AC 2012-4645: REVISIONS TO SOFTWARE ENGINEERING 2004: CURRICULUM GUIDELINES FOR UNDERGRADUATE DEGREE PROGRAMS IN SOFTWARE ENGINEERING

Dr. Mark Ardis, Stevens Institute of Technology

Mark Ardis is a Distinguished Service Professor in the School of Systems and Enterprises at Stevens Institute of Technology. He is interested in the professionalization of software engineering, especially through teaching and technology transfer. In his career, Ardis has helped create academic programs in software engineering at five schools. He received a B.A. in mathematics from Cornell University and a Ph.D. in computer science from the University of Maryland.

Prof. David Budgen, Durham University, UK

David Budgen is a professor of software engineering in the School of Engineering & Computing Sciences at Durham University in the U.K. His research interests include empirical software engineering, evidence-based SE, software design, and healthcare computing. He was awarded a B.Sc. (honors) in physics and a Ph.D. in theoretical physics from Durham University, following which he worked as a research scientist for the Admiralty and then held academic positions at Stirling University and Keele University, before moving to his present post at Durham University in 2005. He is a member of the IEEE Computer Society, the ACM, and the Institution of Engineering & Technology (IET).

Dr. Gregory W. Hislop, Drexel University

Gregory Hislop is a professor of information science and technology and computer science at Drexel University. His interests include software engineering, computing education, and use of technology in education. Prior to joining Drexel, Hislop spent almost 20 years in IT practice with particular emphasis on products and services for enterprise systems management.

Dr. Rene McCauley, College of Charleston

Rene McCauley is a professor in the omputer Science Department at the College of Charleston in South Carolina, USA, where she directs the master's program in computing and information sciences. She is Chair of the ACM Special Interest Group on Computer Science Education (SIGCSE). She co-chaired the 2008 ACM/IEEE-CS task force that revised the computer science curriculum volume. Her research is focused on computer science education, specifically how novices learn to program. She is a member of the ACM and the IEEE Computer Society.

Dr. Mark J. Sebern, Milwaukee School of Engineering

Mark J. Sebern is a professor in the electrical engineering and computer science Department at the Milwaukee School of Engineering (MSOE), and Founding Program Director for MSOE's undergraduate software engineering program. He has served as an ABET program evaluator for software engineering, computer engineering, and computer science, and is currently a member of the ABET Engineering Accreditation Commission.

Revisions to Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering

1. Introduction

Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (*SE 2004*)[1] is one volume in a set of computing curricula adopted and supported by the ACM and the IEEE Computer Society. In order to keep the software engineering guidelines up to date the two professional societies established a review project in early 2011. This paper describes that review effort and plans to revise the guidelines over the next year and a half.

2. Project organization

The charge for the SE2004 Review Team was as follows:

1. Conduct a "consultation process" to collect information and opinion from the principal curriculum stakeholders (industry and academia) about the need for modification of the curriculum model (body of knowledge, curriculum architecture, pedagogy, infrastructure, etc.). It was recommended that the team use a variety of methods for this consultation: web/email communication, surveys, individual academic and industrial contacts, etc.
2. Analyze and assess the results of the consultation process to determine the type and extent of change needed.
3. Prepare a report for the IEEE-CS Educational Activities Board and the ACM Education Board. The report should describe the consultation process, present an analysis and assessment of the information collected, and make recommendations concerning the following:
    a. the type and extent of revision needed to SE2004
    b. an estimate of the amount of effort needed (e.g., number of volunteers and total hours) and a proposed schedule for the recommended revision.

The review team consisted of 6 people: 3 representatives from the IEEE Computer Society and 3 representatives from the ACM. The team was chaired by the leader of the Computer Society group, as they are the lead professional society for this set of curriculum recommendations. We were fortunate to have team members with extensive experience in software engineering education and curriculum design.

The team held one-hour teleconferences monthly until the end of the project, when we held more frequent meetings. Team members volunteered for action items each month and reported progress at the following meeting. Almost all action items were completed on time.

We used Google Docs for some of the shared materials, especially lists of stakeholders and email addresses. This was occasionally useful during meetings, where we could all see the same document under discussion. But the main value of this type of sharing was asynchronous updating by different members of the team.

3. Outreach

The team first constructed a comprehensive list of stakeholder groups from academia, industry and government. Groups included accredited academic programs, special interest groups, and software engineering programs in several countries. We were especially interested in identifying people from industry or government who had an interest in software engineering education.

In some cases we were able to identify key individuals who would forward our email to members of their organizations. For example, we used this method for mailings to two of the ACM special interest groups: the Special Interest Group on Computer Science Education (SIGCSE) and the Special Interest Group on Software Engineering (SIGSOFT). Through contacts in the Computer Society our email was forwarded to all recipients of the Certified Software Development Associate (CSDA) and Certified Software Development Professional (CSDP) certificates[2].

In some cases we constructed our own lists of email addresses. We were able to compose lists for education boards of the ACM and the Computing Society and for several other stakeholder groups. For example, we were able to acquire contact information for heads of accredited programs in computer science and software engineering in the United States and Canada.

The annual Conference on Software Engineering Education and Training (CSEE&T) was held just before our survey was available. This was an important event for us, as it frequently attracts software engineering educators who would be interested in our work. Friends and colleagues in the IEEE helped us arrange a Birds-of-a-Feather session at CSEE&T that provided useful feedback.

We established a community portal page on the Ensemble Computing Portal website[3]. This provided a convenient place to advertise our survey, and it placed us near other curriculum efforts that have portals there. We also distributed flyers at other conferences and wrote about the survey in a SIGSOFT Software Engineering Notes column[4].

4. Survey

We created a survey on SurveyMonkey to collect feedback about the current state and use of *SE 2004*. The first part of the survey asked about the respondent's background and familiarity with *SE 2004*, while later sections asked about specific sections of the guidelines. We allowed respondents to skip some of the more detailed questions about the Software Engineering Education Knowledge (SEEK) section if they felt unprepared to answer those. Respondent background data was important to us, as we were concerned about reaching stakeholders across a wide spectrum of occupations and localities.

We received usable responses from 42 different countries, as shown in Table 1. The majority of survey participants were from North America, even though we worked hard to recruit responses from other parts of the world. We sent email to hundreds of potential participants in Central and South America, Europe and Asia as part of our publicity campaign.

Argentina	3	Ghana	1	Russian Fed.	1
Australia	7	Greece	1	Serbia	1
Austria	3	Hong Kong	1	Singapore	1
Belgium	1	India	9	South Korea	1
Bolivia	1	Israel	1	Spain	3
Brazil	4	Italy	5	Sweden	6
Canada	28	Japan	1	Switzerland	6
Chile	1	Mexico	1	Taiwan	3
China	4	Netherlands	3	Uganda	1
Colombia	4	New Zealand	3	Ukraine	1
Denmark	2	Norway	2	United Kingdom	10
Finland	2	Peru	1	United States	333
France	1	Poland	1	Venezuela	1
Germany	13	Portugal	3	Vietnam	1

Table 1. Country of origin of responses to survey

Most of the respondents had graduate degrees, as shown in Table 2.

Associate Degree	0
Bachelor's Degree	79
Master's Degree	138
PhD or equivalent	245
Other	15

Table 2. Highest degree of respondents

This is not surprising, since our main stakeholder group consists of faculty at colleges and universities.

While 71 identified their degree as being in Software Engineering, only 17 claimed that their degree program had been based upon *SE 2004*. Table 3 lists counts for those who definitely identified their degree program as having been based upon *SE 2004*.

Anna Univ. Tamil Nadu	India	1
Univ. of Castilla, La Mancha	Spain	1
Univ. of Ottawa	Canada	2
Rochester Inst. Tech.	USA	7
Rose-Hulman	USA	3
	Taiwan	1
Vanderbilt Univ.	USA	1
Univ. of Western Ontario	Canada	1

Table 3. Programs based on *SE 2004*

We asked respondents about their role in software engineering education. Table 4 lists the breakdown of roles.

Teacher	134
Researcher	88
Software Developer	156
Administrator	26
Retired	13
Other	60
Not Answered	1

Table 4. Role of respondents

A quick review of 'other' showed a range of responses. Some considered their role as spanning more than one of these equally, while others were consultants, responsible for quality management etc.

There was a fairly even spread of experience in the practice of software engineering amongst respondents. Table 5 shows the profile.

Less than 3 years	63
3-6 years	56
6-9 years	36
9-12 years	54
More than 12 years	268

Table 5. Years of experience with software engineering practice

Experience with software engineering education was also evenly distributed, as shown in Table 6.

Less than 3 years	95
3-6 years	90
6-9 years	51
9-12 years	52
More than 12 years	144
No experience	45

Table 6. Years of experience with software engineering education

A key element of SE 2004 is the structure of the SEEK. We asked survey participants about the overall structure and about specific sections. Most participants agreed that the Knowledge Areas were still relevant, as shown in Figure 1.

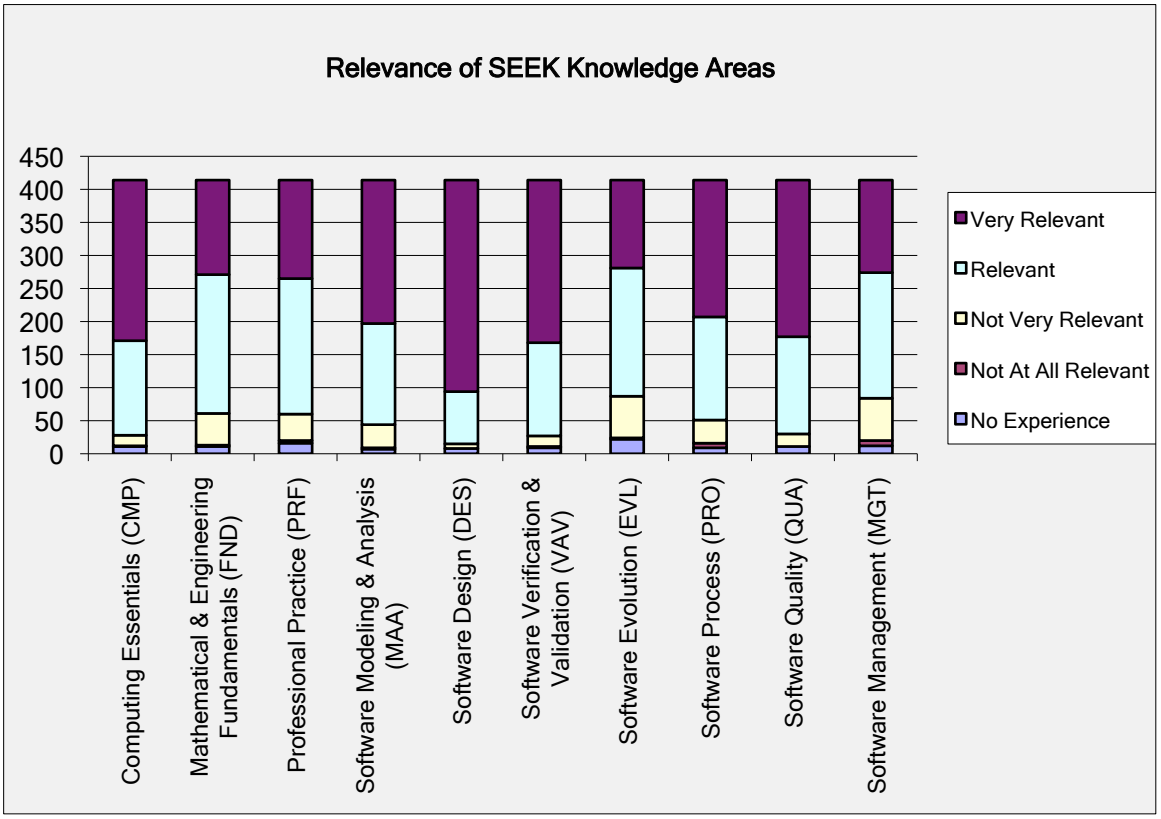

Figure 1. Relevance of SEEK Knowledge Areas

We received useful feedback from some respondents about the relative percentage of time that should be spent on individual topics within knowledge areas. For example, respondents felt that more time should be spent on requirements fundamentals within the Software Modelling and Analysis area, as shown in Figure 2.

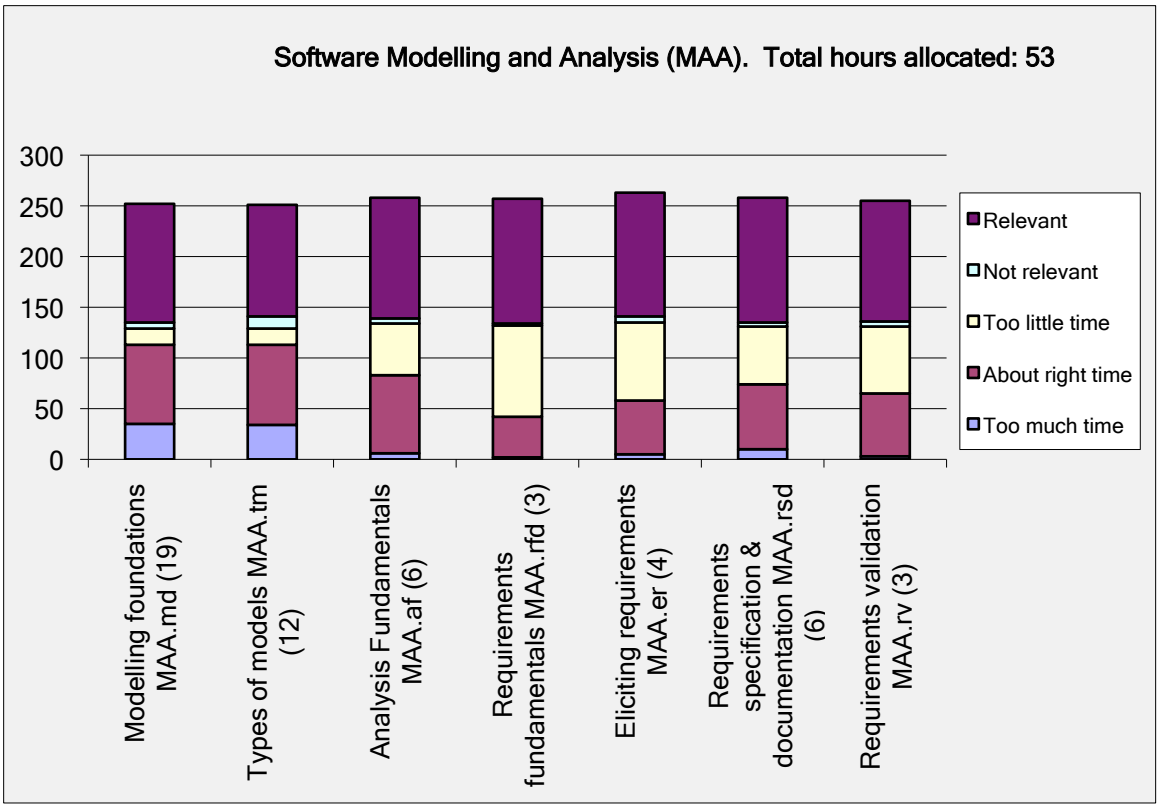

Figure 2. Allocation of time to Software Modelling and Analysis Knowledge Area

We asked about elements that were missing from the SE2004 Curriculum Guidelines or from the SEEK. There were a few trends, especially the incorporation of more modern software development methods, and a need for more emphasis on security.

We asked about sections of *SE 2004* that were useful, and those that were not as useful. It seems clear that different readers of the guidelines use them for different purposes. Although some sections of the recommendations may not be relevant to some readers, they are to others.

5. Revision estimation

Based on the results of the survey of constituents, we found that the overall structure of *SE 2004* is sound and does not need to be changed. The SEEK was found to need only minor revisions. The areas of revision identified include:

- agile methods: these have become more popular and successful
- security: increasingly important as more services are exposed to attack
- service-oriented computing: these have become more popular and important

To estimate the amount of effort required to accomplish the identified revisions we used a Wideband Delphi[5] process to compare estimates and reach consensus. That is, each member of the team proposed an initial estimate of expected effort and duration for revision, review of revisions, and final editing to respond to stakeholder feedback. Estimates included comments explaining rationale for the values chosen. After sharing these estimates we each proposed a new

set of estimates, with new rationales. We converged on a common estimate of expected effort and duration for the revision project within a couple rounds of this process.

6. Description of SE 2013 project

A project to revise the guidelines, called SE 2013, has been approved by the IEEE Computer Society and the ACM. A team of 6 volunteers from the Computer Society and the ACM will meet once/month by teleconference to coordinate the work. Table 7 shows the major milestones for the revision project. Note that we hope to obtain useful feedback on an early draft of revisions from attendees at ASEE 2012. The entire project should take about a year and a half.

Activity	Planned Completion Date	Expected Result or Deliverable
Kickoff and initial planning	February 1, 2012	Project plan
Make first draft of updates to guidelines	June 15, 2012	Initial draft of SE 2013
Present first draft of updates for review at ASEE 2012	June 20, 2012	Collected comments from stakeholders
Revise and complete updates to guidelines	October 1, 2012	Revised draft of SE 2013
Present final draft of guidelines for public review at FIE 2012	October 15, 2012	Collected comments from stakeholders
Collect and respond to feedback	March 1, 2013	Final version of SE 2013
Present final version of guidelines at SIGCSE 2013	March 9, 2013	Increased dissemination, feedback from stakeholders
Present final version of guidelines at CSEE&T 2013	May 23, 2013	Increased dissemination, feedback from stakeholders

Table 7. Project milestones for SE 2013

Other curriculum revision efforts are being conducted concurrently with SE 2013. Two of particular interest are CS 2013, a revised set of guidelines for computer science curricula, and the Guide to the Software Engineering Body of Knowledge (SWEBOK)[6], which is undergoing review of recent updates. We expect to work closely with members of both of those projects.

After completing the revisions we plan on sharing results with stakeholders at important conferences, such as SIGCSE and CSEE&T. We also hope to publish our results in journals and magazines that will reach other stakeholders.

7. Acknowledgments

We thank the many respondents to our survey who provided much useful feedback on the content and use of *SE 2004*. We also thank our colleagues who helped disseminate the call for participation in the survey. Finally, we thank the professional societies that sponsored this project and provided helpful resources.

8. References

1. ACM/IEEE-CS Joint Task Force on Computing Curricula, *Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*, August 2004. http://www.acm.org/education/curricula.html

2. IEEE Computer Society Certified Software Development Associate (CSDA) and Certified Software Development Professional (CSDP), http://www.computer.org/portal/web/certification

3. Ensemble Computing Portal, http://www.computingportal.org/se2004rtf

4. Ardis and P. Henderson, Software Engineering Education (SEEd): Guidelines for Software Engineering Education, *ACM SIGSOFT Software Engineering Notes 36(4)*, July 2011.

5. Boehm, Barry W., *Software Engineering Economics*, Englewood Cliffs, NJ: Prentice Hall, 1981, ISBN: 0138221227

6. A. Abran et al. (ed.s) *Guide to the Software Engineering Body of Knowledge (SWEBOK)*, IEEE CS Press. www.computer.org/portal/web/swebok/htmlformat