
5	Course 2555	Developing Microsoft.net Applications for Windows (Visual C#.net)
6	Course 2389	Programming ADO.net
7	Course 2557	Building COM+ Applications Using Microsoft.net Enterprise Services
8	Course 2710	Analyzing Requirements and Defining Microsoft.NET Solution Architectures
9	Course 2071	Querying Microsoft SQL Server2000 With Transact- SQL
10	Course 2073	Programming a Microsoft SQL Server 2000 Database

c. Study in detail these courses from different perspectives:

- Course objectives & contents
- Prerequisites
- Achievement at course completion
- Required technical tools

- Audience - Certification exams

d. Select a primary programming language

Programming languages provide a striking example of conflict between academia and industry: languages used in industry and not offered in many curricula; and languages offered in curricula, but not used in industry. On the other hand, computer science departments, for their first programming languages, have taught languages that are not used in industry.

It is better for students to learn programming languages that they will use at work.

For this point the department decided to adopt C# as a primary programming language.

C# has become a main language for developing applications as well as systems software; it is also a well-designed object-oriented language.

e. Identify academic courses related to the certification.

In this step, academic courses related to certification are listed in the following table.

Table 3: Academic Courses Related to Certification.

MCAD & MCSD courses		**Academic Course**	
		Course Code	Course Name
Course 2310	Developing Microsoft ASP.net Web Applications Using Visual Studio.net	4102 4103	Network programming Software Engineering(2)
Course 2640	Upgrading Web Development Skills From ASP to Microsoft ASP.net	4102 4103	Network programming Software Engineering(2)
Course 2524	Developing XML web Services Using Microsoft ASP.net		
Course 2663	Programming With XML in the Microsoft.Net Framework	4102	Network programming
Course 2124	Programming with Microsoft C#	1201 2103 2102	Structured Programming Object Oriented Programming Data Structures

Course 2555	Developing Microsoft.net Applications for Windows (Visual C#.net)	2103 3104 3203	Object Oriented Programming Human Computer Interfaces Software engineering(1)
Course 2389	Programming ADO.net	3203	Software engineering(1)
Course 2557	Building COM+ Applications Using Microsoft.net Enterprise Services		
Course 2710	Analyzing Requirements and Defining Microsoft.NET Solution Architectures	2201 3103 3203 3104	Systems Analysis Systems Design Software Engineering(1) Human Computer Interfaces
Course 2071	Querying Microsoft SQL Server2000 With Transact- SQL	3101 3203	Data Base Systems Software Engineering(1)
Course 2073	Programming a Microsoft SQL Server 2000 Database	3101 3203	Data Base Systems Software Engineering(1)

f- Correlate the MCAD & MCSD requirements and the Academic courses taught at the Department.

The topics of the certification courses are correlated to the academic courses as illustrated in the following tables.

Table 4: Cortication Courses versus Academic

(1) Course 2310: Developing Microsoft ASP.NET Web Applications.

No	MCAD & MCSD Course #2310	Academic Courses	
		4102	4103
1	Overview of the Microsoft .NET Framework	√	√
2	Using Microsoft Visual Studio .NET	√	√
3	Creating an ASP.NET Web Application Project Using Visual Studio .NET	√	
4	Using Microsoft .NET-Based Languages	√	√
5	Create a new project in Visual Studio .NET for a c# class	√	√
6	Creating a Microsoft ASP.NET Web Form	√	
7	- Creating the default. aspx Web Form - Creating the life.aspx Web Form	√	
8	Adding Code to a Microsoft ASP.NET Web Form	√	
9	Creating a **Page_Load** Event Procedure		
10	Tracing in Microsoft ASP.NET Web Applications	√	
11	- Using Trace Statements - Tracing into a Component	√	
12	Validating User Input	√	
13	Using Required Field Validator, Validation Summary, Compare Validator, Regular Expression Validator Controls	√	
14	Creating User Controls	√	
15	Accessing Relational Data Using Microsoft Visual Studio .NET		√
16	Connecting to a Database Paging and Selection in a Data Grid Control		√
17	Accessing Data with Microsoft ADO.NET		√
18	Using a Sql Data Reader Viewing Data from the Database		√
19	Calling Stored Procedures with Microsoft ADO.NET		√
20	Reading and Writing XML Data	√	
21	Consuming and Creating XML Web Services		
22	Managing State	√	
23	Using Cookies	√	
24	Helping to Protect a Microsoft ASP.NET Web Application		√
25	Securing Your Web Site Using Forms-Based Authentication		√

(√): A topic covered by the academic course.

(2) Course 2124: Programming with Microsoft C# .

No	MCAD & MCSD Course #2124	Academic Courses		
		1201	2103	2102
1	Overview of the Microsoft .NET Platform	√		
2	Overview of C#	√		
3	Using Value-Type Variables	√		√
4	Statements and Exceptions	√		
5	Methods and Parameters	√		
6	Arrays	√		√
7	Essentials of Object-Oriented Programming		√	
8	Using Reference-Type Variables		√	√
9	Creating and Destroying Objects		√	√
10	Inheritance in C#		√	√
11	Aggregation, Namespaces, and Advanced Scope		√	√
12	Operators and Events		√	
13	Properties and Indexers		√	
14	Attributes		√	√

(3) Course 2389: Programming with ADO .NET

No	MCAD & MCSD Course #2389	Academic Courses
		3203
1	Data-Centric Applications and ADO.NET	√
2	Performing Connected Database Operations & Connecting to Data Sources	√
3	Building Data Sets: - Building, Binding, Opening, and Saving Data Sets - Manipulating Data Sets	√
4	Reading and Writing XML with ADO.NET	
5	Building Data Sets from Existing Data Sources& Retrieving Data into a Disconnected Application	√
6	Building and Consuming a Web Service That Uses ADO.NET & Troubleshooting an ADO.NET Application	√

(4) Course 2555: Developing .NET Applications for Windows Visual C# .NET .

No	MCAD & MCSD Course #2555	Academic Courses		
		2103	3104	4103
1	Introducing Windows Forms	√	√	
2	Working with Controls	√	√	
3	Building Controls	√	√	
4	Using Data in Windows Forms Applications	√	√	√
5	Interoperating with Managed Objects	√	√	√
6	Printing From a Windows Forms Application	√	√	
7	Asynchronous Programming	√		
8	Enhancing the Usability of Applications		√	
9	Deploying Windows Forms Applications		√	√
10	Securing Windows Forms Applications		√	√

(5) Course 2710: Analyzing requirements & defining .NET Solution Architectures.

No	MCAD & MCSD Course #2710	Academic Courses			
		2201	3103	3203	3104
1	Introduction to Designing Business Solutions		√	√	
2	Gathering and Analyzing Information	√		√	
3	Envisioning the Solution & Developing a Vision/Scope Document	√	√	√	
4	Creating the Conceptual Design	√	√	√	
5	Analyzing Requirements	√			
6	Creating the Logical Design		√	√	
7	Creating the Physical Design		√		
8	Designing the Presentation Layer		√		√
9	Creating the User Interface		√		√
10	Designing the Data Layer		√	√	
11	Creating a Data Schema		√	√	
12	Designing Security Specifications		√		
13	Completing the Planning Phase			√	
14	Prioritizing Bugs			√	

(6) Course 2663: Programming with XML in the Microsoft .NET Framework .

No	MCAD & MCSD Course #2524	Academic Courses
		4102
1	Introduction to XML in the .NET Framework.	√
2	Parsing XML	√
3	Validating XML.	√
4	Writing XML	√
5	Querying XML	√
6	Manipulating Cached XML	
7	Transforming XML	
8	Serializing Objects as XML	

(7) Course 2071: Querying Microsoft SQL Server 2000 with Transact-SQL .

No	MCAD & MCSD Course #2071	Academic Courses	
		3101	3203
1	Introduction To Transact-SQL	√	
2	Using Transact-SQL	√	
3	Retrieving Data	√	
4	Grouping And Summarizing Data	√	
5	Joining Multiple Tables	√	
6	Working With Sub-queries	√	
7	Modifying Data	√	
8	Querying Full-Text Indexes	√	
9	Introduction To Programming Objects		√

(8) Course 2073: Programming a Microsoft SQL Server 2000 Database.

No	MCAD & MCSD Course #2073	Academic Courses	
		3101	3203
1	Overview of Programming SQL Server	√	
3	Creating and Managing Databases	√	
5	Creating Data Types and Tables	√	
7	Implementing Data Integrity	√	
9	Planning Indexes & Determining the Indexes of a Table		√
11	Creating and Maintaining Indexes & Viewing Index Statistics		√
13	Implementing Views		√
15	Implementing Stored Procedures		√
17	Implementing User-Defined Functions		√
19	Implementing Triggers		√
21	Using Distributed Data		
22	Optimizing Query Performance		
23	Analyzing Queries		
24	Managing Transactions and Locks		

g. For the covered topics, the material of the certification course is used to improve the academic course contents.

h. Establish an alliance with a private provider of the industry certificate training. The provider supplies courses at a reduced cost to the students based on the percent of coverage of certification course topics by the academic courses.

i. Delivering the Certification Examination to the Student.

An exam location should be assigned to students. We found that offering them in-house exam is much better than sending them off to other locations /institutions. We are currently able to offer the certification examinations that we have incorporated into our program in-house.

The first results from applying this approach are promising.

80 students entered the following MCTS exams:

Exams	Description
Exam 70–526:	Microsoft .NET Framework 2.0 - Windows-Based Client Development
Exam 70–536:	Microsoft .NET Framework 2.0 – Application Development Foundation

The percent of students passed the exams from the first round reaches 70%.

5. Conclusion

Sound education must include a combination of all the three pedagogical paradigms: functional, transactional, and critical.

Under the job market pressure, universities are asked to satisfy better the needs of the software industry.

Due to the complementary nature of industry certification and academic qualifications , and the inherent flexibility provided by The ACM model of CS curriculum , it is in the interests of individual academics and the academic institutions to give serious consideration to integration of industrial certifications with the academic curriculum if given certifications are also complementary to the objectives of the curriculum.

There are varsities of models and case studies that illustrate how integration may be successfully implemented. However a standard model for integration is not recommended. Integration should be studied case by case as many factors are involved.

It is also envisaged to adopt a national strategy for this predominant topic.This topic will remain under investigation and research.

References

[1] Cosgrove, S. "Academic Qualification & Industry Certification: Integrating Education With Practice". Paper Presented At The 17th NACCQ, 6th - 9th July 2004.

[2] Shackelford, R., Et Al. "Computing Curricula 2005: The Overview Report Including The Guide To Undergraduate Degree Programs In Computing". Joint Task Force ACM, AIS, IEEE-CS**:** 46, 2005.

[3] Cegielski, C. "Who Values Technology Certification?". Communications Of The ACM, (47:10), 2004, Pp. 103-105.

[4] Adelman, C. A "Parallel Postsecondary Universe: The Certification System In Information Technology". Office Of Educational Research And Improvement, U.S. Department Of Education. 2000. Retrieved From The ACM Digital Library.

[5] "On The Right Track With Professional Certification". Computimes Malaysia, , February 8, P. 1,1999.

[6] Leo Hitchcock, "Industry Certification And Academic Degrees: Complementary, Or Poles Apart?". SIGMIS-CPR'07, April 19–21, 2007, Copyright 2007 ACM 978-1-59593-641-7/07/0004

[7] Montante, R., & Khan, Z. "Specialized Certification Programs In Computer Science". Proceedings Of The Thirty-Second SIGCSE Technical Symposium On Computer Science Education SIGCSE '01, February 21-25, 2001, ACM SIGCSE Bulletin (33:1), 2001, Pp.371-375

[8] Melrose, M. "Got A Philosophical Match? Does It Matter? " In O. Zuber-Skerrit (Ed.), New Directions In Action Research. Falmer Press, London: 1996, Pp. 49-65.

[9] Clear, T. "Online Education – But Is It Education? ". SIGSCE Bulletin,1999, Pp. 17-18.

[10] ACM And IEE-CS. "Computer Science Curriculum 2008: An Interim Revision Of CS 2001". December 2008

[11] Alexis Koster, "Are Academic Programs Adequate For The Software Profession?", Eabr & Tlc Conferences Pro, Prague, Czech Republic,2009.

[12] Brookshire, R. "Information Technology Certification: Is This Your Mission? ". Information Technology, Learning, And Performance, 2004, Pp. 1-2.

[13] J. Barrie Thompson, "Software Engineering Practice And Education An International View",Seese'08, May 13, 2008, Leipzig, Germany. Copyright 2008 Acm

[14] Ursula Fuller, Bob Keim, "Assessing Students' Practice Of Professional Values", Iticse'08, June 30–July 2, 2008, Madrid, Spain.Copyright 2008 ACM

[15] Roberton, G., & Corbett, E. "Creating Intriguing Synergies With Course Offerings: Integrating Industry Qualifications -An ITP Perspective". Paper Presented At The 17th NACCQ 6-9 July 2004.

[16] Jovanovic, R., Bentley, J., Stein, A., & Nikakis, C."Implementing Industry Certification In An IS Curriculum: An Australian Experience". Information Systems Education Journal, (4:59), August 2006.