European Journal of Scientific Research
ISSN 1450-216X Vol.48 No.3 (2011), pp.352-360

http://www.eurojournals.com/ejsr.htm

Quality Criteria's of Computing Disciplines

Feras Mmohamad Al-Azzah
Alzaytoonah University of Jordan, The Faculty of Science and IT
Department of Computer Information System
E-mail: drferas@alzaytoonah.edu.jo

Abdelfatah Aref Yahya
Alzaytoonah University of Jordan Dean, The Faculty of Science and IT
Department of Computer Science
E-mail: drtamimi@alzaytoonah.edu.jo

Abstract

This paper analyze study plans of the five approved disciplines in the computing space of international universities and provides a mechanism and a set of quality standards that you must follow when creating or defining a study plan of and academic content of any discipline belongs to the computing space to improve the quality of higher education through links between the expected outputs and the proposed study plan.

Keywords: Quality Criteria's, Educational process, IT, Software Engineering, Computer Science, Information Systems, Curriculum, Computing Disciplines

Introduction

Computing means any goal-oriented activity requiring, benefiting from, or creating computers. Thus, computing includes designing and building hardware and software systems for a wide range of purposes; processing, structuring, and managing various kinds of information; doing scientific studies using computers; making computer systems behave intelligently; creating and using communications and entertainment media; finding and gathering information relevant to any particular purpose.

Because society needs people to do computing well, we must think of computing not only as a profession but also as a discipline. Computing is not just a single discipline but is a family of disciplines that attract quality students and prepare them to be capable and responsible professionals, scientists, and engineers.

Regarding to Computing Curricula Report (CC2005) as a cooperative project of The Association for Computing Machinery (ACM), The Association for Information Systems (AIS), The Computer Society (IEEE-CS), the growing family of computing-related disciplines contends five disciplines: computer engineering, computer science, information systems, information technology and software engineering.

Brief description of the area of each discipline reflects the need for them, the position in the computing space, identifies the fields and areas of knowledge and skill that must be provided in person who chooses to study any of these disciplines.

Brief Description of the Computing Disciplines

Computer Engineering

Computer engineering is concerned with the design and construction of computers and computer-based systems. It involves the study of hardware, software, communications, and the interaction among them. Its curriculum focuses on the theories, principles, and practices of traditional electrical engineering and mathematics and applies them to the problems of designing computers and computer-based devices.

Computer Science

Computer science spans a wide range, from its theoretical and algorithmic foundations to cutting-edge developments in robotics, computer vision, intelligent systems, bioinformatics, and other exciting areas. Computer science offers a comprehensive foundation that permits graduates to adapt to new technologies and new ideas.

Information Systems

Information systems specialists focus on integrating information technology solutions and business processes to meet the information needs of businesses and other enterprises. This discipline's perspective on information technology emphasizes information, and views technology as an instrument for generating, processing, and distributing information.

Information Technology

IT refers to undergraduate degree programs that prepare students to meet the computer technology needs of business, government, healthcare, schools, and other kinds of organizations. In some nations, other names are used for such degree programs.

Software Engineering

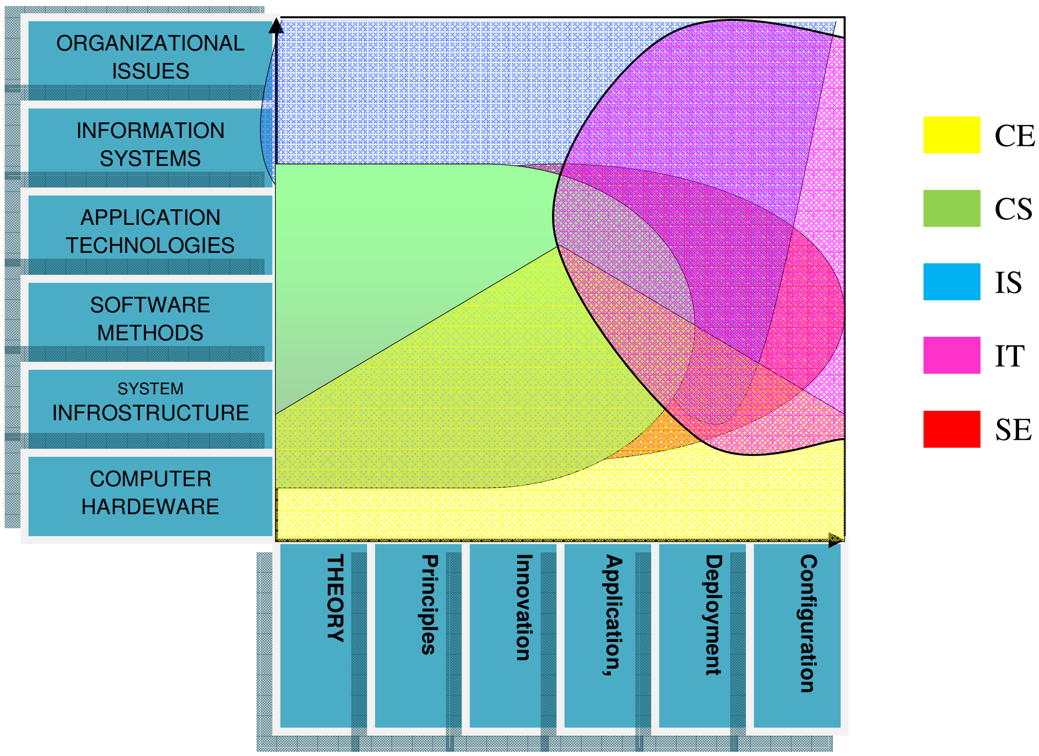
Software engineering is the discipline of developing and maintaining software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. SE seeks to integrate the principles of mathematics and computer science with the engineering practices developed for tangible, physical artifacts.

To illustrate the commonalities and differences among computing disciplines, we have created graphic characterizations of them. They suggest how each discipline occupies the problem space of computing as shown in Figure1. The horizontal range runs from Theory, Principles, and Innovation on the left, to Application, Deployment, and Configuration on the right. The vertical range runs from Computer Hardware and Architecture at the bottom, to Organizational Issues and Information Systems at the top

Problem Scope

The most important element identifying the outputs of any discipline is a study plan for academic content of the materials to be studied, and although there are many studies and reports that dealt with the specialty of this perspective, but absent from this research, reports, concepts and principles of quality for an objective assessment specialties in the space of computing, where they adopted these studies on the comparison between what is available in the study plans and what is actually required to create a new discipline.

Figure 1: The problem space of computing



The most important of these reports is done by the (ACM) in collaboration with associations and scientific bodies in the area of computing which is based on two tables for comparison between different disciplines in the computing space, namely:

- Table.1 that summarizes the *inputs* provided to students by degree programs,
- Table.2 focuses on *outputs*, summarizing the relative capability expectations of graduates.

Table.1 represents the consensus of judgment reached by the CC2005. The report formulated this consensus from an examination of the discipline-specific body of knowledge found in the most recent curriculum volume for each of the computing disciplines.

(m) represents the minimum called for by the curriculum guidelines, and (M) max represents the greatest emphasis one might expect in the typical case of a student who chooses to undertake optional work in that area or who graduates from an institution that requires its students to achieve mastery beyond that required by the curriculum reports. Because the difference between the min and max values can be large, programs with the same degree name may differ substantially because of the local choices made in determining their requirements. Both min and max values refer to what can be reasonably expected in the general case. For any individual student or degree program, the min value might be as low as zero and the max value might be as high as five, regardless of prevailing curricular standards.

Table 1: Comparative weight of computing and non-computing topics across the five kinds of degree programs

N	*Knowledge Area*	*CE*		*CS*		*IS*		*IT*		*SE*	
		m	**M**	**m**	**M**	**m**	**M**	**m**	**M**	**m**	**M**
1	Programming Fundamentals	4	4	4	5	2	4	2	4	5	5
2	Integrative Programming	0	2	1	3	2	4	3	5	1	3
3	Algorithms and Complexity	2	4	4	5	1	2	1	2	3	4
4	Computer Architecture and Organization	5	5	2	4	1	2	1	2	2	4
5	Operating Systems Principles & Design	2	5	3	5	1	1	1	2	3	4
6	Operating Systems Configuration & Use	2	3	2	4	2	3	3	5	2	4

7	Net Centric Principles and Design	1	3	2	4	1	3	3	4	2	4
8	Net Centric Use and configuration	1	2	2	3	2	4	4	5	2	3
9	Platform technologies	0	1	0	2	1	3	2	4	0	3
10	Theory of Programming Languages	1	2	3	5	0	1	0	1	2	4
11	Human-Computer Interaction	2	5	2	4	2	5	4	5	3	5
12	Graphics and Visualization	1	3	1	5	1	1	0	1	1	3
13	Intelligent Systems (AI)	1	3	2	5	1	1	0	0	0	0
14	Information Management (DB) Theory	1	3	2	5	1	3	1	1	2	5
15	Information Management (DB) Practice	1	2	1	4	4	5	3	4	1	4
16	Scientific computing (Numerical methods)	0	2	0	5	0	0	0	0	0	0
17	Legal / Professional / Ethics / Society	2	5	2	4	2	5	2	4	2	5
18	Information Systems Development	0	2	0	2	5	5	1	3	2	4
19	Analysis of Business Requirements	0	1	0	1	5	5	1	2	1	3
20	E-business	0	0	0	0	4	5	1	2	0	3
21	Analysis of Technical Requirements	2	5	2	4	2	4	3	5	3	5
22	Engineering Foundations for SW	1	2	1	2	1	1	0	0	2	5
23	Engineering Economics for SW	1	3	0	1	1	2	0	1	2	3
24	Software Modeling and Analysis	1	3	2	3	3	3	1	3	4	5
25	Software Design	2	4	3	5	1	3	1	2	5	5
26	Software Verification and Validation	1	3	1	2	1	2	1	2	4	5
27	Software Evolution (maintenance)	1	3	1	1	1	2	1	2	2	4
28	Software Process	1	1	1	2	1	2	1	1	2	5
29	Software Quality	1	2	1	2	1	2	1	2	2	4
30	Comp Systems Engineering	5	5	1	2	0	0	0	0	2	3
31	Digital logic	5	5	2	3	1	1	1	1	0	3
32	Embedded Systems	2	5	0	3	0	0	0	1	0	4
33	Distributed Systems	3	5	1	3	2	4	1	3	2	4
34	Security issues and principles	2	3	1	4	2	3	1	3	1	3
35	implementation and mgt	1	2	1	3	1	3	3	5	1	3
36	Systems administration	1	2	1	1	1	3	3	5	1	2
37	Management of Info Systems Org.	0	0	0	0	3	5	0	0	0	0
38	Systems integration	1	4	1	2	1	4	4	5	1	4
39	Digital media development	0	2	0	1	1	2	3	5	0	1
40	Technical support	0	1	0	1	1	3	5	5	0	1
1	Organizational Theory	0	0	0	0	1	4	1	2	0	0
2	Decision Theory	0	0	0	0	3	3	0	1	0	0
3	Organizational Behavior	0	0	0	0	3	5	1	2	0	0
4	Organizational Change Management	0	0	0	0	2	2	1	2	0	0
5	General Systems Theory	0	0	0	0	2	2	1	2	0	0
6	Risk Management (Project, safety)	2	4	1	1	2	3	1	4	2	4
7	Project Management	2	4	1	2	3	5	2	3	4	5
8	Business Models	0	0	0	0	4	5	0	0	0	0
9	Functional Business Areas	0	0	0	0	4	5	0	0	0	0
10	Evaluation of Business Performance	0	0	0	0	4	5	0	0	0	0
11	Circuits and Systems	5	5	0	2	0	0	0	1	0	0
12	Electronics	5	5	0	0	0	0	0	1	0	0
13	Digital Signal Processing	3	5	0	2	0	0	0	0	0	2
14	VLSI design	2	5	0	1	0	0	0	0	0	1
15	HW testing and fault tolerance	3	5	0	0	0	0	0	2	0	0
16	Mathematical foundations	4	5	4	5	2	4	2	4	3	5
17	Interpersonal communication	3	4	1	4	3	5	3	4	3	4

Table 2: Relative performance capabilities of computing graduates by discipline

No	*Area Performance Capability*	*CE*	*CS*	*IS*	*IT*	*SE*
	Algorithms					
1	Prove theoretical results	3	5	1	0	3
2	Develop solutions to programming problems	3	5	1	1	3
3	Develop proof-of-concept programs	3	5	3	1	3
4	Determine if faster solutions possible	3	5	1	1	3
	Application programs					

5	Design a word processor program	3	4	1	0	4
6	Use word processor features well	3	3	5	5	3
7	Train and support word processor users	2	2	4	5	2
8	Design a spreadsheet program (e.g., Excel)	3	4	1	0	4
9	Use spreadsheet features well	2	2	5	5	3
10	Train and support spreadsheet users	2	2	4	5	2
	Computer programming					
11	Do small-scale programming	5	5	3	3	5
12	Do large-scale programming	3	4	2	2	5
13	Do systems programming	4	4	1	1	4
14	Develop new software systems	3	4	3	1	5
15	Create safety-critical systems	4	3	0	0	5
16	Manage safety-critical projects	3	2	0	0	5
	Hardware and devices					
17	Design embedded systems	5	1	0	0	1
18	Implement embedded systems	5	2	1	1	3
19	Design computer peripherals	5	1	0	0	1
20	Design complex sensor systems	5	1	0	0	1
21	Design a chip	5	1	0	0	1
22	Program a chip	5	1	0	0	1
23	Design a computer	5	1	0	0	1
	Human-computer interface					
24	Create a software user interface	3	4	4	5	4
25	Produce graphics or game software	2	5	0	0	5
26	Design a human-friendly device	4	2	0	1	3
	Information systems					
27	Define information system requirements	2	2	5	3	4
28	Design information systems	2	3	5	3	3
29	Implement information systems	3	3	4	3	5
30	Train users to use information systems	1	1	4	5	1
31	Maintain and modify information systems	3	3	5	4	3
	Information management (Database)					
32	Design a database mgt system (e.g., Oracle)	2	5	1	0	4
33	Model and design a database	2	2	5	5	2
34	Implement information retrieval software	1	5	3	3	4
35	Select database products	1	3	5	5	3
36	Configure database products	1	2	5	5	2
37	Manage databases	1	2	5	5	2
38	Train and support database users	2	2	5	5	2
	IT resource planning					
39	Develop corporate information plan	0	0	5	3	0
40	Develop computer resource plan	2	2	5	5	2
41	Schedule/budget resource upgrades	2	2	5	5	2
42	Install/upgrade computers	4	3	3	5	3
43	Install/upgrade computer software	3	3	3	5	3
	Intelligent systems					
44	Design auto-reasoning systems	2	4	0	0	2
45	Implement intelligent systems	2	4	0	0	4
	Networking and communications					
46	Design network configuration	3	3	3	4	2
47	Select network components	2	2	4	5	2
48	Install computer network	2	1	3	5	2
49	Manage computer networks	3	3	3	5	3
50	Implement communication software	5	4	1	1	4
51	Manage communication resources	1	0	3	5	0
52	Implement mobile computing system	5	3	0	1	3
53	Manage mobile computing resources	3	2	2	4	2
	Systems Development Through Integration					
54	Manage an organization's web presence	2	2	4	5	2
55	Configure & integrate e-commerce software	2	3	4	5	4
56	Develop multimedia solutions	2	3	4	5	3

57	Configure & integrate e-learning systems	1	2	5	5	3
58	Develop business solutions	1	2	5	3	2
59	Evaluate new forms of search engine	2	4	4	4	4

Discipline-specific accreditation is a means of demonstrating that a degree program meets an independent standard of quality, but the meaning of that standard varies. Its rigor is determined by the accrediting body's policies and practices and by any government regulations that might apply. In some cases, accreditation certifies that a degree program has met a minimum quality standard. In other cases, there exist both minimum standards and higher standards.

While discipline-specific accreditation is concerned with program quality, it is important not to reach unwarranted conclusions about the relationship between accreditation and quality. One must be familiar with both the discipline and the national context in order to reach appropriate conclusions.

Meaningful quality improvement requires more than simply calculating student grade point averages and collecting data from end-of-term student satisfaction surveys. If a program has little to point to beyond collection of these basic data, it is reasonable to have doubts about whether there is an adequate focus on systematically improving the quality of both the degree program itself, and of the graduates it produces.

To improve the quality of a study plan for any discipline the researchers adopted a quality standard appropriate to assess the study plans for each discipline based on the required output for that discipline. Allowing an assessment of objective, scientific novelty of each discipline in the computing space based on its mission, and provides indicators and data that indicate the level of quality in the study plan provided by this discipline.

Analysis

In our analysis we used The most familiar measure of dependence between two quantities the correlation coefficient. It is obtained by dividing the covariance of the two variables by the product of their standard deviations. The correlation coefficient $\rho_{X,Y}$ between two variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y is defined as:

$$\rho_{X,Y} = \frac{\mathrm{cov}\,(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y}$$

Where E is the expected value operator and cov means covariance.

The correlation is 1 in the case of an increasing linear relationship, −1 in the case of a decreasing linear relationship, and some value between -1 and 1 in all other cases, indicating the degree of linear dependence between the variables. The closer the coefficient is to either −1 or 1, the stronger the correlation between the variables.

If the variables are independent, correlation coefficient is 0, but the converse is not true because the correlation coefficient detects only linear dependencies between two variables..

If we have a series of n measurements of X and Y written as x_i and y_i where i = 1, 2,..., n, then the *sample correlation coefficient*, can be used to estimate the correlation between X and Y. The sample correlation coefficient is written

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_i - \overline{x})\,(y_i - \overline{y})}{(n-1)s_x s_y}$$

Where x and y are the sample means of X and Y, s_x and s_y are the sample standard deviations of X and Y.

Table 3: Incomes and Outcomes Correlation between Computing Disciplines

	CS		IS		IT		SE	
	In	**Out**	**In**	**Out**	**In**	**Out**	**In**	**Out**
CE	**0.4536**	**0.0451**	**-0.4022**	**-0.6771**	**0.0138**	**-0.5716**	**0.3712**	**0.0867**
CS			**-0.1219**	**-0.1734**	**0.2389**	**-0.2697**	**0.6711**	**0.7482**
IS					**0.2761**	**0.8703**	**0.1052**	**-0.1037**
IT							**0.3248**	**-0.2006**

The analysis of the data shows a set of provisions that call for decisions and appropriate steps to ensure the quality of education:

- Digital comparison demonstrates that the study plan of computer science discipline intersect significantly with computer engineering with more than 43%, while the relationship is very weak (close to 0%) between their output.
- Results of the analysis show that the Study Plan for Information Systems intersect a small percentage which does not exceed 23% with a major information technology, while very strong relationship between the outputs of majors with more than 87%.
- Mathematical analysis shows that the study plan to allocate software engineering, compared with the study plan for the Major Computer Engineering share approximately 37% of subjects, while the output of majors have nothing in common between them.
- Software Engineering allocates shares by 67% with computer science curriculum, and their outputs share by more than 74%.

Results

The analysis result of disciplines currently available in the space of computing and the analysis of study plans of academic content for each discipline are the following:

- Adoption of a list of domains and sub-themes for each area to cover all the actual needs of space based computing as a reference for output to be produced to serve the community and consistent with other scientific and technological developments globally and give him marks for the assessment of weights from 0 to 5.
- Adoption of a lesson plan to allocate computing disciplines include academic content corresponds to the areas and subjects of the existing pre-defined for output of this discipline in the same sequence with determine weights for each subject or academic content.
- Determine quality standard of the proposed study plan for a discipline what is actually required of specialization compared with his plan, compared with the existing reference is greater than eighty per cent of match between.
- Determination of the quality standard for the adoption of message allocate computing specialties of space compared with what is currently available from the disciplines and by reference to the same pre-defined list of output is (<0.4) less than 40% mismatch between the specialization introduced, the disciplines are now available.
- Set quality standard for the adoption of the specialty discipline of computing is interdisciplinary space (> 0.6) more than sixty per cent of the areas of knowledge and themes belong to the existing pre-defined space for output to the disciplines of computing.

  Note that the result of the Fifth dictated by the current and future development of the presence of a third dimension to space computing category is benefiting from specialization to be developed.

Conclusion

The study concludes after studying and analyzing the data and data on the five disciplines adopted in the space of computing to submit proposals defining the criteria for quality minutes to deal with the study plans of disciplines adopted and mechanisms to improve and develop, as well as provide a clear vision and scientific development of new disciplines based on the quality standards that must be adopted.

References

[1] [ANSI/IEEE Standard 729-1983] IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineers, Inc., NY, 1983.

[2] [CC2001] ACM/IEEE-Curriculum 2001 Task Force. Computing Curricula 2001, Computer Science. IEEE Computer Society Press and ACM Press, December 2001. (www.computer.org/curriculum)

[3] [CE2004] IEEE/ACM Joint Task Force on Computing Curricula. Computer Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. IEEE Computer Society Press and ACM Press, December 2004. (http://www.computer.org/curriculum.

[4] [IS2002] ACM/AIS/AITP Joint Task Force on Information Systems Curricula. IS2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery, Association for Information Systems, and Association for Information Technology Professionals, 2002. (http://www.acm.org/education/curricula.html)

[5] [IT2006] The ACM SIGITE Task Force on IT Curriculum. Information Technology 2006, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. As this is written, the IT curriculum report is in development. We expect the review and development process to be completed in 2006. At the moment, the current draft is called *IT2005* and is available for public review and comment.

[6] [K-12] ACM K-12 Task Force Curriculum Committee. A Model Curriculum for K-12 Computer Science. ACM Press, 2004. (http://csta.acm.org/Curriculum/sub/k12final1022.pdf)

[7] [UKQAA, 2000] Quality Assurance Agency for Higher Education Computing. A Report on Benchmark Levels for Computing. Southgate House, Gloucester, England, April 2000.

[8] (http://www.qaa.ac.uk/academicinfrastructure/benchmark/honours/computing.asp)

[9] BUTKUS, M.A., AND KELLEY, M.B. 2004. Approach for integrating professional practice issues into undergraduate environmental engineering design projects. *Journal of Professional Issues in Engineering Education and Practice 130:3*, 166-172.

[10] CHUMER, M.J. 2002. Towards An Understanding Of User-Centeredness Within Information Technology Diffusion: A Self-Ethnography. *Doctoral dissertation, Rutgers The State University of New Jersey*. Advisor: Ronald E. Rice.CSTB99: Committee on Information Technology Literacy, National Research Council 1999. *Being Fluent*

[11] EKSTROM, J.J., DARK, M.J., LUNT, B.M., AND REICHGELT, H. 2006. A research agenda for information technology: does research literature already exist?. In *Proceedings of the 7th Conference on Information Technology Education* (Minneapolis, Minnesota, USA, October 19 - 21, 2006) .

[12] REDWINE, S.T. (editor) 2007. Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software. US Department of Homeland Security.Available at

[13] https://buildsecurityin.us-cert.gov/daisy/bsi/resources/dhs/927.html.

[14] [UKQAA, 2000] Quality Assurance Agency for Higher Education, "Computing, a report on benchmark levels for Computing," Southgate House, Gloucester, England, April 2000.

[15] [Zadeh, 1968] Lofti A. Zadeh. Computer science as a discipline. Journal of Engineering Education, 58(8) :913-916, April 1968.
[16] [Shaw, 1991] Mary Shaw and James E Tomayko. Models for undergraduate courses in software engineering. Pittsburgh: Software Engineering Institute, Carnegie Mellon University, January 1991.
[17] [ITiCSE'04] Innovation and Technology in Computer Science Education, <http://www.iticse04.leeds.ac.uk/>, Leeds, England, 28-30 June 2004.
[18] [ITiCSE'03] Innovation and Technology in Computer Science Education, <http://www.cs.utexas.edu/users/csed/iticse/>, Thessaloniki, Greece, 30 June – 2 July 2003
[19] [UKQAA, 2000] Quality Assurance Agency for Higher Education Computing. A Report on Benchmark Levels for Computing. Southgate House, Gloucester, England, April 2000.
[20] (http://www.qaa.ac.uk/academicinfrastructure/benchmark/honours/computing.asp)
[21] http://www.acm.org/education/curricula.html
[22] http://computer.org/curriculum.